

V10 Topologies and Dynamics of Gene Regulatory Networks

Who are the players in GRNs? SILAC technology

What are the kinetic rates?

DREAM3 contest for network reconstruction

Algorithm by team of Mark Gerstein

Rates of mRNA transcription and protein translation

ARTICLE

doi:10.1038/nature10098

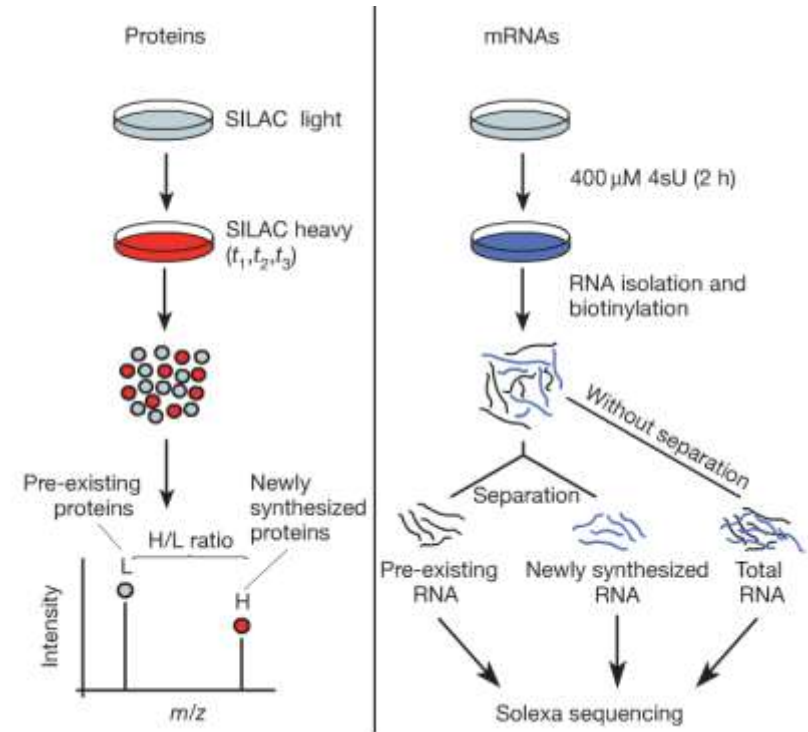
Global quantification of mammalian gene expression control

Björn Schwanhäusser¹, Dorothea Busse¹, Na Li¹, Gunnar Dittmar¹, Johannes Schuchhardt², Jana Wolf⁶, Wei Chen¹ & Matthias Selbach¹

SILAC: „stable isotope labelling by amino acids in cell culture“ means that cells are cultivated in a medium containing heavy stable-isotope versions of essential amino acids.

When non-labelled (i.e. light) cells are transferred to heavy SILAC growth medium, newly synthesized proteins incorporate the heavy label while pre-existing proteins remain in the light form.

Schwanhäuser et al. Nature 473, 337 (2011)



Parallel quantification of mRNA and protein turnover and levels. Mouse fibroblasts were pulse-labelled with heavy amino acids (SILAC, left) and the nucleoside 4-thiouridine (4sU, right). Protein and mRNA turnover is quantified by mass spectrometry and next-generation sequencing, respectively.

Rates of mRNA transcription and protein translation

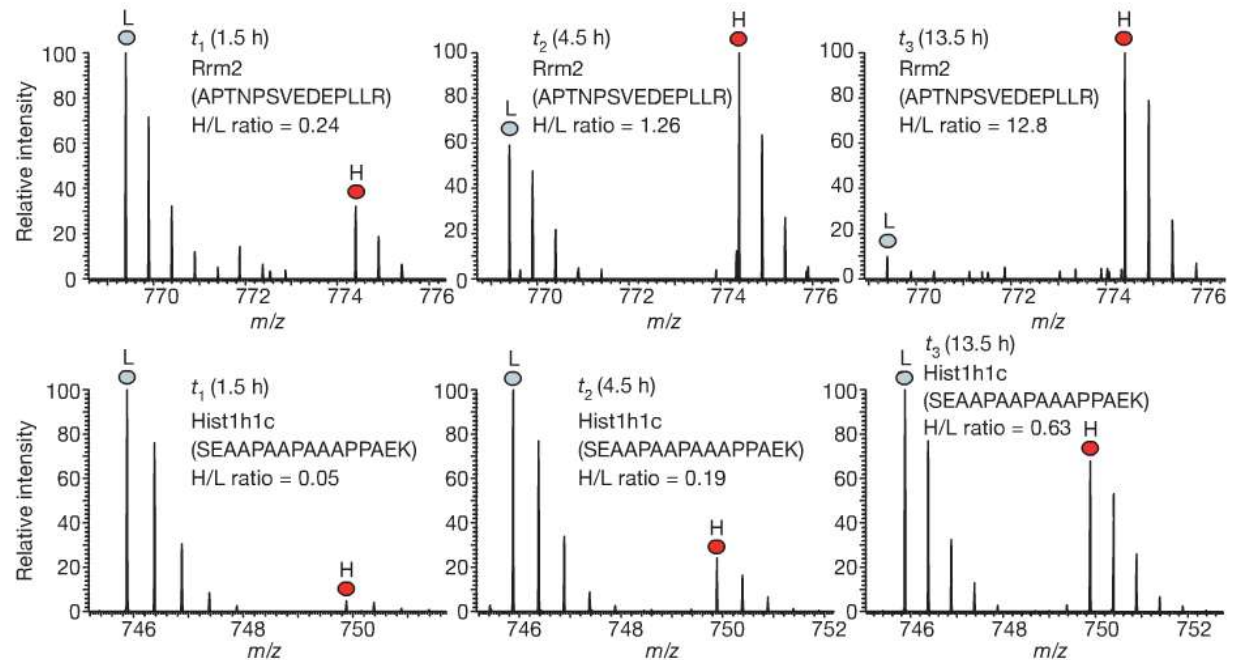
84,676 peptide sequences were identified by MS and assigned to 6,445 unique proteins.

5,279 of these proteins were quantified by at least three heavy to light (H/L) peptide ratios

Mass spectra of peptides for two proteins.

Top: high-turnover protein
Bottom: low-turnover protein.

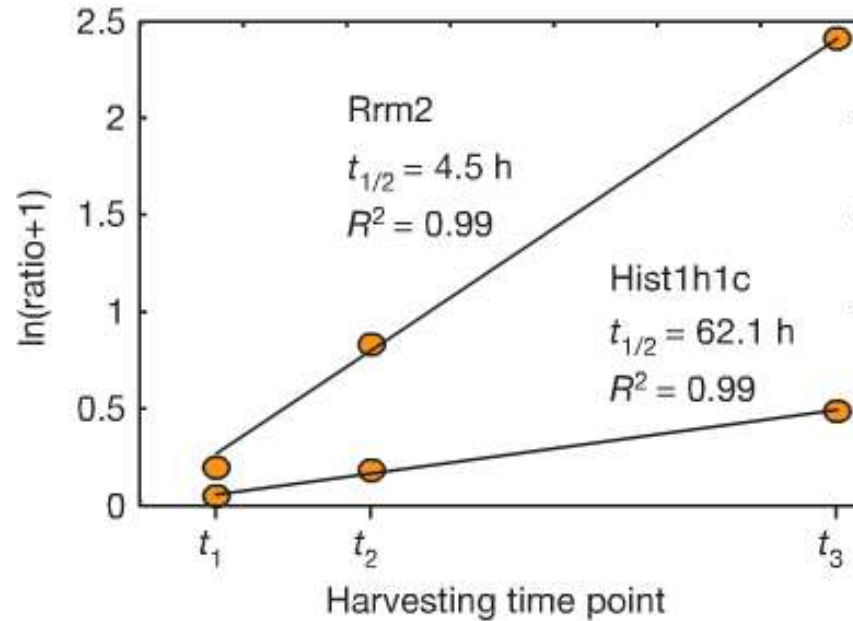
Over time, the heavy to light (H/L) ratios increase.



Schwanhäuser et al. Nature 473, 337 (2011)

Protein half-lives

Protein half-lives were calculated from log H/L ratios at all three time points using linear regression.



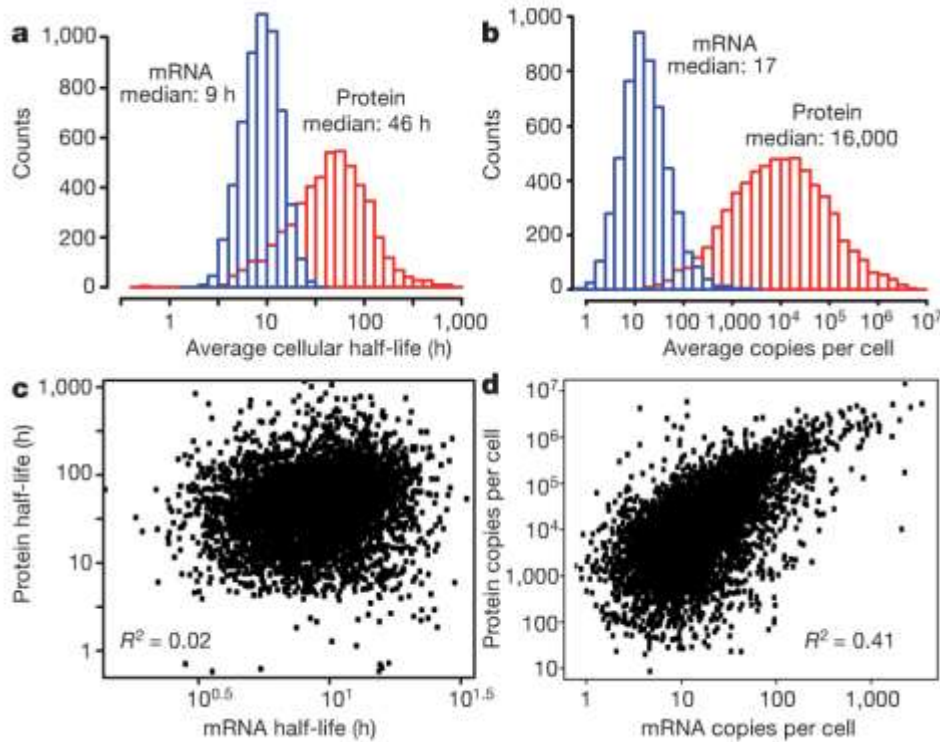
The same is done to compute mRNA half-lives (not shown).

Schwanhäuser et al. Nature 473, 337 (2011)

mRNA and protein levels and half-lives

a, b, Histograms of mRNA (blue) and protein (red) half-lives (a) and levels (b).

Proteins were on average 5 times more stable (9h vs. 46h) and 900 times more abundant than mRNAs and spanned a higher dynamic range.



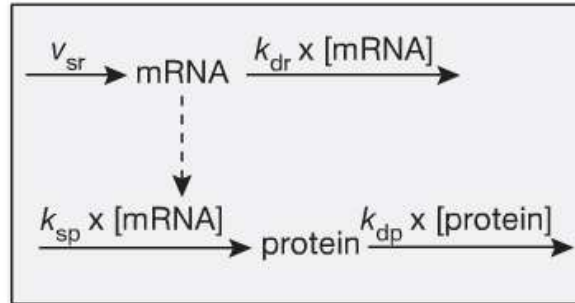
c, d, Although mRNA and protein levels correlated significantly, correlation of half-lives was virtually absent

Schwanhäuser et al. Nature 473, 337 (2011)

Mathematical model

a

A widely used minimal description of the dynamics of transcription and translation includes the synthesis and degradation of mRNA and protein, respectively



$$\frac{dR}{dt} = v_{sr} - k_{dr}R$$

$$\frac{dP}{dt} = k_{sp}R - k_{dp}P$$

The mRNA (R) is synthesized with a constant rate v_{sr} and degraded proportional to their numbers with rate constant k_{dr} .

The protein level (P) depends on the number of mRNAs, which are translated with rate constant k_{sp} .

Protein degradation is characterized by the rate constant k_{dp} .

The synthesis rates of mRNA and protein are calculated from their measured half lives and levels

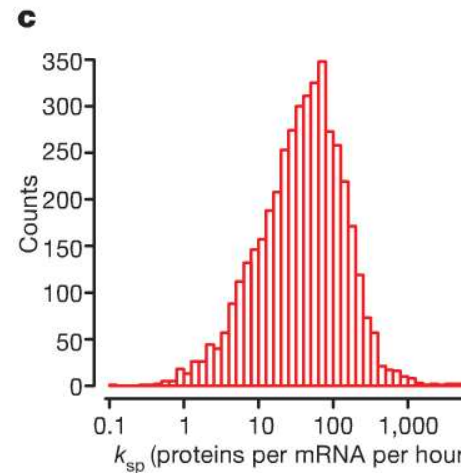
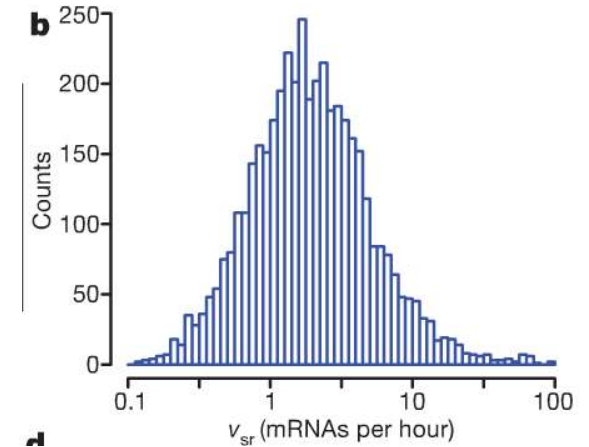
Schwanhäuser et al. Nature 473, 337 (2011)

Computed transcription and translation rates

Average cellular transcription rates predicted by the model spanned two orders of magnitude.

The median is about 2 mRNA molecules per hour (b). An extreme example is Mdm2 with more than 500 mRNAs per hour

The median translation rate constant is about 40 proteins per mRNA per hour



Calculated translation rate constants are not uniform

Schwanhäuser et al. Nature 473, 337 (2011) Bioinformatics III

Maximal translation constant

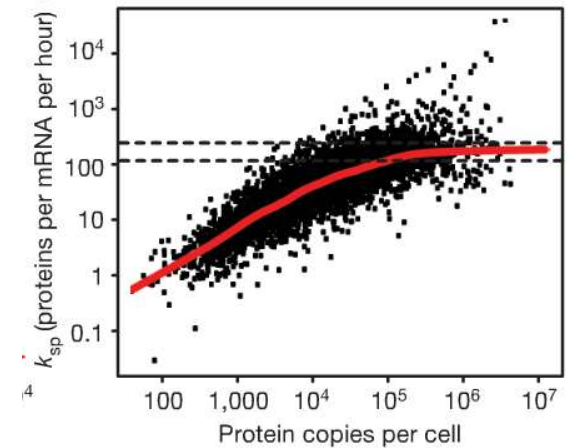
Abundant proteins are translated about 100 times more efficiently than those of low abundance

Translation rate constants of abundant proteins saturate between approximately 120 and 240 proteins per mRNA per hour.

The maximal translation rate constant in mammals is not known.

The estimated maximal translation rate constant in sea urchin embryos is 140 copies per mRNA per hour, which is surprisingly close to the prediction of this model.

Schwanhäuser et al. Nature 473, 337 (2011)



Mathematical reconstruction of Gene Regulatory Networks

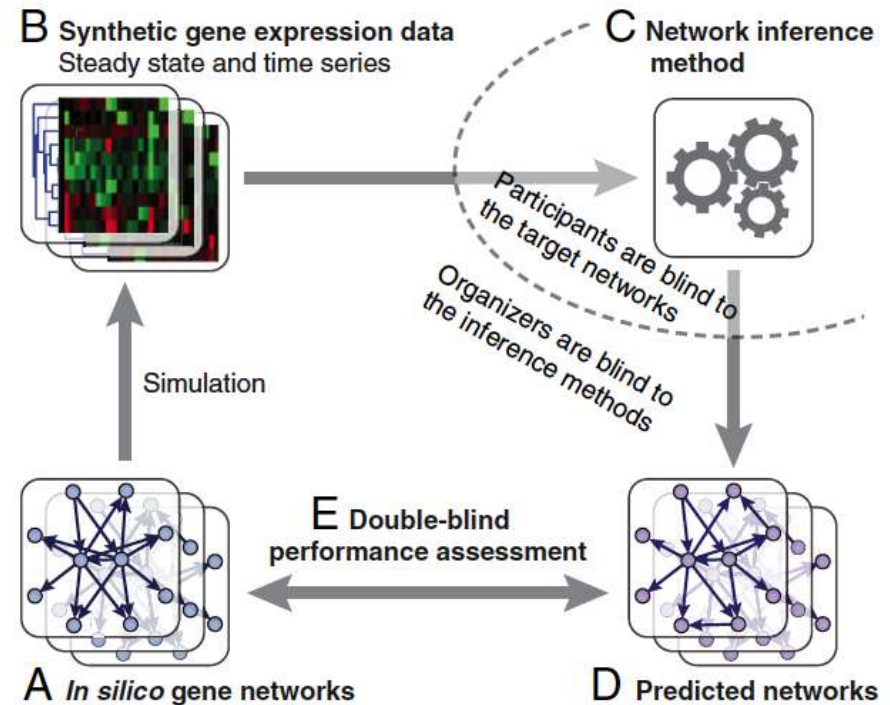
DREAM: **D**ialogue on **R**everse **E**ngineering
Assessment and **M**ethods

Aim:
systematic evaluation of methods for
reverse engineering of network topologies
(also termed network-inference methods).

Problem:
correct answer is typically not known for real
biological networks

Approach:
generate synthetic data

Marbach et al. PNAS 107, 6286 (2010)



Gustavo Stolovitzky/IBM

Generation of Synthetic Data

Transcriptional regulatory networks are modelled consisting of genes, mRNA, and proteins.

The state of the network is given by the vector of mRNA concentrations x and protein concentrations y .

We model only transcriptional regulation, where regulatory proteins (TFs) control the transcription rate (activation) of genes (no epigenetics, microRNAs etc.).

The gene network is modeled by a system of differential equations

$$\frac{dx_i}{dt} = m_i \cdot f_i(\mathbf{y}) - \lambda_i^{\text{RNA}} \cdot x_i$$

$$\frac{dy_i}{dt} = r_i \cdot x_i - \lambda_i^{\text{Prot}} \cdot y_i,$$

where m_i is the maximum transcription rate, r_i the translation rate, λ_i^{RNA} and λ_i^{Prot} are the mRNA and protein degradation rates and $f_i(\cdot)$ is the so-called **input function** of gene i .

Marbach et al. PNAS 107, 6286 (2010)

The input function $f_i()$

The input function describes the relative activation of the gene, which is between 0 (the gene is shut off) and 1 (the gene is maximally activated), given the transcription-factor (TF) concentrations \mathbf{y} .

We assume that binding of TFs to cis-regulatory sites on the DNA is in quasi-equilibrium, since it is orders of magnitudes faster than transcription and translation.

In the most simple case, a gene i is regulated by a single TF j . In this case, its promoter has only two states: either the TF is bound (state S_1) or it is not bound (state S_0).

The probability $P(S_1)$ that the gene i is in state S_1 at a particular moment is given by the *fractional saturation*, which depends on the TF concentration y_j

$$P\{S_1\} = \frac{\chi_j}{1 + \chi_j} \quad \text{with} \quad \chi_j = \left(\frac{y_j}{k_{ij}} \right)^{n_{ij}}$$

where k_{ij} is the dissociation constant and n_{ij} the Hill coefficient (formula not derived here).

Marbach et al. PNAS 107, 6286 (2010)

The input function $f_i()$

$$P\{S_1\} = \frac{\chi_j}{1 + \chi_j} \quad \text{with} \quad \chi_j = \left(\frac{y_j}{k_{ij}}\right)^{n_{ij}}$$

$P(S_1)$ is large if the concentration of the TF j is large and if the dissociation constant is small (strong binding).

The bound TF activates or represses the expression of the gene. In state S_0 the relative activation is α_0 and in state S_1 it is α_1 .

Given $P(S_1)$ and its complement $P(S_0)$, the input function $f_i(y_j)$ is obtained, which computes the mean activation of gene i as a function of the TF concentration y_j

$$f(y_j) = \alpha_0 P\{S_0\} + \alpha_1 P\{S_1\} = \frac{\alpha_0 + \alpha_1 \chi_j}{1 + \chi_j}$$

Marbach et al. PNAS 107, 6286 (2010)

The input function $f_i()$

This approach can be used for an arbitrary number of regulatory inputs.

A gene that is controlled by N TFs has 2^N states: each of the TFs can be bound or not bound.

Thus, the input function for N regulators would be

$$f(\mathbf{y}) = \sum_{m=0}^{2^N-1} \alpha_m P\{S_m\}$$

Marbach et al. PNAS 107, 6286 (2010)

Synthetic gene expression data

Gene knockouts were simulated by setting the maximum transcription rate of the deleted gene to zero, knockdowns by dividing it by two.

Time-series experiments were simulated by integrating the networks using different initial conditions.

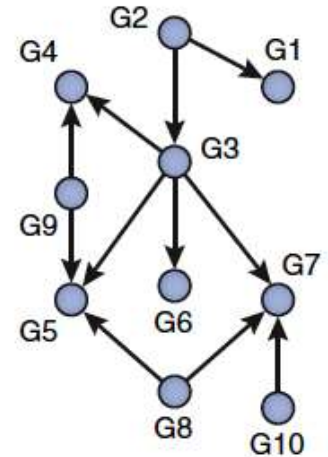
For the networks of size 10, 50, and 100, they provided 4, 23, and 46 different time series, respectively. For each time series, a different random initial condition was used for the mRNA and protein concentrations.

Each time series consisted of 21 time points.

Trajectories were obtained by integrating the networks from the given initial conditions using a Runge-Kutta solver.

White noise with a standard deviation of 0.05 was added after the simulation to the generated gene expression data.

Marbach et al. PNAS 107, 6286 (2010)

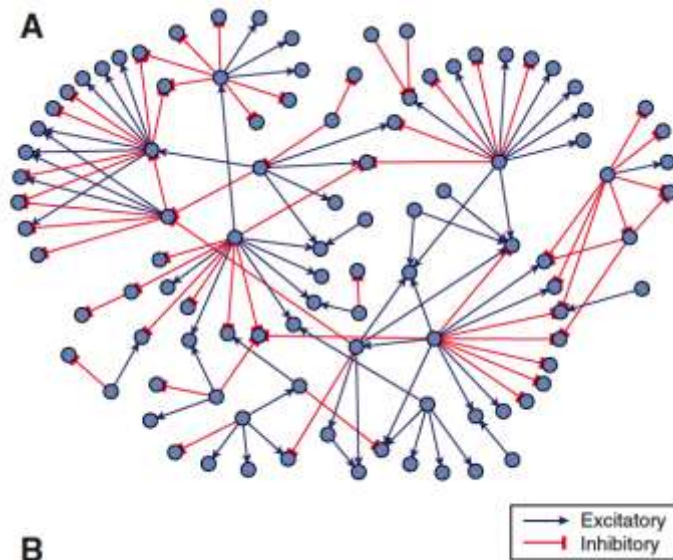


Synthetic networks

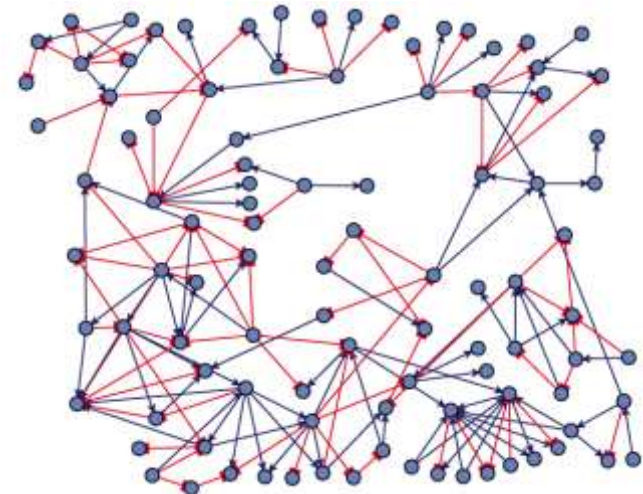
The challenge was structured as three separate subchallenges with networks of 10, 50, and 100 genes, respectively. For each size, five in silico networks were generated.

These resembled realistic network structures by extracting modules from known transcriptional regulatory network for *Escherichia coli* (2x) and for yeast (3x).

Example network *E.coli*



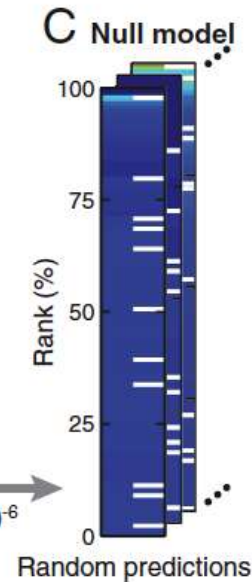
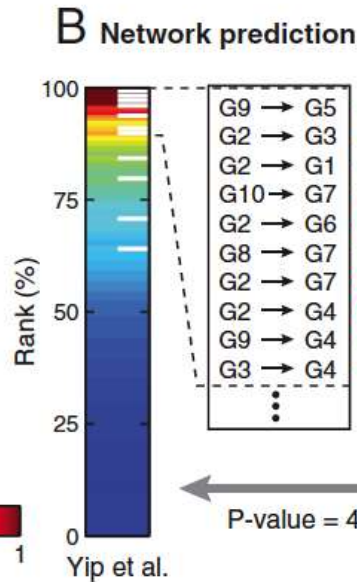
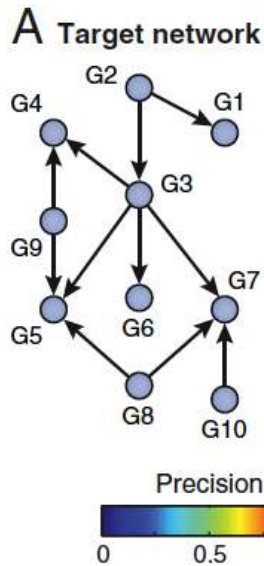
Example network yeast



Marbach et al. PNAS 107, 6286 (2010)

Evaluation of network predictions

(A) The true connectivity of one of the benchmark networks of size 10.



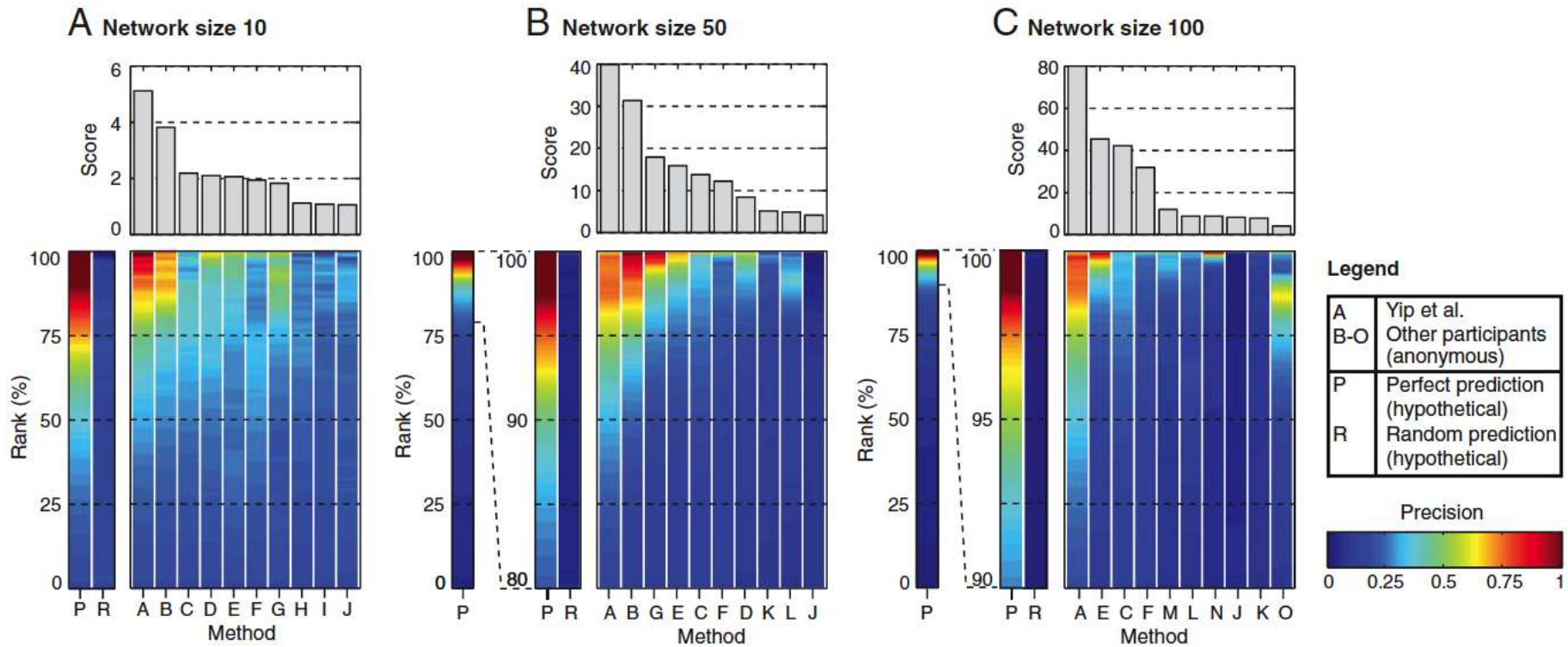
(C) The network prediction is evaluated by computing a P-value that indicates its statistical significance compared to random network predictions.

(B) Example of a prediction by the best-performer team. The format is a ranked list of predicted edges, represented here by the vertical colored bar. The white stripes indicate the true edges of the target network. A perfect prediction would have all white stripes at the top of the list.

Inset shows the first 10 predicted edges: the top 4 are correct, followed by an incorrect prediction, etc. The color indicates the precision at that point in the list. E.g., after the first 10 predictions, the precision is 0.7 (7 correct predictions out of 10 predictions).

Marbach et al. PNAS 107, 6286 (2010)


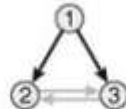

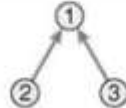
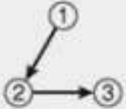
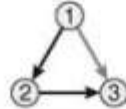
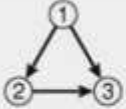
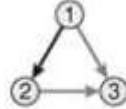
Similar performance on different network sizes







The method by Yip et al. gave the best results for all 3 network sizes.

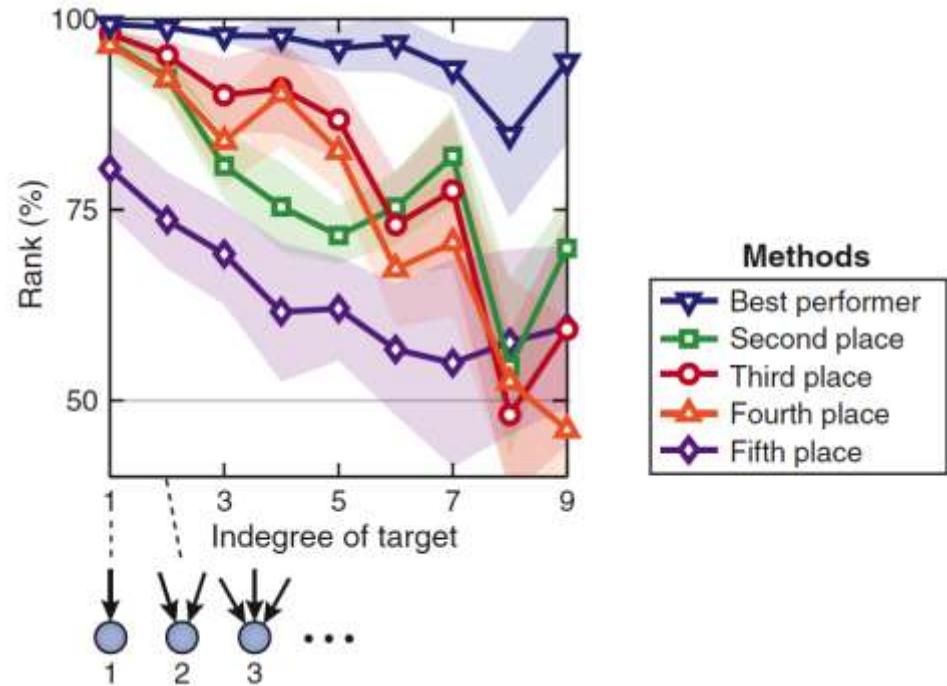
Marbach et al. PNAS 107, 6286 (2010)

Error analysis

	(A) True structure	(B) Prediction confidence	(C) Systematic prediction error
Fan-out			Fan-out error Incorrect prediction of links between co-regulated nodes (co-regulation misinterpreted as interaction)
Fan-in			Fan-in error Reduced prediction confidence for multiple inputs (difficulties in predicting combinatorial regulation)
Cascade			Cascade error Incorrect prediction of "shortcuts" (indirect interaction misinterpreted as direct interaction)
FFL			Feed-Forward Loop (FFL) Same type of error as in fan-in

 70-80%
 60-70%
 50-60%
 No arrow: < 50%

Median prediction confidence



Left: 3 typical errors made in predicted networks.

We will now discuss the best-performing method by Yip et al.

Only this method gives stable results independent of the indegree of the target (right)

Marbach et al. PNAS 107, 6286 (2010)

Improved Reconstruction of *In Silico* Gene Regulatory Networks by Integrating Knockout and Perturbation Data

Kevin Y. Yip¹, Roger P. Alexander², Koon-Kiu Yan², Mark Gerstein^{1,2,3*}

¹ Department of Computer Science, Yale University, New Haven, Connecticut, United States of America, ² Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut, United States of America, ³ Program in Computational Biology and Bioinformatics, Yale University, New Haven, Connecticut, United States of America

Best performing team in DREAM3 contest

Applied a simple noise model and linear and sigmoidal ODE models.

Predictions from the 3 models were combined.



Mark Gerstein/Yale

Yip et al. PLoS ONE 5:e8121 (2010)

Cumulative distribution function

The **cumulative distribution function** (CDF) describes the probability that a real-valued random variable X with a given probability distribution will be found at a value less than or equal to x .

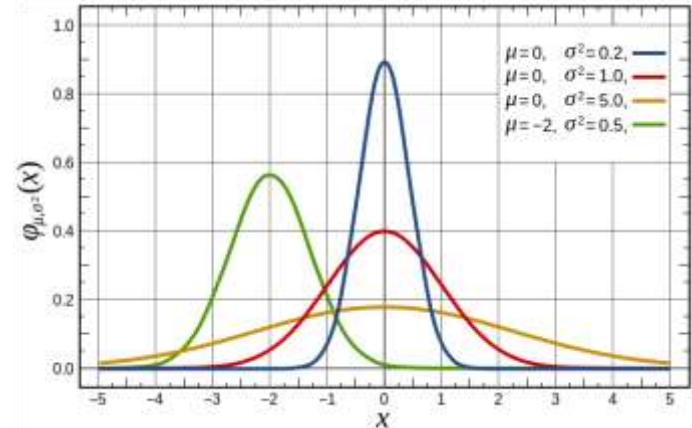
$$F_X(x) = P(X \leq x),$$

$$F(x) = \int_{-\infty}^x f(t) dt.$$

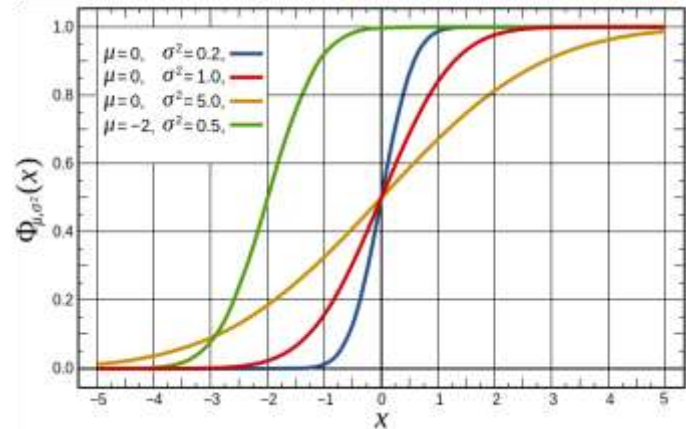
The complementary cumulative distribution function (ccdf) or simply the tail distribution addresses the opposite question and asks how often the random variable is *above* a particular level. It is defined as

$$\bar{F}(x) = P(X > x) = 1 - F(x).$$

www.wikipedia.org



Different normal distributions



CDF of the normal distribution

Noise model

If we were given:

x_a^b : observed expression level of gene a in deletion strain of gene b , and

x_a^{wt*} : real expression level of gene a in wild type x_a^{wt*} (without noise)

we would like to know whether the deviation $x_a^b - x_a^{wt*}$ is merely due to noise.

→ Need to know the variance σ^2 of the Gaussian,
assuming the noise is non systematic so that the mean μ is zero.

Later, we will discuss the fact that x_a^{wt*} is also subject to noise so that we are only provided with the observed level x_a^{wt} .

Yip et al. PloS ONE 5:e8121 (2010)

Noise model

The probability for observing a deviation at least as large as $x_a^b - x_a^{wt*}$ due to random chance is

$$2[1 - \Phi(\frac{|x_a^b - x_a^{wt*}|}{\sigma})]$$

where Φ is the cumulative distribution function of the standard Gaussian distribution.

The deviation is taken relative to the width (standard dev.) of the Gaussian which describes the magnitude of the „normal“ spread in the data.

1 - CDF measures the area in the tail of the distribution.

The factor 2 accounts for the fact that we have two tails left and right.

The complement of this

$$p_{b \rightarrow a} = 1 - 2[1 - \Phi(\frac{|x_a^b - x_a^{wt*}|}{\sigma})] = 2\Phi(\frac{|x_a^b - x_a^{wt*}|}{\sigma}) - 1$$

is the probability that the deviation is due to a real (i.e. non-random) regulation event.

Yip et al. PloS ONE 5:e8121 (2010)

Noise model

One can then rank all the gene pairs (b,a) in descending order of $p_{b \rightarrow a}$.

For this we first need to estimate σ^2 from the data.

Two difficulties.

(1) the set of genes a not affected by the deleted gene b is unknown. This is exactly what we are trying to learn from the data.

(2) the observed expression value of a gene in the wild-type strain, x_a^{wt} , is also subjected to random noise, and thus cannot be used as the gold-standard reference point x_a^{wt*} in the calculations

Use iterative procedure to progressively refine estimation of $p_{b \rightarrow a}$.

Yip et al. PloS ONE 5:e8121 (2010)

Noise model

We start by assuming that the observed wild-type expression levels x_a^{wt} are reasonable rough estimates of the real wild type expression levels x_a^{wt*} .

For each gene a , our initial estimate for the variance of the Gaussian noise is set as the sample variance of all the expression values of a in the different deletion strains $b_1 - b_n$.

Repeat the following 3 steps for a number of iterations:

(1). Calculate the probability of regulation $p_{b \rightarrow a}$ for each pair of genes (b,a) based on the current reference points x_a^{wt} .

Then use a p-value of 0.05 to define the set of potential regulation:

if the probability for the observed deviation from wild type of a gene a in a deletion strain b to be due to random chance only is less than 0.05, we treat $b \rightarrow a$ as a potential regulation.

Otherwise, we add (b,a) to the set P of gene pairs for refining the error model.

Yip et al. PloS ONE 5:e8121 (2010)

Noise model

(2) Use the expression values of the genes in set P to re-estimate the variance of the Gaussian noise.

$$\sigma^2 = \frac{\sum_{(b,a):P} (x_a^b - x_a^{wt})^2}{|P| - 1}$$

(3) For each gene a , we re-estimate its wild-type expression level by the mean of its observed expression levels in strains in which the expression level of a is unaffected by the deletion

$$\hat{x}_a^{wt} = \frac{x_a^{wt} + \sum_{b:(b,a) \in P} x_a^b}{1 + |b : (b,a) \in P|}$$

After the iterations, the probability of regulation $p_{b \rightarrow a}$ is computed using the final estimate of the reference points \hat{x}_a^{wt} and the variance of the Gaussian noise σ^2 .

Yip et al. PloS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

For time series data after an initial perturbation, we use differential equations to model the gene expression rates. The general form is as follows:

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_n)$$

with x_i : expression level of gene i ,

$f_i(\dots)$: function that explains how the expression rate of gene i is affected by the expression level of all the genes in the network, including the level of gene i itself.

Yip et al. PloS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

Various types of function f_i have been proposed.

We consider two of them. The first one is a linear model

$$\frac{dx_i}{dt} = a_{i0} - a_{ii}x_i + \sum_{j \in S} a_{ij}x_j$$

a_{i0} : basal expression rate of gene i in the absence of regulators,

a_{ij} : decay rate of mRNA transcripts of i ,

S : set of potential regulators of i (we assume no self regulation, so i not element of S).

For each potential regulator j in S , a_{ij} explains how the expression of i is affected by the abundance of j .

A positive a_{ij} indicates that j is an activator of i , and a negative a_{ij} indicates that j is a suppressor of i .

The linear model contains $|S| + 2$ parameters a_{ij} .

Yip et al. PloS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

The linear model assumes a linear relationship between the expression level of the regulators and the resulting expression rate of the target.

But real biological regulatory systems seem to exhibit nonlinear characteristics. The second model assumes a sigmoidal relationship between the regulators and the target

$$\frac{dx_i}{dt} = \frac{b_{i1}}{1 + \exp(-a_{i0} - \sum_{j \in S} a_{ij}x_j)} - b_{i2}x_i$$

b_{i1} : maximum expression rate of i , b_{i2} : its decay rate

The sigmoidal model contains $|S| + 3$ parameters.

Try 100 random initial values and refine parameters by Newton minimizer so that the predicted expression time series give the least squared distance from the real time series.

Score: negative squared distance

Yip et al. PloS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

- Batch 1 contains the most confident predictions: all predictions with probability of regulation ($p_{b \rightarrow a} > 0.99$ according to the noise model learned from homozygous deletion data)
- Batch 2: all predictions with a score two standard deviations below the average according to all types (linear AND sigmoidal) of differential equation models learned from perturbation data
- Batch 3: all predictions with a score two standard deviations below the average according to all types of guided differential equation models learned from perturbation data, where the regulator sets contain regulators predicted in the previous batches, plus one extra potential regulator
- Batch 4: as in batch 2, but requiring the predictions to be made by only one type (linear OR sigmoidal) of the differential equation models as opposed to all of them.
- Batch 5: as in batch 3, but requiring the predictions to be made by only one type of the differential equation models as opposed to all of them
- Batch 6: all predictions with $p_{b \rightarrow a} > 0.95$ according to both the noise models learned from homozygous and heterozygous deletion data, and have the same edge sign predicted by both models
- Batch 7: all remaining gene pairs, with their ranks within the batch determined by their probability of regulation according to the noise model learned from homozygous deletion data

Yip et al. PLoS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

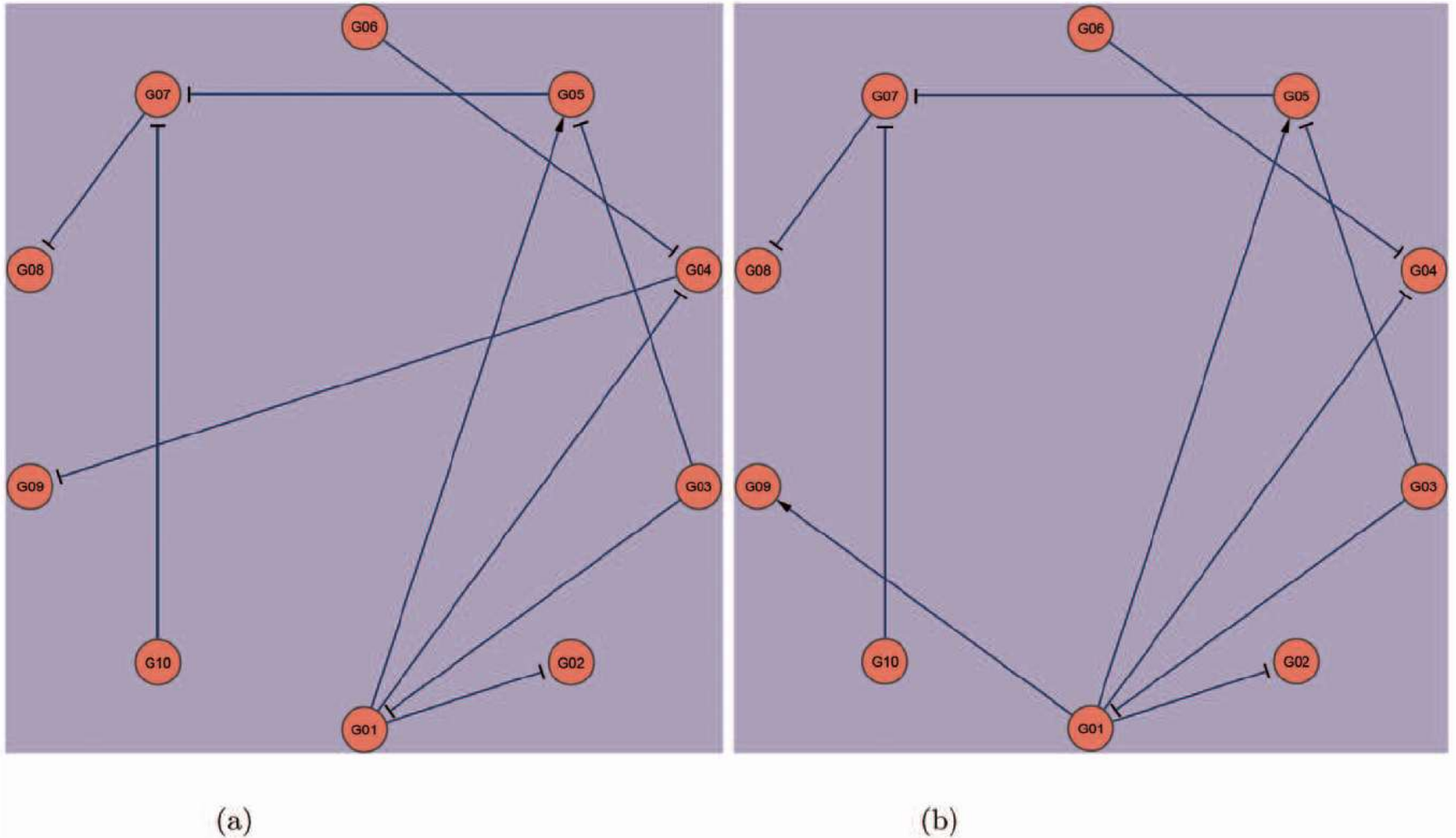


Figure 1. The Yeast1-size10 network. (a) The actual network. (b) Our top-10 predictions.

Yip et al. PLoS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

Table 3. Prediction accuracy per batch on the size 10 networks.

Batch	Ecoli1		Ecoli2		Yeast1		Yeast2		Yeast3	
	Predicted	Correct	Predicted	Correct	Predicted	Correct	Predicted	Correct	Predicted	Correct
1	11	7	16	12	11	9	13	9	12	8
2	6	1	4	0	5	0	5	1	5	4
3	0	0	1	1	3	0	1	0	1	0
4	5	1	8	0	7	0	4	2	4	0
5	4	0	8	1	6	0	10	3	5	1
6	1	1	0	0	0	0	0	0	0	0
7	63	1	53	1	58	1	57	10	63	9
Total	90	11	90	15	90	10	90	25	90	22

Interpretation:

A network with 10 nodes has 10×9 possible edges

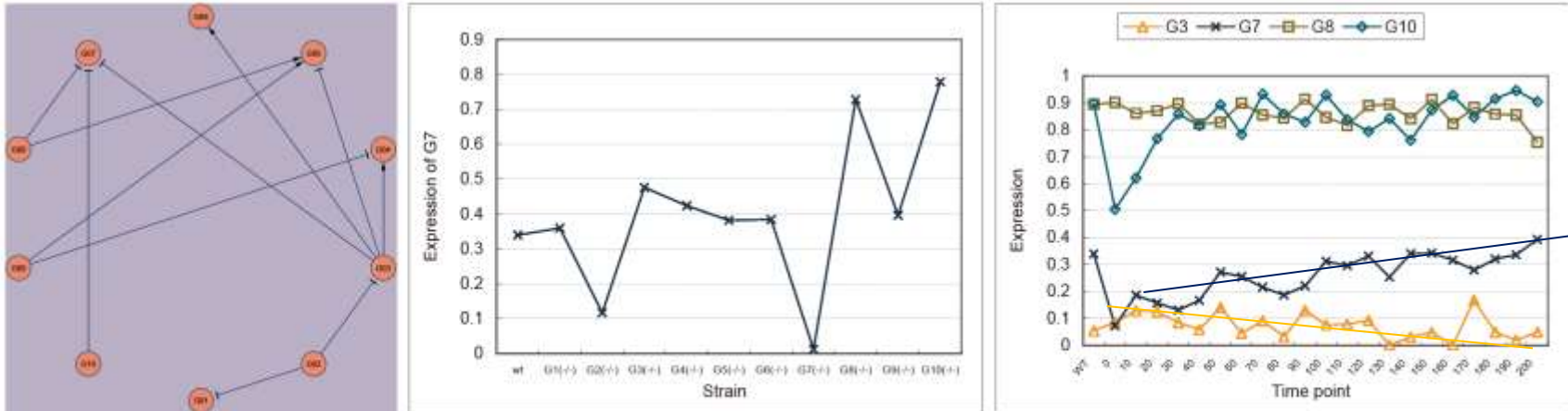
Batch 1 already contains many of the correct edges (7/11 – 8/22).

The majority of the high-confidence predictions are correct (7/11 – 8/12).

Batch 7 contains only 1 correct edge for the E.coli-like network, but 9 or 10 correct edges for the Yeast-like network.

Yip et al. PloS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data



Not all regulation arcs can be detected from deletion data (middle):

Left: G7 is suppressed by G3, G8 and G10

Right: G8 and G10 have high expression levels in wt.

Middle: removing the inhibition by G3 therefore only leads to small increase of G7 which is difficult to detect.

However the right panel suggests that the increased expression of G7 over time is anti-correlated with the decreased level of G3

→ This link was detected by the ODE-models in batch 2

Yip et al. PloS ONE 5:e8121 (2010)

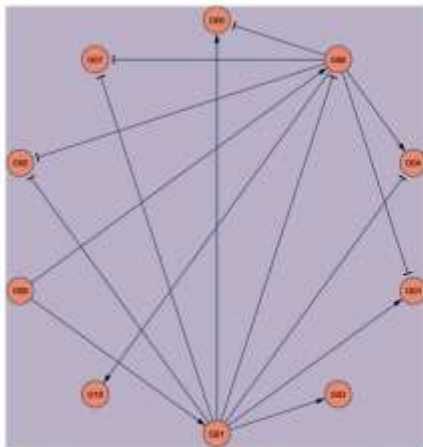
Learning ODE models from perturbation time series data

Another case:

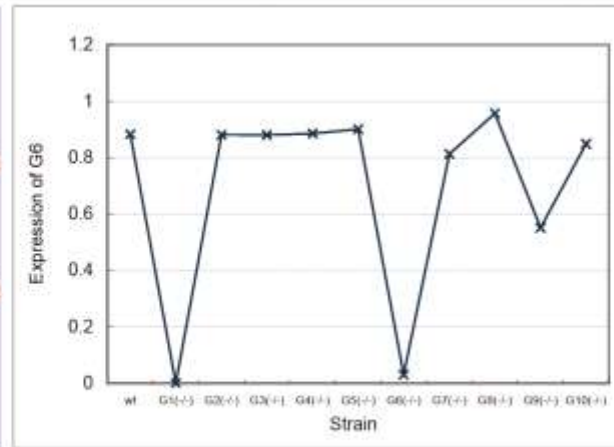
Left: G6 is activated by G1 and suppressed by G5. G1 also suppresses G5. G1 therefore has 2 functions on G6. When G1 is expressed, deleting G5 (middle) has no effect.

Right: G6 appears anti-correlated to G1. Does not fit with activating role of G1.

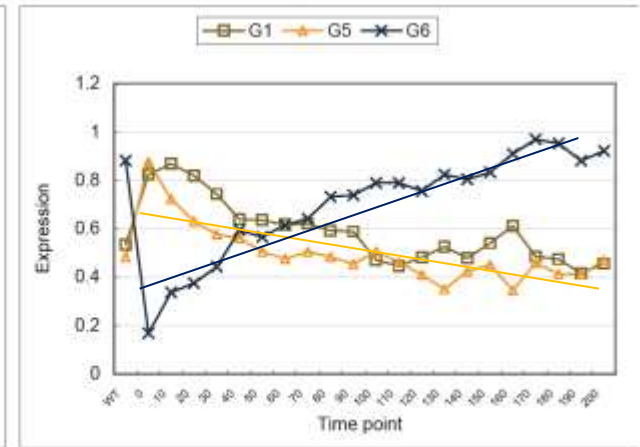
But G5 is also anti-correlated with G6 → evidence for inhibitory role of G5.



(d)



(e)



(f)

Yip et al. PLoS ONE 5:e8121 (2010)

Summary : deciphering GRN topologies is hard

GRN networks are hot topic.

They give detailed insight into the circuitry of cells.

This is important for understanding the molecular causes e.g. of diseases.

New data are constantly appearing.

The computational algorithms need to be adapted.

Perturbation data (knockouts and time series following perturbations) are most useful for mathematic reconstruction of GRN topologies.

Yip et al. PloS ONE 5:e8121 (2010)