Model-based redesign of global transcription regulation

Javier Carrera[1,2], Guillermo Rodrigo[1] and Alfonso Jaramillo[3,4]

[1]Instituto de Biologia Molecular y Celular de Plantas, CSIC-Universidad Politecnica de Valencia, Camino de Vera, 46022 Valencia, Spain.
[2]Instituto de Aplicaciones en Tecnologias de la Informacion y las Comunicaciones Avanzadas (ITACA), Universidad Politecnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain.
[3]Laboratoire de Biochimie, Ecole Polytechnique - CNRS, Route de Saclay, 91128 Palaiseau Cedex, France.
[4]Epigenomics Project, Universite d'Evry Val d'Essonne – Genopole - CNRS, 523 Terrasses de l'Agora, 91034 Evry Cedex, France.

SUPPORTING MATERIAL: Additional Files and Figures

► Additional File 1: *InferGene_Package.zip*
InferGene Package contains: Generator Artificial Genomes (genArtGen.cc and Linpack library), InferOpe (inferope.cc), ZTop (ZTop.c), InferGene (infergene.cc, Linpack library) and Linpack library (blas1_d.c, blas1_d.o, blas1_d.h, linpack_d.c, linpack_d.o, linpack_d.h). This file is available upon request. Additionally, the user needs to download the CLR algorithm [1].

► Additional File 2: *InferEcoli.xml*
SBML file [5] containing the regulatory model of *E. coli.* Files used to infer the *E. coli* network: microarray data from M3D [2] (Affymetrix expression normalized via RMA [3] for 4345 genes and 189 experimental conditions); genome annotation from RegulonDB [4], list of 328 transcription factors and operons (4345 genes are clustered in 3333 operons).

► Additional File 3: *BioFunctEcoli.xls*
Classification of biological functions in *E. coli* from EcoCyc [6] using expression profiles not included on the training set. We show the degree of connectivity for each biological function depending on the number of protein-protein interactions. We also show the number of genes regulated by 1-10 TFs and non-regulated genes (from constitutive promoters).

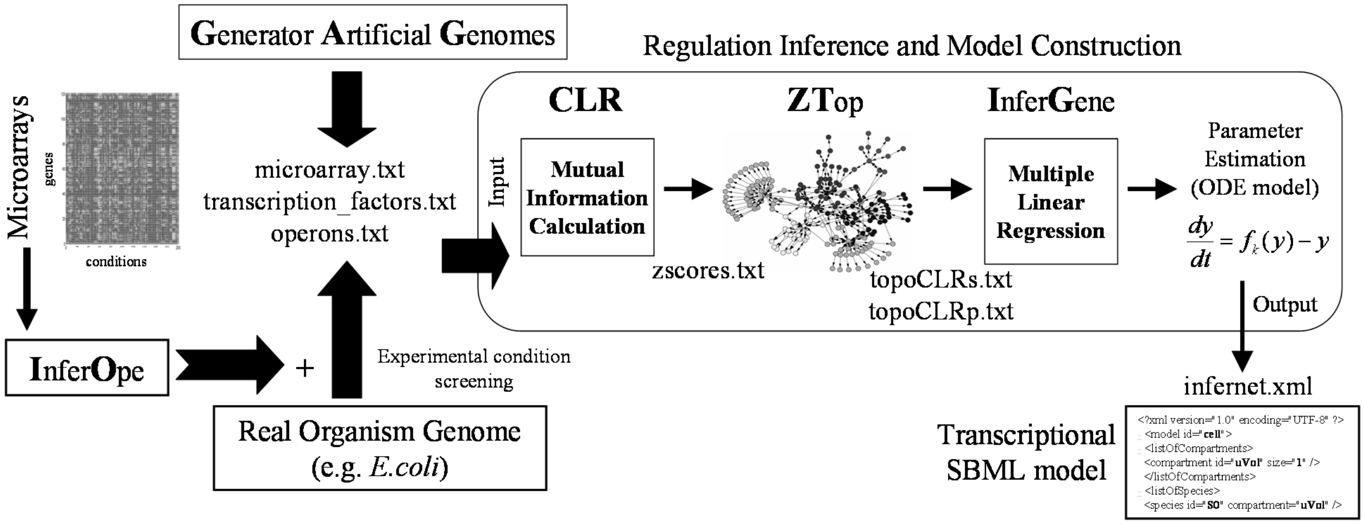
► Additional File 4: *ScoringConditions.xls*
Histogram of the expression error (i.e., error between the measured expression and the predicted one) across all genes for a fixed condition $\Psi_c = \frac{1}{n} \sum_{g \in genes} |\hat{y}_{gc} - y_{gc}|$ where *n* is the number of genes. This analysis has been done for *E. coli* using the M3D compendium.

► Additional File 5: *ScoreOperonsEcoli.xls*
Prediction of InferGene inferring the *E. coli* network over 189 microarrays. We show the expression error for each operon $\Delta_{op} = \frac{1}{n} \frac{1}{m} \sum_{g \in operon} \sum_{c \in set} |\hat{y}_{gc} - y_{gc}|$, where $\hat{y}_{gc}$ is the predicted expression profile under conditions that were not used in the original training data set, y_{gc} the measured gene expression of each operon, n the number of operons, and m the number of conditions.

► Additional File 6: *ScoreOperonsAG.xls*
Prediction of InferGene for an artificial genome of 500 genes with 250 different perturbations generated with GAG. We show the expression error for each operon $\Delta_{op} = \frac{1}{n} \frac{1}{m} \sum_{g \in operon} \sum_{c \in set} |\hat{y}_{gc} - y_{gc}|$, where $\hat{y}_{gc}$ is the predicted expression profile under conditions that were not used in the original training data set, y_{gc} the measured gene expression of each operon, *n* the number of operons, and m the number of conditions. In addition, we show the parameter deviation $\Gamma = \frac{1}{n} \frac{1}{p} \sum_{g \in genes} \sum_{p \in param} |\hat{\beta}_{gp} - \beta_{gp}|$ where $\hat{\beta}_{gp}$ is the predicted parameter, β_{gp} the model parameters, and p the number of parameters.

The InferGene package

The InferGene package is a set of applications (see Fig. S1) developed in C++, which can be applied to infer regulatory networks from expression profiles. The **G**enerator of **A**rtificial **G**enomes is applied to build synthetic models and microarray data. **I**nfer**O**pe is an algorithm to infer operons based on co-expression patterns. This is useful only in case of prokaryotes. The **CLR** algorithm outputs the z-scores for all possible interactions, and **ZT**op converts this z-score matrix into a matrix of regulations according to a given threshold (e.g., 5-10). Finally, **I**nfer**G**ene constructs the SBML model of the inferred regulatory network.

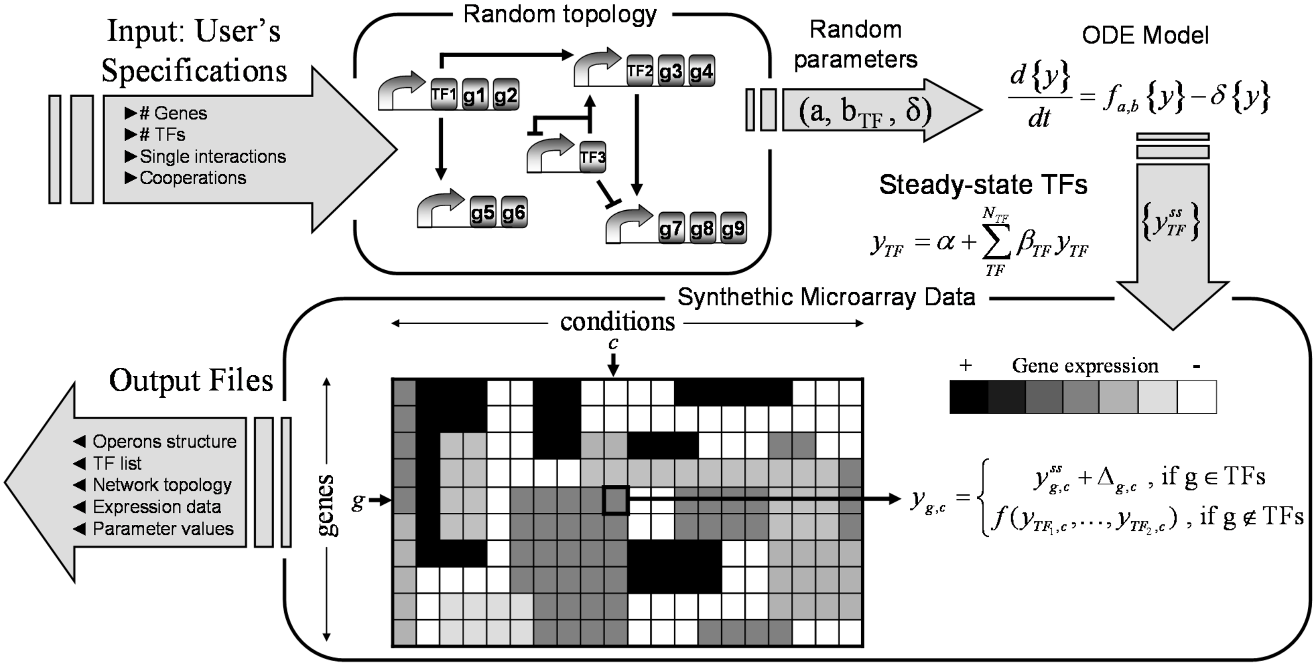

Generator Artificial Genomes
Regulation Inference and Model Construction
Microarrays
genes
conditions
microarray.txt
transcription_factors.txt
operons.txt
Input
CLR
Mutual Information Calculation
ZTop
zscores.txt
topoCLRs.txt
topoCLRp.txt
InferGene
Multiple Linear Regression
Parameter Estimation (ODE model)
$\frac{dy}{dt} = f_k(y) - y$
Output
InferOpe
+
Experimental condition screening
Real Organism Genome (e.g. E.coli)
infernet.xml
Transcriptional SBML model

```
<?xml version="1.0" encoding="UTF-8" ?>
- <model id="cell">
- <listOfCompartments>
  <compartment id="uVol" size="1" />
  </listOfCompartments>
- <listOfSpecies>
  <species id="S0" compartment="uVol" />
```

Fig. S1: Flux diagram of the InferGene package applications.

In silico generation of microarray data

We have developed a Generator of Artificial Genomes (GAG) to *in silico* create expression profiles (see Fig. S2). To design such genomes, we specify the topological properties of the network. Then, random kinetic parameters are associated. The output is the set of synthetic expression profiles, as well as the network final topology and parameters. In Fig. 3 of the manuscript we show the algorithm performance for several artificial genomes. In Fig. S3 we show the prediction of the kinetic parameters for the correctly predicted regulations, achieving correlations above 0.90.



Input: User's Specifications
►# Genes
►# TFs
►Single interactions
►Cooperations
Random topology
TF1 g1 g2
TF2 g3 g4
TF3
g5 g6
g7 g8 g9
Random parameters
(a, b_{TF}, δ)
ODE Model
$\frac{d\{y\}}{dt} = f_{a,b}\{y\} - \delta\{y\}$
Steady-state TFs
$y_{TF} = \alpha + \sum_{TF}^{N_{TF}} \beta_{TF} y_{TF}$
$\{y_{TF}^{ss}\}$
Synthethic Microarray Data
conditions
c
genes
g
+ Gene expression -
$y_{g,c} = \begin{cases} y_{g,c}^{ss} + \Delta_{g,c}, & \text{if } g \in \text{TFs} \\ f(y_{TF_1,c}, \ldots, y_{TF_2,c}), & \text{if } g \notin \text{TFs} \end{cases}$
Output Files
◄ Operons structure
◄ TF list
◄ Network topology
◄ Expression data
◄ Parameter values

Fig. S2: Generation of an artificial genome model to get synthetic microarray data. We have developed a computational algorithm (GAG) to construct such model, where the user inputs the total number of genes and TFs as well as the percentage of single regulations and cooperations. GAG generates a random network following those specifications with the corresponding model parameters: the constitutive transcription rate a, the regulatory coefficients b_TF, and the mRNA degradation coefficient δ (see ODE equations in the manuscript). In the presented network, consisting of 5 operons, 3 TFs and 9 non-regulatory genes (g_i), arrows mean activation and blunt lines repression. The regulatory function (f) is assumed linear and the expression is calculated in the steady state (ss), where α=a/δ and β_TF=b_TF/δ. Later, GAG gives the *in silico* microarray data. We select a TF or a subset of TFs and we perturb the expression in the steady state (Δ_g,c). Then we recomputed the whole expression profile using the model. Therefore, GAG outputs the list of operons and TFs, the regulatory network (adjacency matrix) with the corresponding model parameters, and the synthetic expression profiles (represented as a color-scale grid, where genes are disposed in rows and experiments in columns).

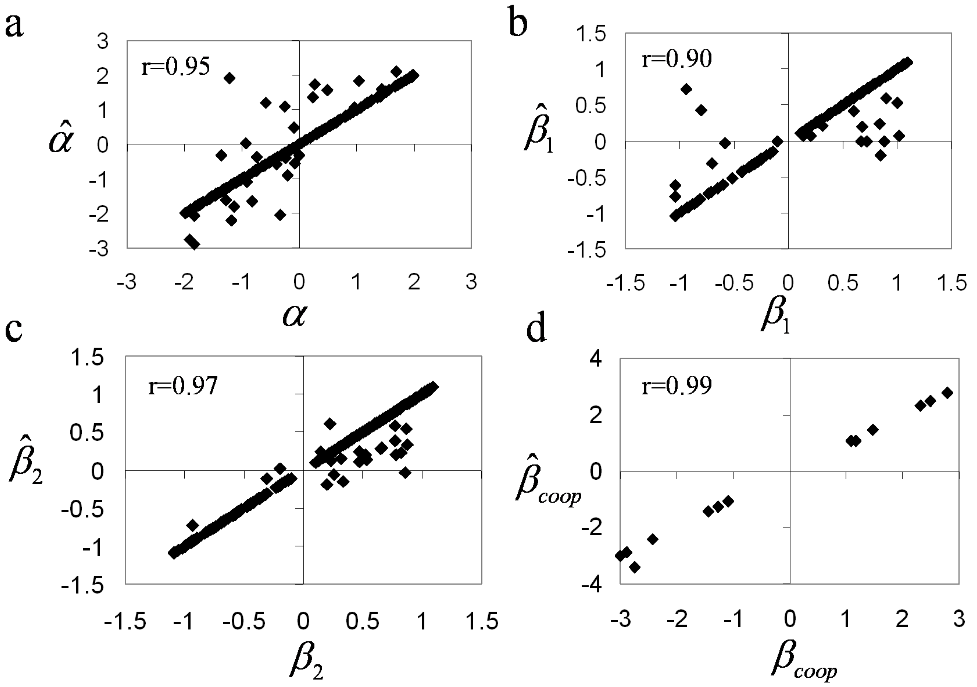


Fig. S3: Prediction of the model parameters for an artificial genome of 500 genes and 50 TFs with 50 perturbations for training. Each condition is generated by perturbing the steady-state of 12% of the TFs. We show the correlation between the estimated parameters (^α, ^β_1, ^β_2, ^β_coop) and the predefined parameters (α, β_1, β_2, β_coop) for the model. In (a) estimation of basal transcription rates (α), in (b) regulatory coefficients for promoters regulated by one TF (β_1), in (c) regulatory coefficients for promoters regulated by more than one TF (β_2), and in (d) regulatory coefficients for synergistic effects (β_coop). Every plot shows a correlation coefficient (r) above 0.90.

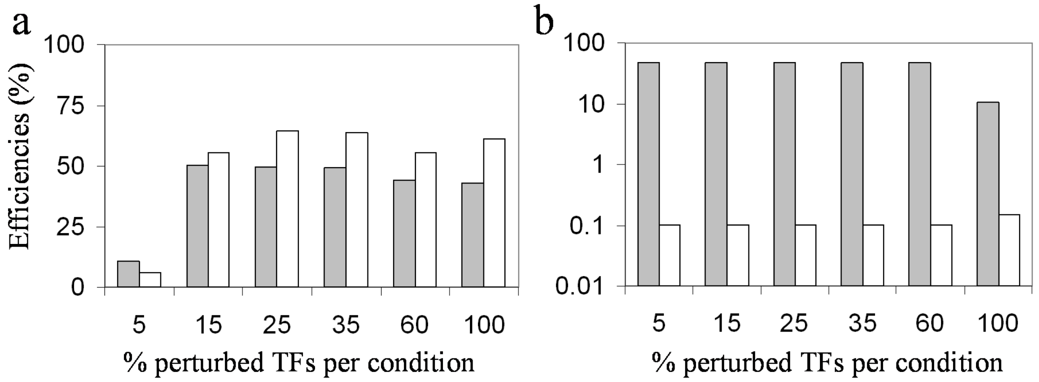


Fig. S4: Prediction fitness for different sets of perturbed TFs (% referred to the total number of TFs) per condition when generating *in silico* expression data (sensitivity in gray, precision rate in white). In (a) single interactions, in (b) combinatorial regulations.

To generate *in silico* data we have perturbed the TF steady states and we recomputed the rest of gene expression values. By perturbing one single TF per condition is a good approach to unveil single interactions, but at this time synergistic regulations are difficult (even impossible) to capture. On the other hand, by perturbing many TFs at the same time can produce. In that way, an optimization of the number of TFs to be perturbed in each experimental condition is required for generating synthetic microarray data. We evaluate the fitness as precision and recall for different TF sets (see Fig. S4).

The performance of the algorithm depends on the quantity and also quality of data. As synthetic data are very clean, the efficiency of the algorithm reaches high values. However, with real data this performance can be dramatically reduced due to the microarray noise. Therefore, we have added a term of noise to the *in silico* expression values to generate more realistic data. In that way, the noisy value is $\tilde{y}=(1+\eta)\,y$, where η is a uniform random distribution between its maximum amplitude (i.e., -|η|< η< |η|). In Fig. S5 we show the efficiency of the algorithm using synthetic data (200 conditions from a genome of 5000 genes) for different noise amplitudes. The noise in the *E. coli* expression data used corresponds to |η|<0.1 (see Fig. S6), thus we expect high inference efficiencies.

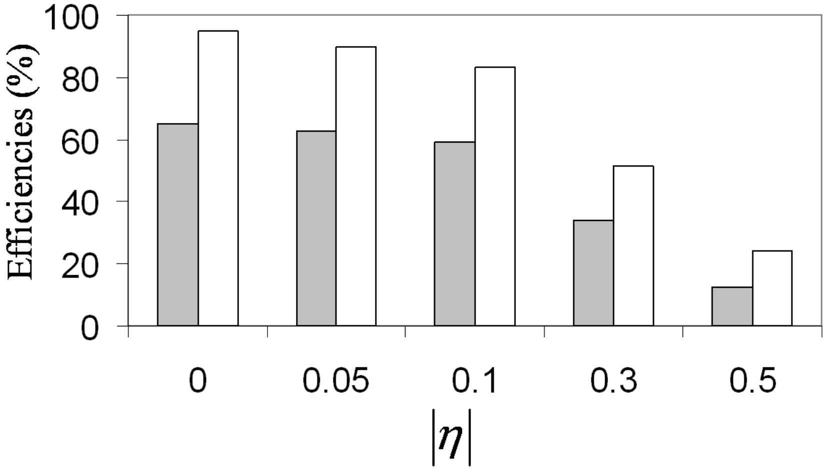


Fig. S5: Efficiency of the algorithm versus the noise level using synthetic data. We show the precision rate (white) and sensitivity (gray) for different noise amplitudes on the expression value. Here, we have selected interactions with z-score up to 6. We have used an artificial genome of 5000 genes and 500 TFs, generating 200 conditions by perturbing randomly the 12% of TFs in each condition without combinatorial regulations.

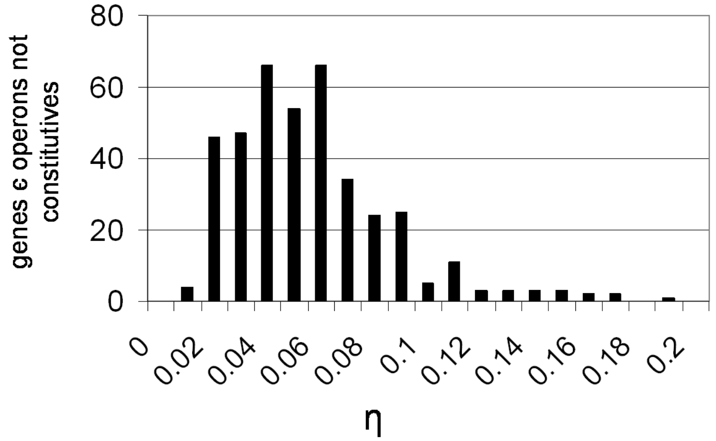


Fig. S6: Noise distribution in the *E. coli* expression data used computed throughout the gene expressions from the same operon.

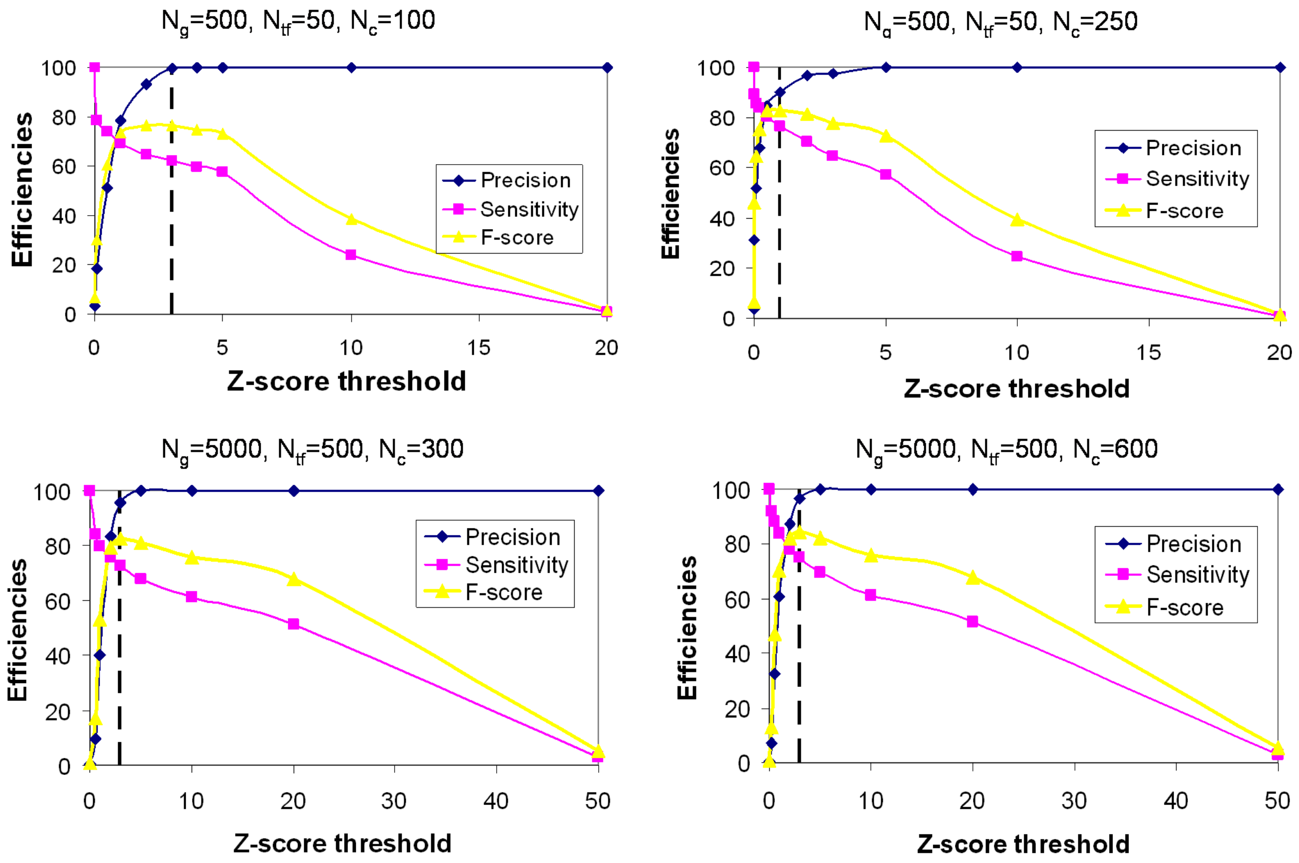
Cut-off threshold selection

For every pair operon-TF a Z-score is calculated using its Mutual Information (MI). Then, to infer the topology of the network it is necessary to establish a threshold. Regulations will be selected if their Z-score is above the threshold. However, it is difficult to fix that, and this value may depend on the nature of the system and the data used. To address this question, we have constructed several artificial genomes with different size and generated expression data. We define F-score (F) as the global performance of the inference following

$$F=\frac{2PS}{P+S}$$

where P is the precision rate and S the sensitivity. We have analyzed the degree of prediction of the resulted network for different threshold values showing an optimum value for the threshold (see Figs. S7). We have generated two genomes, one of 500 genes and 50 TFs and another of 5000 genes and 500 TFs. Then, we have created for each one two data sets varying the number of conditions (from 100 to 250 in the first case, and from 300 to 600 in the other one).

In Fig. S8 we superpose the resulting F-score for the different systems observing a common region to select the optimum threshold value (approximately between 2 and 7). Values close to 2 will have a higher sensitivity than precision, and values close to 7 will have higher precision than sensitivity. Notice that instead for low scales where the inference reaches an optimum, for large genomes the F-score reaches a light flat from a given threshold value giving a more precise model. This allows us to fix 6.92 as threshold to infer the *E. coli* regulatory model.


N_g=500, N_{tf}=50, N_c=100

N_g=500, N_{tf}=50, N_c=250

N_g=5000, N_{tf}=500, N_c=300

N_g=5000, N_{tf}=500, N_c=600

Fig. S7: Efficiencies (Precision, Sensitivity and F-score) with respect to the selected Z-score threshold for different artificial genomes and data sets.

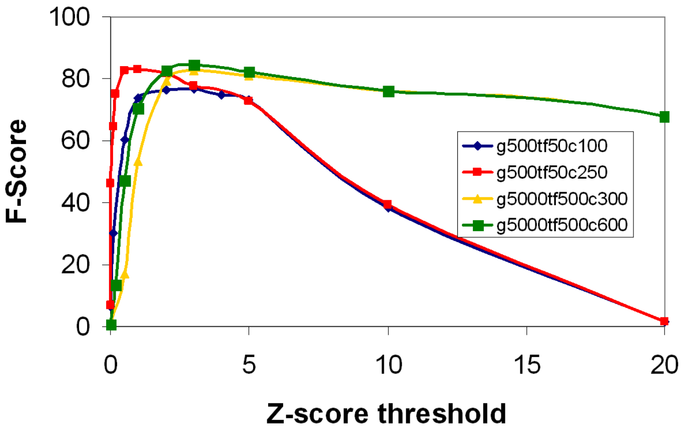


Fig. S8: F-score of several inferred topologies against the selected Z-score threshold.

Genome annotation

We have superposed genome annotation to study the best predicted biological functions. For that, we have used the EcoCyc classification to group genes by biological functions and to rank those groups according to their level of prediction (see Fig. S9).

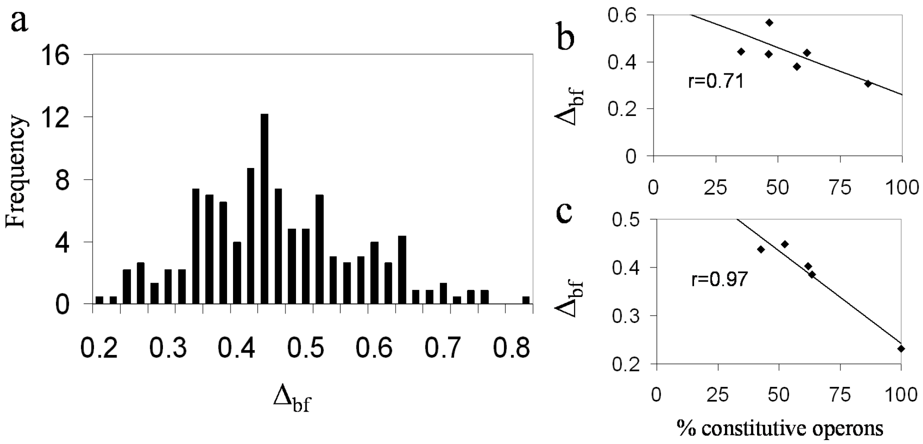


Fig. S9: Scoring *E. coli* biological functions. (a) Histogram of the normalized expression error on the transcriptomic profile predicted for each biological function (Δ_{bf}). Functions from the EcoCyc database at minimum level. (b,c) Linear regressions between Δ_{bf} and the number of constitutive operons for the biological functions of gene product location and cell processes, respectively.

Error distributions

We have computed the expression error in operons by classifying them according to their promoter type (i.e., constitutive, regulated by one TF, regulated by two TFs, and regulated by more than two TFs). In Fig. S10 we plot this showing that the operons with two-regulated promoters are better predicted.

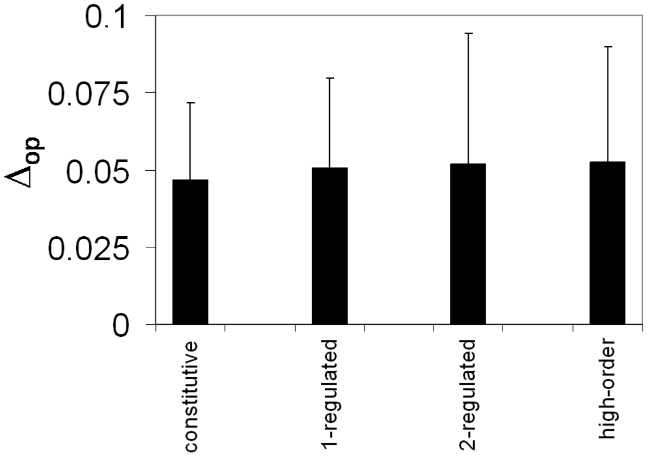


Fig. S10: Expression error in operons with different types of promoters.

We have analyzed the predictive power of InferGene by calculating a score based on the error made on the expression (Δ_{op}), and other score based on the error made on the model parameters (Γ). To perform such analysis we have generated a network with GAG of 500 genes across 250 conditions (see Fig. S11). The median for Δ_{op} was 0.009, and for Γ was around 0.01.

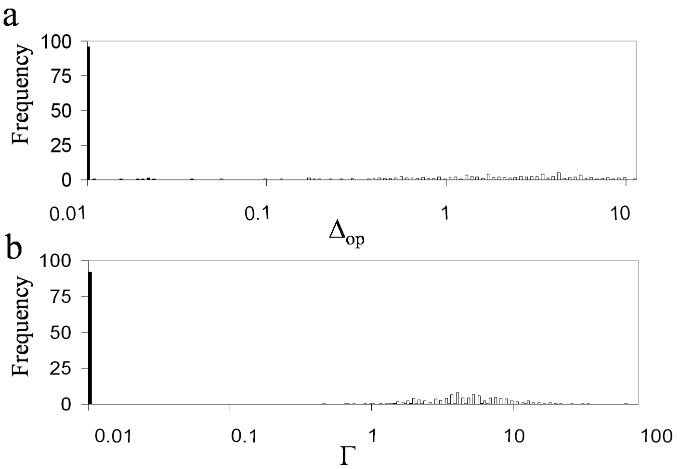


Fig. S11: Predictive power of InferGene against an artificial genome of 500 genes accross 250 different experiments generated with GAG. (a) Distribution of the expression errors in the operons (Δ_{op}). (b) Distribution of the parameter errors (Γ). White bars represent random distributions in both cases. The operons in which there are only TFs are not considered to compute those scores.

It is clear that a large number of experimental conditions (perturbations) is required to construct an accurate regulatory model. However, not all conditions contribute equally to unveil such regulations. Perturbations that affect the TF expressions are very fruitful to capture the transcription network. At the same time, there are conditions in which the model predicts better the expression profile. In that way, the histogram computed with the M3D compendium for *E. coli,* under conditions not included in the training set, approximately shows a normal distribution (see Fig. S12). Here, we score each condition using $\Psi_c = \frac{1}{n} \sum_{g \in genes} |\hat{y}_{gc} - y_{gc}|$. Such scores are provided in the additional file 4.

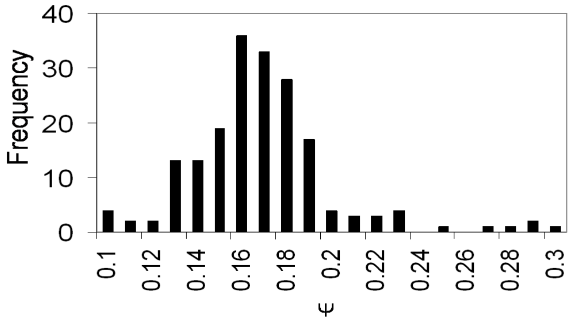


Fig. S12: Histogram of the expression error across all genes for a fixed condition. This plot is referred to *E. coli*.

In addition, we have performed a K-fold cross-validation (K=9, dividing the set of conditions in 9 parts) to validate the predictions. Like that, we use 169 conditions for training and 20 for testing, repeating the process 9 times. Notice that one testing set has 29 conditions. In Fig. S13 we plot all histograms of Δ_{op} for each fold showing that all distributions are not significantly different. Moreover, we have computed the asymptotic distribution (mean 0.04) by averaging all previous histograms (see Fig. S14).

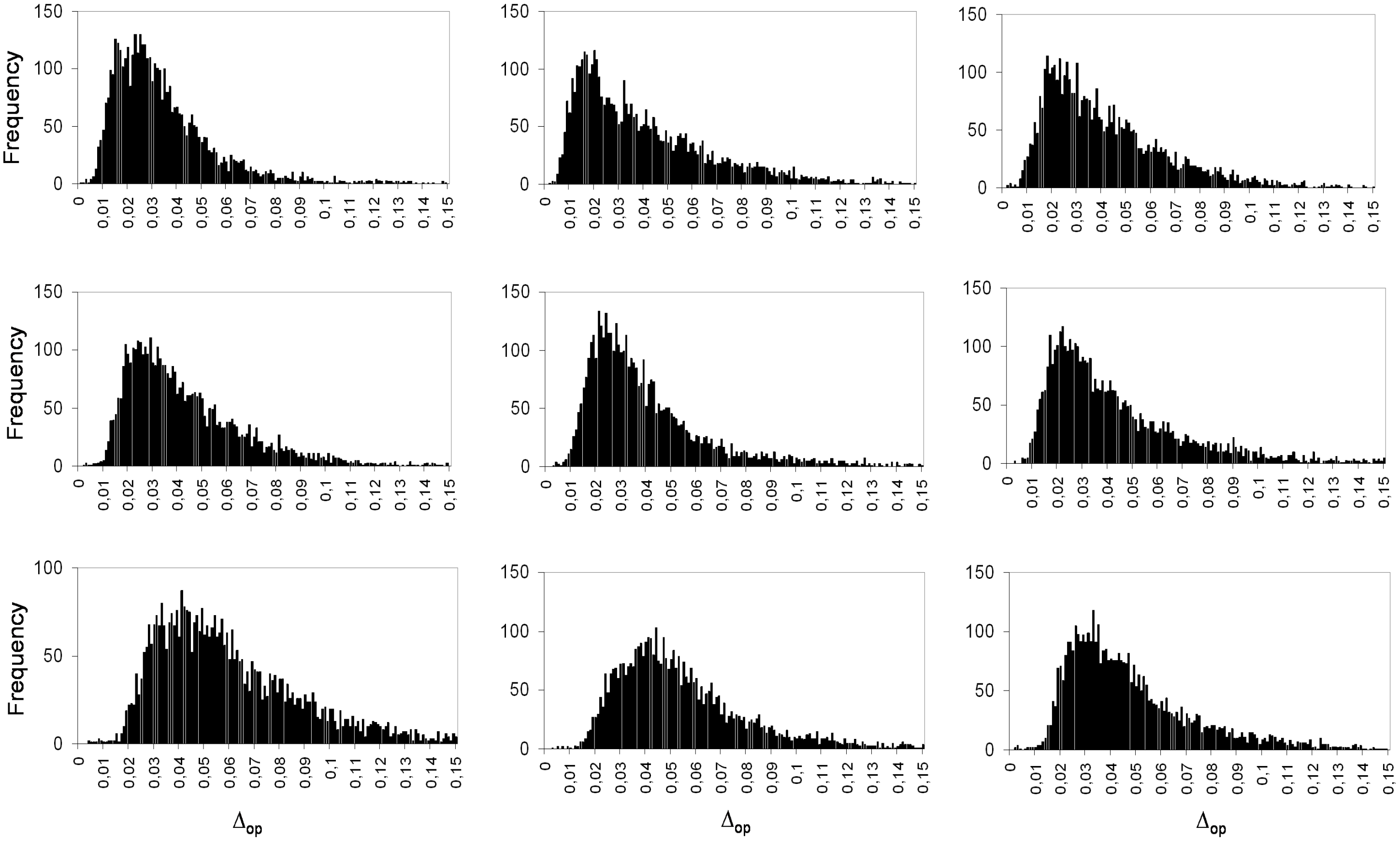


Fig. 13: Histograms of the expression errors in the operons (Δ_{op}) for each fold.

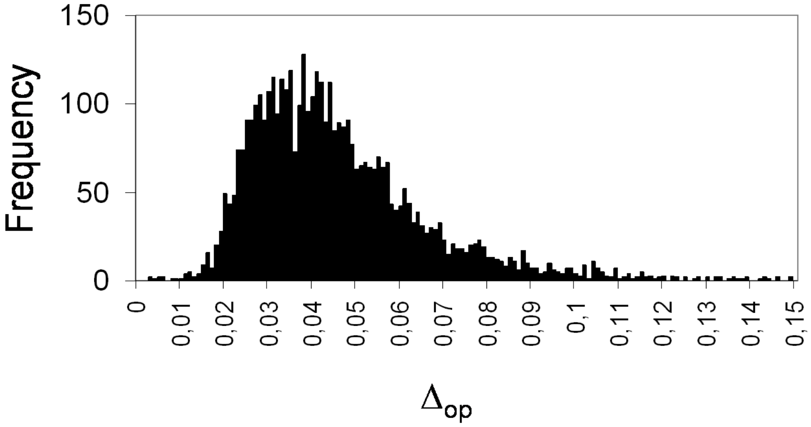
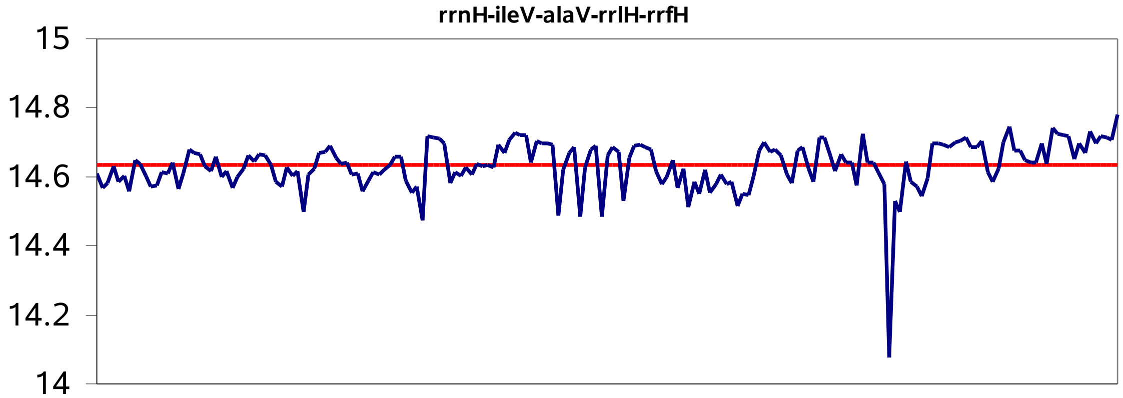


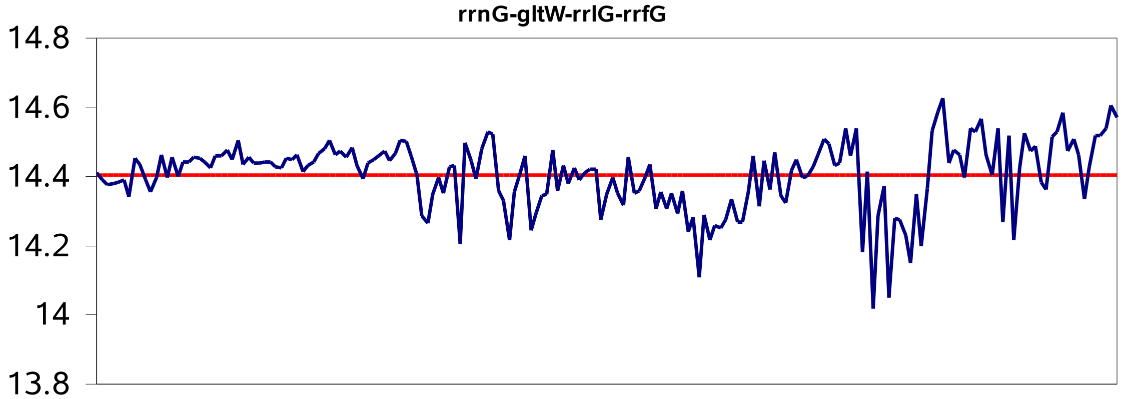
Fig. S14: Histogram of the asymptotic expression errors in the operons (Δ_{op}) computed by K-fold cross-validation (K=9).

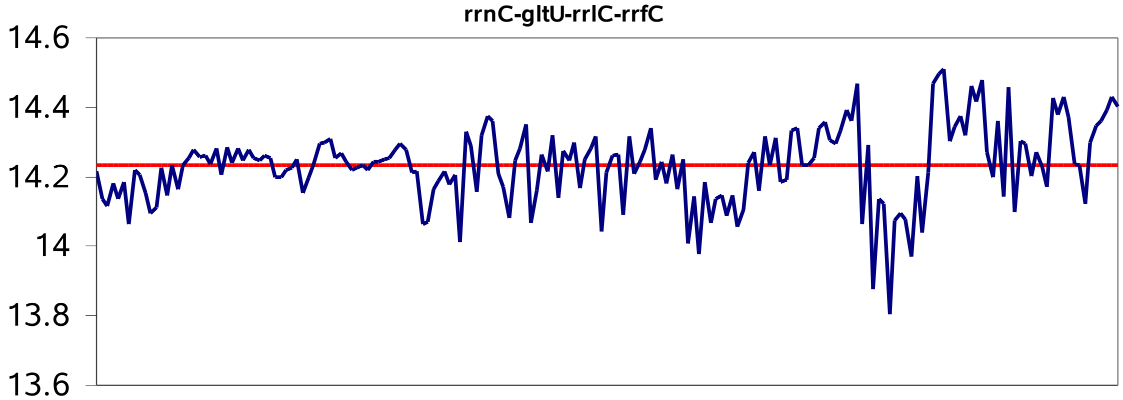
Prediction of expression profiles

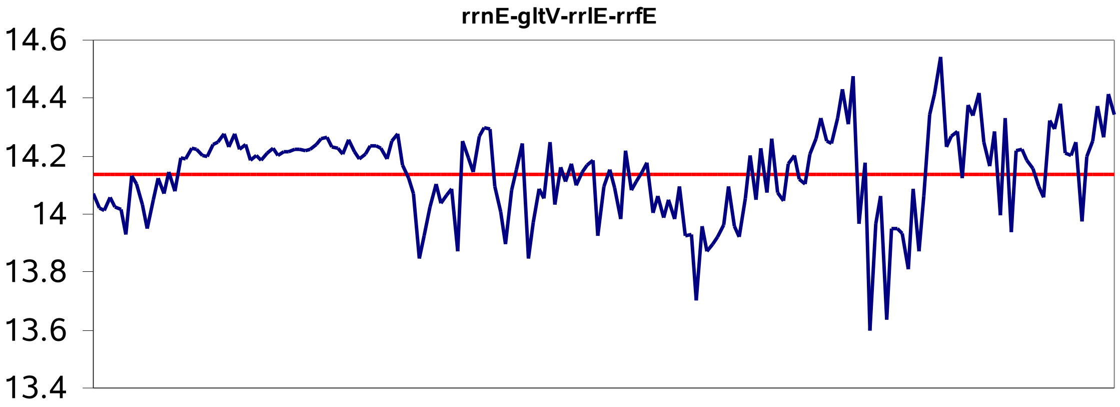
We have applied our kinetic model for *E. coli* to obtain the transcriptomic profiles under several experimental conditions (those included in the training set and the new perturbations). Each plot shows the experimental profile (blue line) and the profile by our model (red line) in the best predicted operons according to error based on the expression values (Δ_{op}).

Operons with constitutive promoters:


rrnH-ileV-alaV-rrlH-rrfH


rrnG-gltW-rrlG-rrfG


rrnC-gltU-rrlC-rrfC



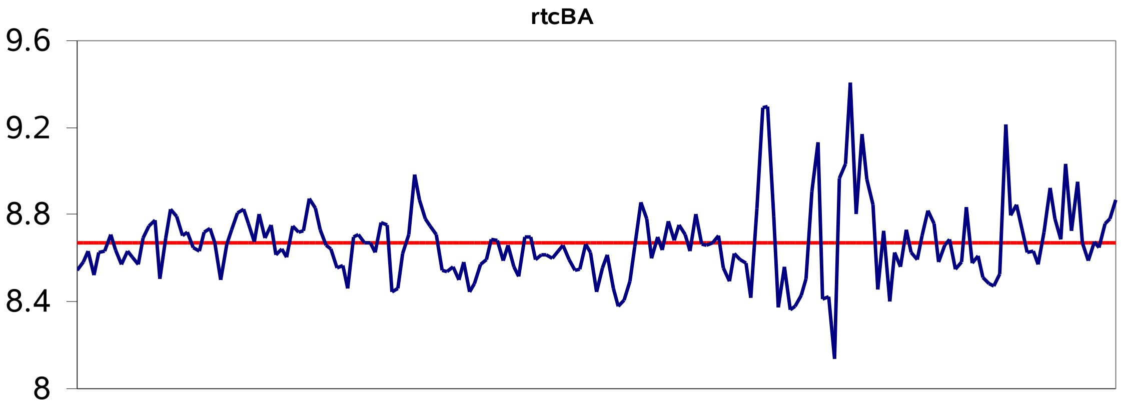
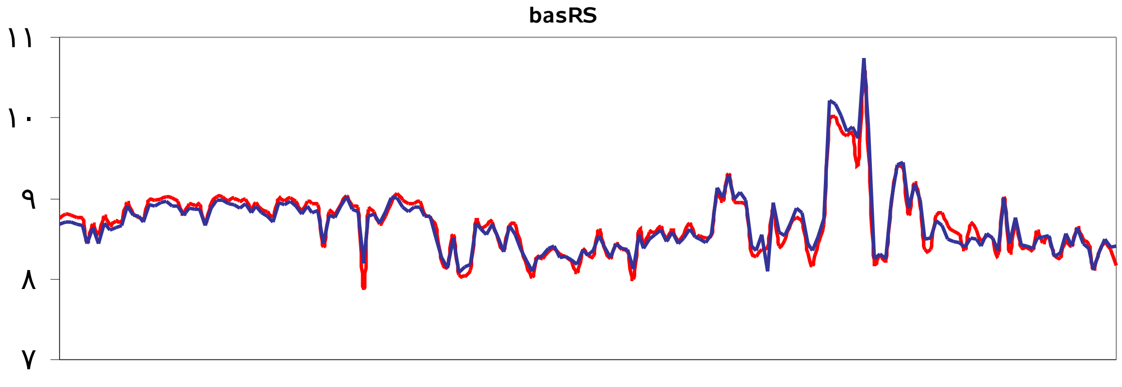


Fig. S15: Predicted profiles of operons with constitutive promoters (model in red, experiments in blue).

Operons with one-regulated promoters:



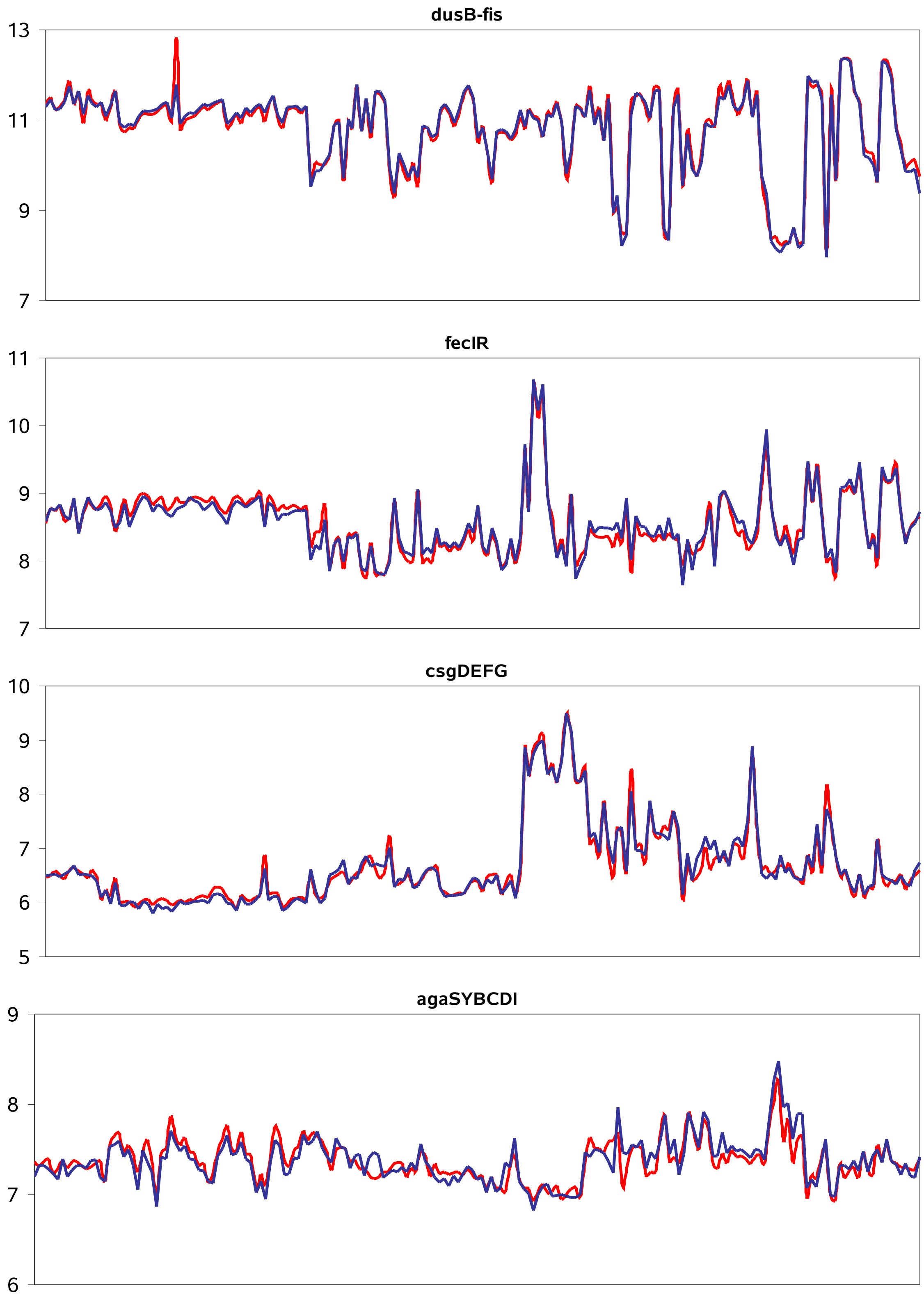
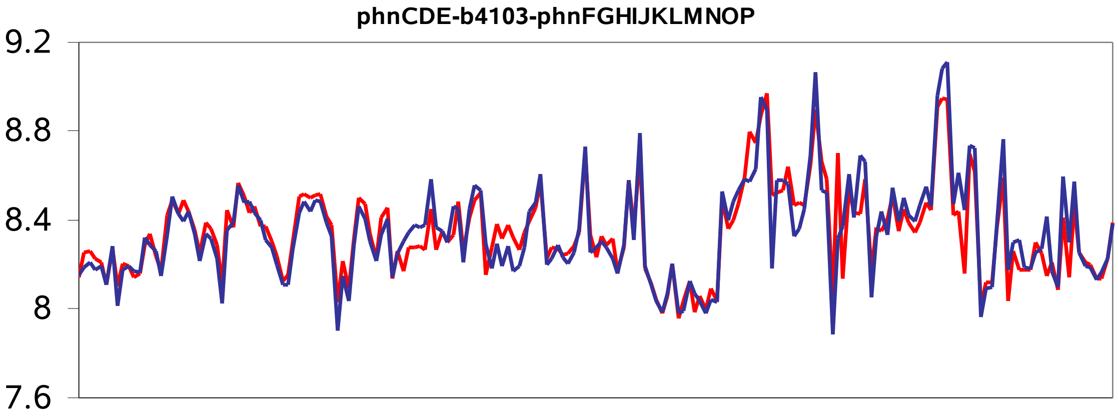
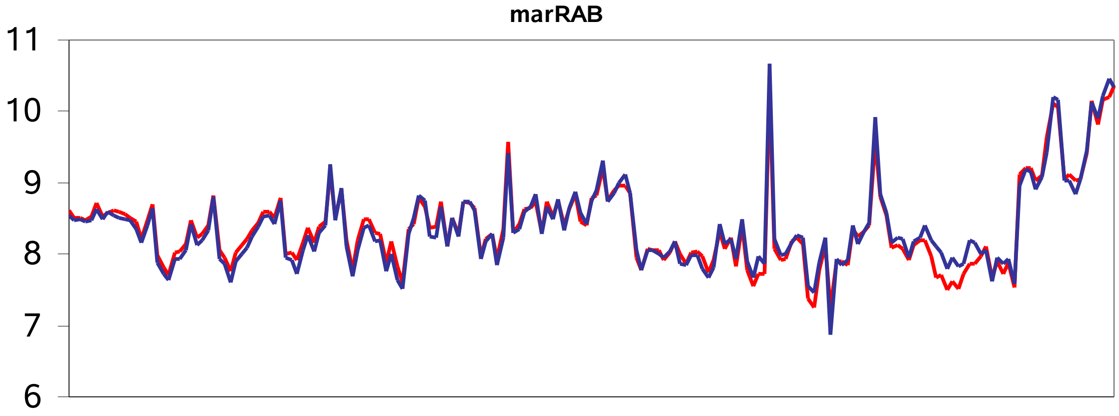
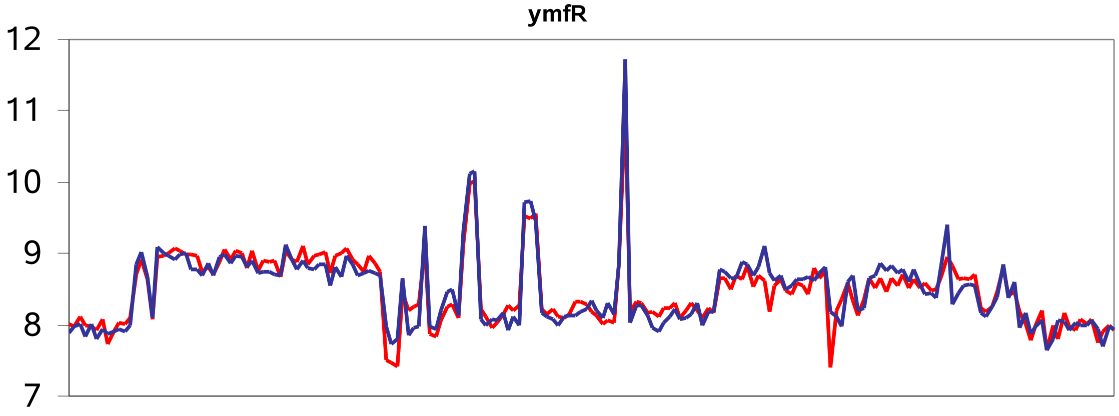


Fig. S16: Predicted profiles of operons with promoters by one TFs (model in red, experiments in blue).

Operons with two-regulated promoters:


phnCDE-b4103-phnFGHIJKLMNOP


marRAB


ymfR

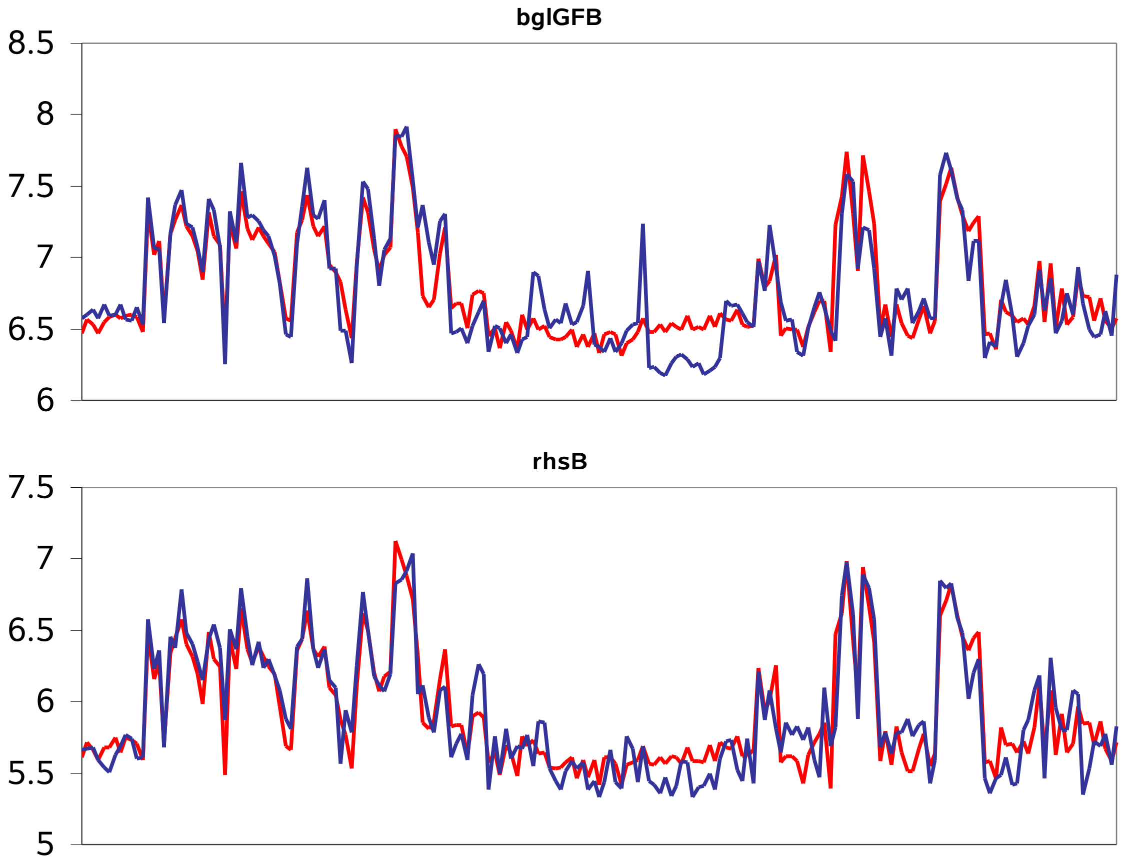
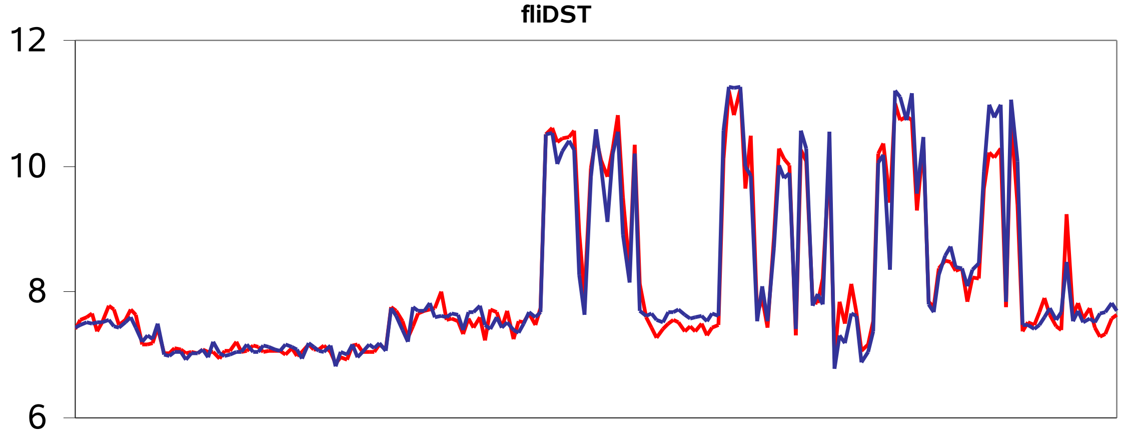


Fig. S17: Predicted profiles of operons with promoters regulated by two TFs (model in red, experiments in blue).

Operons with high-order promoters:



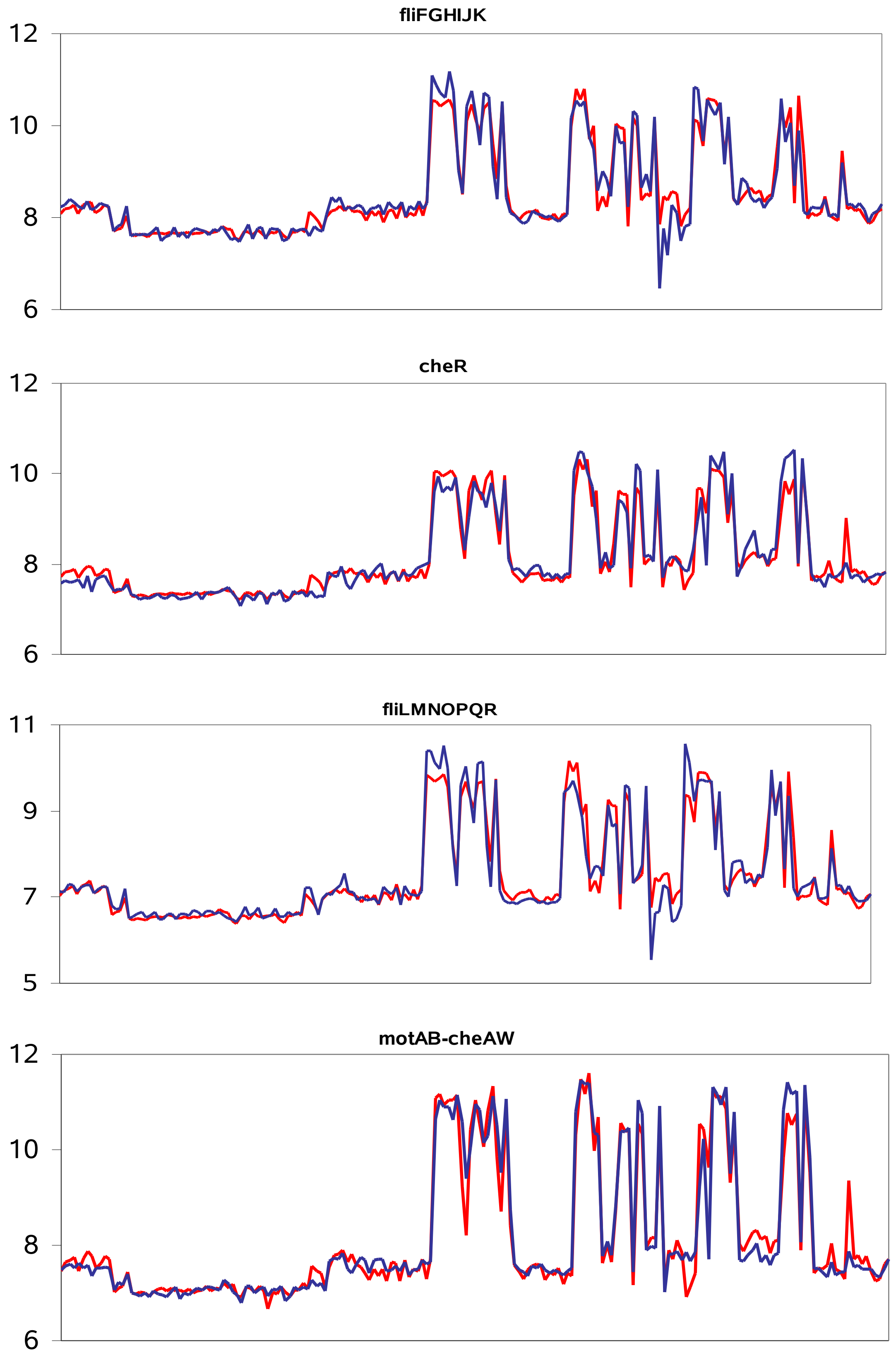


Fig. S18: Predicted profiles of operons with promoters regulated by three or more TFs (model in red, experiments in blue).

Redesign of global transcription regulation

Gene regulations allow organisms to adapt their cellular processes under external changes. In that way, one interesting question is to study the redesign of transcription regulations at the global level. For that, knockouts of transcription factors, especially if they are master regulators, constitutes a good starting point. In addition, we study the addition of new regulatory links in the genome network by putting together wild-type promoters and ORFs of TFs. For that, plasmids or chromosomal insertions can be used.

We have applied the model to predict the expression profiles under knockouts of TFs and transcriptional rewirings in *E. coli* [7] (see Fig. S19). For the TF knockouts, we use conditions from the training set, where the TFs *appY, crp, fnr, recA, arcA, cspA, oxyR, soxS,* and *hns* are perturbed. For simplicity and to work with a linear model, we infer a new model by neglecting the combinatorial regulations. In addition, we incorporate interactions between TFs from RegulonDB to improve the transcription regulatory core. This model also gives good results when predicting expression profiles. We impose in the model the knockout effect by removing the kinetic interaction of the corresponding TF. Then, we solve the linear system. In Fig. S21 we plot the predicted expression versus the experimental one (for the whole transcriptome and for the TFs) for all knockouts. We show how the model is able to reproduce the entire transcriptomic profile. We also study the effect on the cell of the knockout versus the wild-type. We show how some regulations (e.g., *appY*) are not necessary to maintain a given expression profile. However, the lack of other regulations, especially from transcription hubs (e.g., *crp*), confers to the cell a totally different expression profile (see Fig. S20).

On the other hand, we use Isalan *et al.* data [7] as an experimental validation for the prediction of rewired transcription. These recently published microarray data are in Affymetrix units. A rewiring is achieved by using a plasmid with wild-type promoters and TFs. Then, we can over-express wild-type promoters fused to TFs belonging to other operons. Those TFs control many genes. The resulting cell has a new regulatory map. We modify accordingly the model to account for such perturbations. We solve the system of equations given by the TF network to get the expressions for the regulators. Then, we obtain the whole transcriptomic profile. The two rewirings here considered are: 1) the promoter upstream of *malT* with the ORF of *fliA*, and 2) the promoter upstream of *rpoS* with the ORF of *ompR*. In Fig. S21 we plot the predicted expression versus the experimental one (for the whole transcriptome and for the TFs) for the two rewiring experiments. We show how the model is able to reproduce the entire transcriptomic profile. Notice that we have corrected the mean of the Isalan's data distribution to the one obtained from the training set data to use our model. In addition, the model has been bounded according to the Affymetrix scale in order to account for saturation effects in gene regulation.

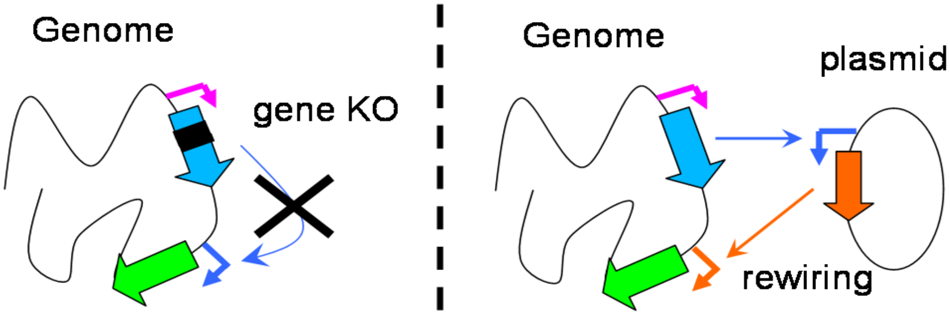


Fig. S19: Left: scheme of gene (TF) knockout. Right: scheme of transcription rewiring. Arrows represent transcription regulations.

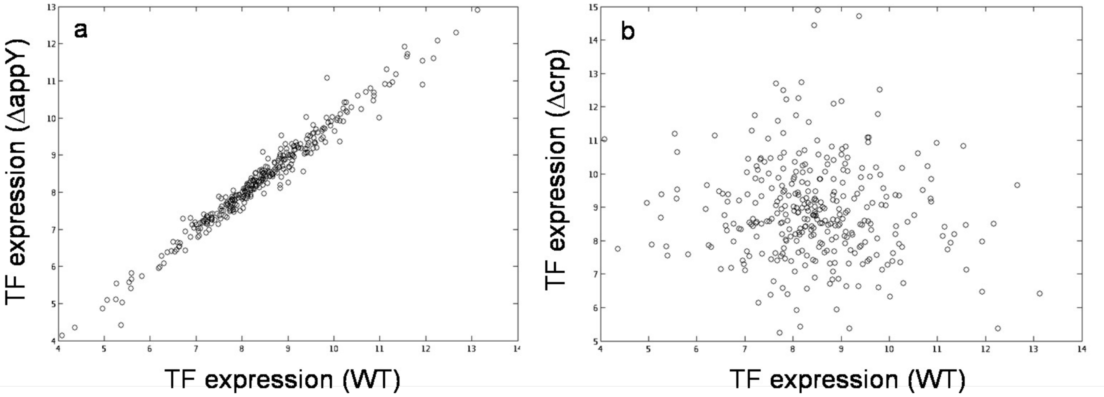
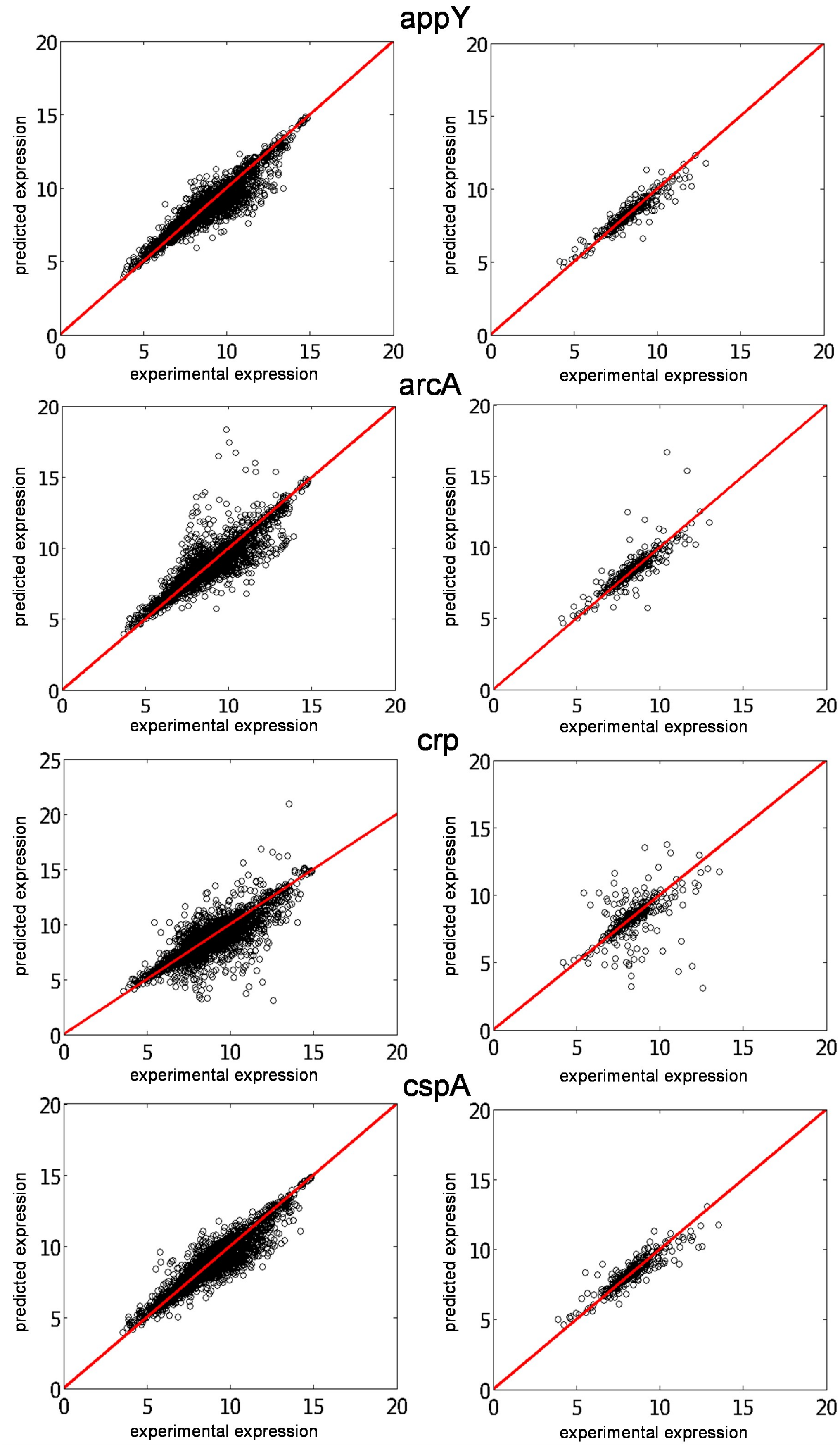


Fig S20: Plot of the experimental gene (TF) expression of knockouts versus the wild-type (WT). In (a) knockout of appY versus WT, in (b) knockout of crp versus WT.

The values of the relative expression error (Δop) are (first column: error computed using all genes, second column: error computed only using the TFs):

Knockout		
recA	0.037	0.040
appY	0.042	0.039
arcA	0.051	0.049
arcA, fnr	0.056	0.053
fnr	0.051	0.048
oxyR	0.050	0.044
oxyS	0.041	0.039
crp	0.062	0.089
cspA	0.048	0.047
hns	0.054	0.065
Rewiring		
Promotor=rpoS, ORF=ompR	0.178	0.162
Promotor=malT, ORF=fliA	0.187	0.163

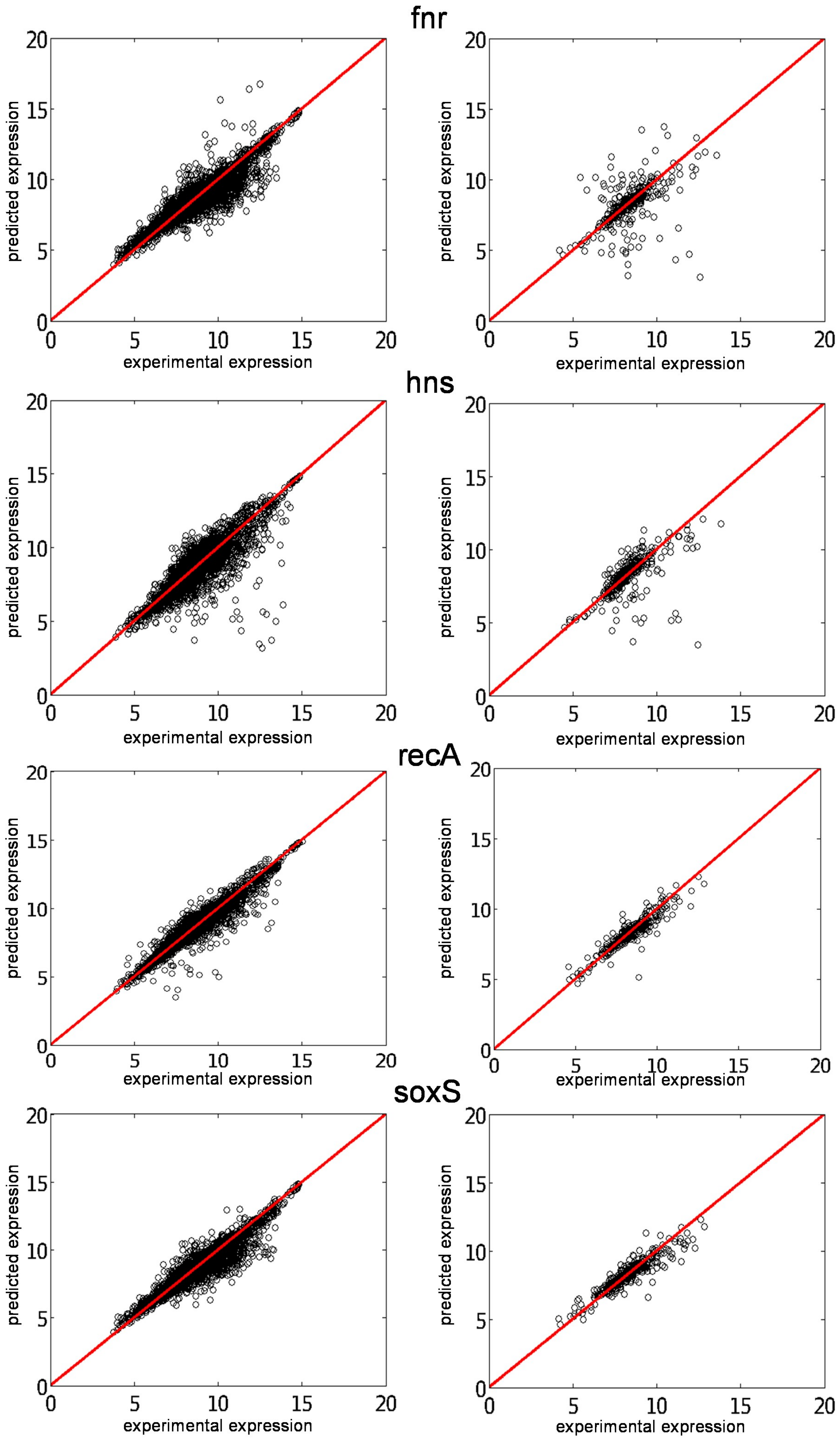


appY

arcA

crp

cspA


fnr
hns
recA
soxS
predicted expression
experimental expression
0
5
10
15
20

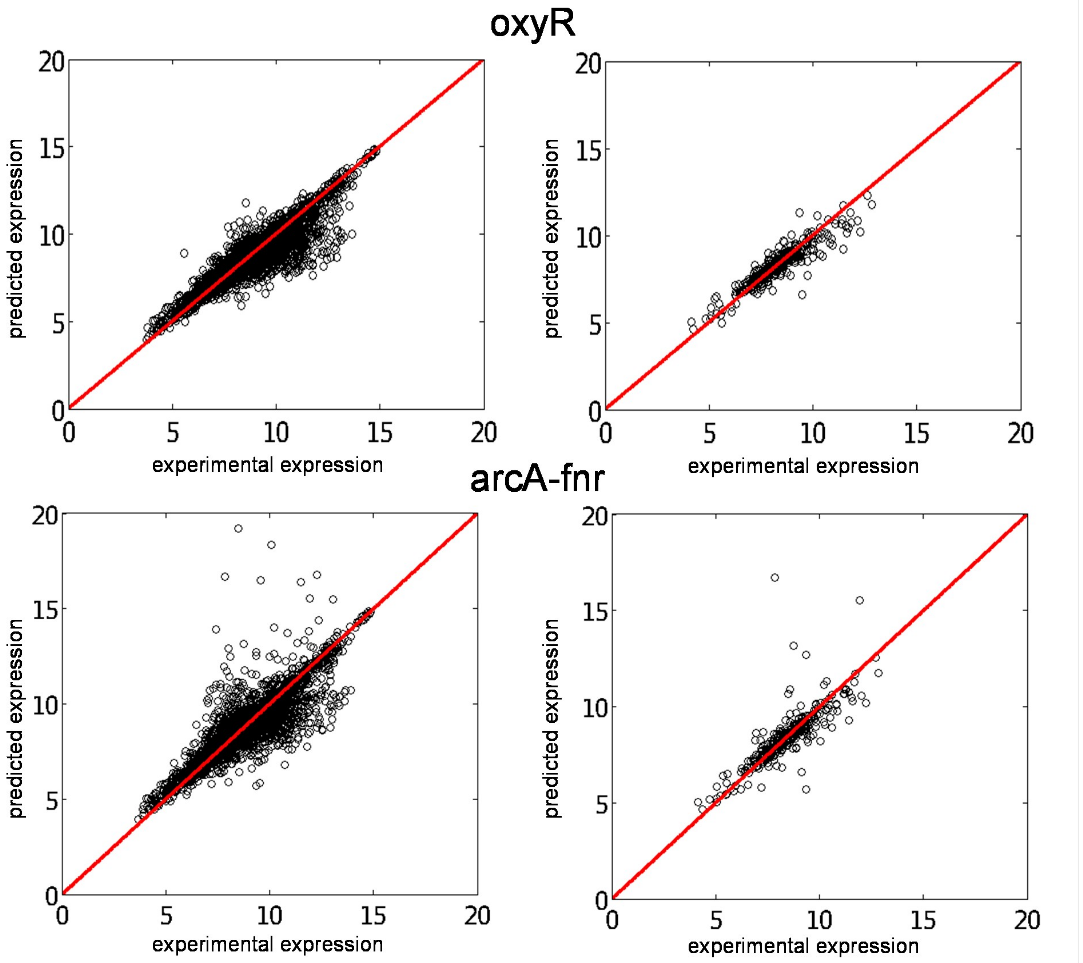


Fig. S21: Gene expression prediction versus experimental values [2] in knockouts of TFs. We solve the system of linear equations from the inferred model by removing the regulatory effect of the corresponding TF. Right: whole transcriptomic profile. Left: profile of TFs.

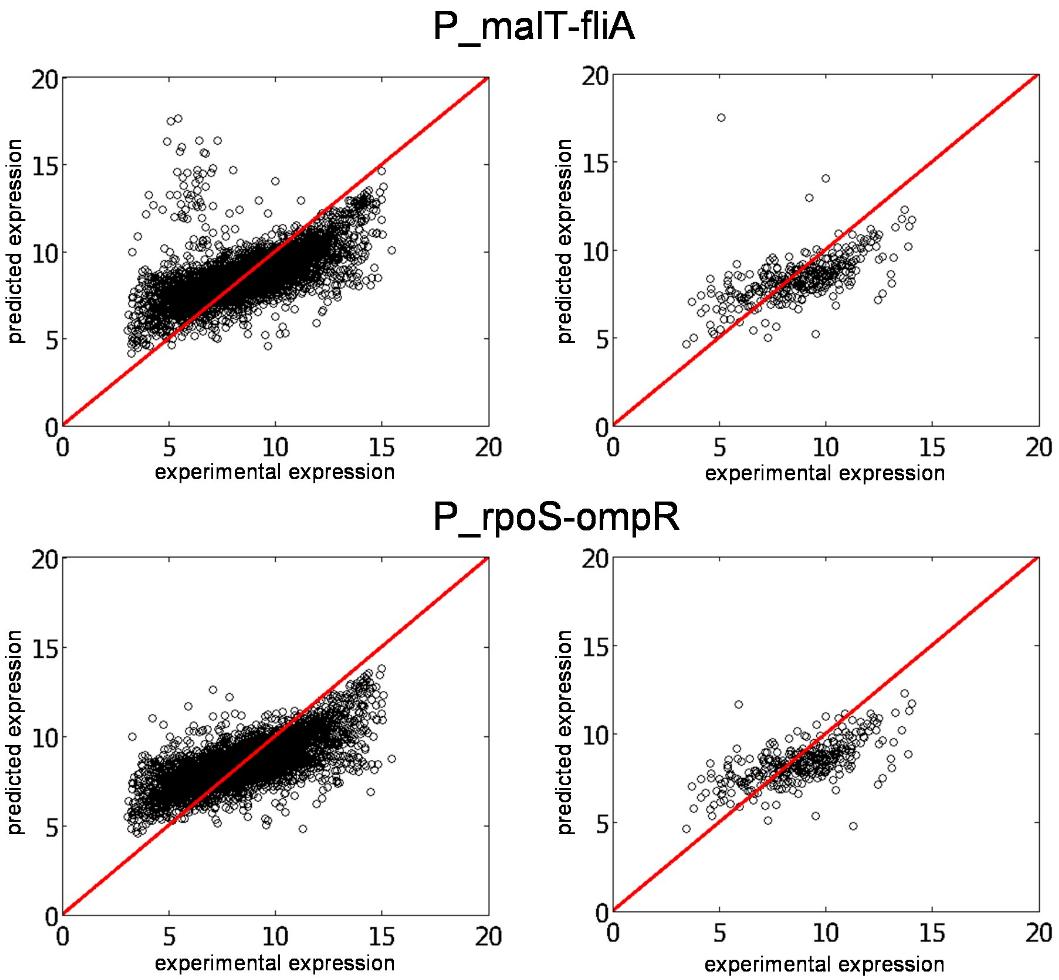


Fig. S22: Gene expression prediction versus experimental values [7] in rewired bacteria. We solve the system of linear equations from the inferred model by imposing the new regulatory effects of the corresponding TFs. Right: whole transcriptomic profile. Left: profile of TFs.

Non-transcriptional regulations

Our approach can also be applied to infer subnetwork models. In Gardner et al. 2003 [8], the NIR algorithm was developed and applied to the SOS pathway (an *E. coli* regulatory network of 9 genes in charge of DNA reparation). The algorithm generated a linear model from expression data. Here, we have applied our algorithm to the same system with the aim of comparing these two approaches. For that, we have used the same data set published there. We have applied InferGene to obtain two models of such system. The first model only assumes interactions between genes and TFs, and the second model (effective model) considers all genes as possible regulators.

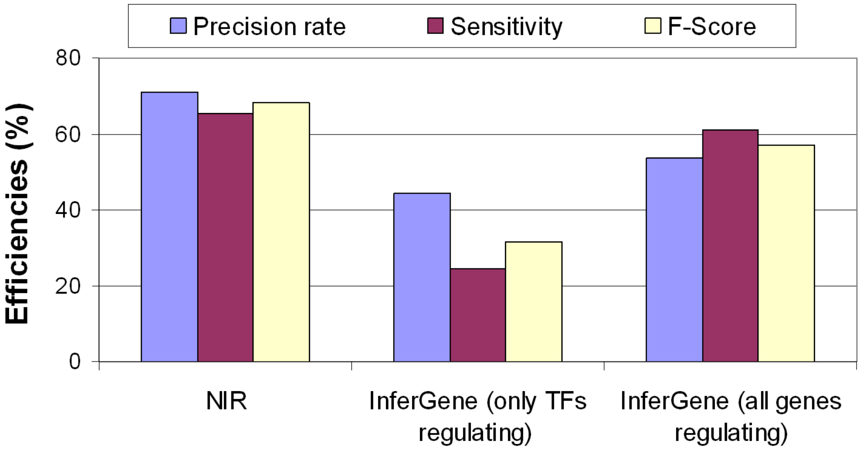


Fig. S23: Efficiencies (precision rate, sensitivity, and F-score) of the inferred topologies using NIR and InferGene. With InferGene we have obtained two networks: the first one takes just the pairs genes-TFs, and in the second model all genes can act as regulators.

In Fig. S23 we plot the efficiencies for the three inferred models. We show how the NIR algorithm captures lightly a better topology, and how the effective model by InferGene is more accurate than the pure transcriptional one. However, NIR requires the specification of the degree of connectivity. This fact can produce high inaccuracies in large-scale networks where the connectivity distribution is mostly scale-free, although it gives good results in small systems. Then, we applied the models to predict the expression profiles. In Fig. S24 we plot the relative expression errors in average for the three models. We can see how the effective model has lowest error in the prediction. In addition, in Fig. S25 we detail that errors by genes.

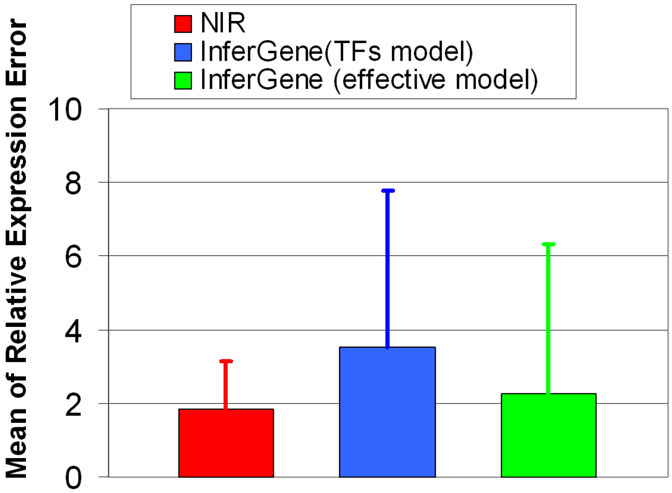


Fig. S24: Relative expression error in average for the SOS pathway using the inferred models from NIR and InferGene. Maximum errors are also plotted by means of bars.

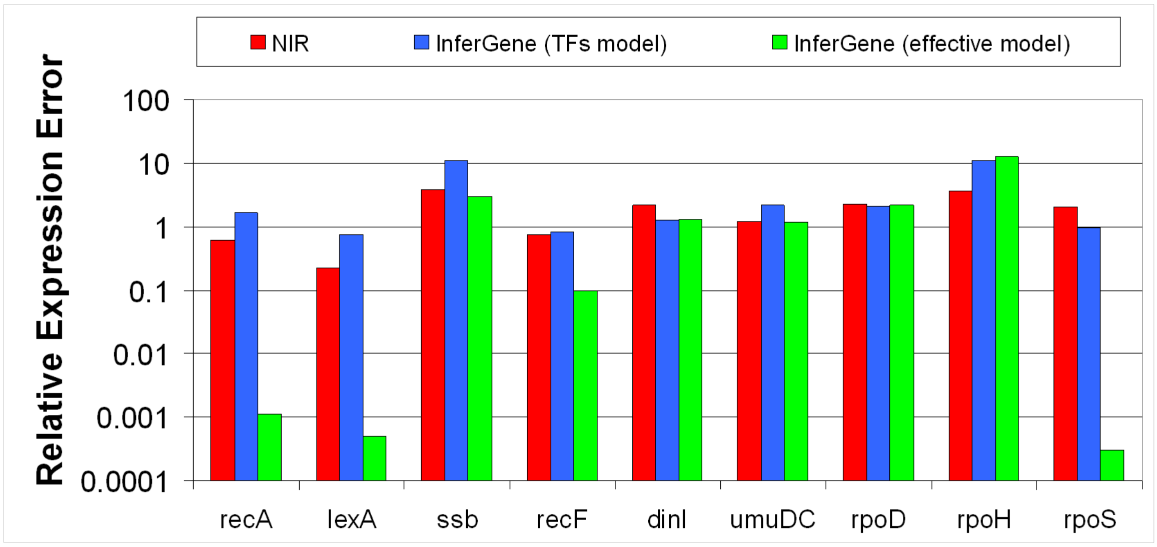


Fig. S25: Relative expression errors of genes from the SOS pathway using NIR and InferGene.

References

[1] Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. Plos Biol, 5, e8.

[2] Faith, J. J., Driscoll, M. E., Fusaro, V. A., Cosgrove, E. J., Hayete, B., Juhn, F. S., Schneider, S. J., and Gardner, T. S. (2008). Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. Nucleic Acids Res, 36, D866–D870.

[3] Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics, 4, 249–264.

[4] Salgado, H., Gama-Castro, S., Peralta-Gil, M., Diaz-Peredo, E., Sanchez-Solano, F., Santos-Zavaleta, A., Martinez-Flores, I., Jimenez-Jacinto, V., Bonavides-Martinez, C., Segura-Salazar, J., Martinez-Antonio, A., and Collado-Vides, J. (2006). RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. Nucleic Acids Res, 34, D394.

[5] Hucka, M., et al. (2003). The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics, 19, 524–531.

[6] Karp, P. D., Riley, M., Saier, M., Paulsen, I. T., Collado-Vides, J., Paley, S. M., Pellegrini-Toole, A., Bonavides, C., and Gama-Castro, S. (2002). The EcoCyc DataBase. Nuclear Acids Res, 30, 56–58.

[7] Isalan, M., Lemerle, C., Michalodimitrakis, K., Horn, C., Beltrao, P., Raineri, E., Garriga-Canut, M., Serrano, L. (2008) Evolvability and hierarchy in rewired bacterial gene networks. Nature 452, 840-845.

[8] Gardner, T. S., di Bernardo, D., Lorenz, D., and Collins, J. J. (2003) Inferring genetic networks and identifying compound mode of action via expression profiles. Science, 301, 102-105.