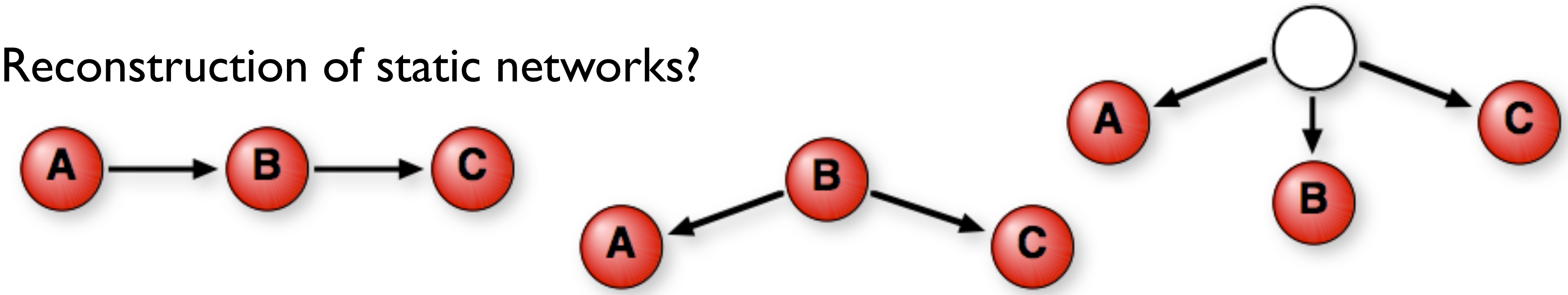
V13 –
Reconstruction of
Gene Regulatory Networks
- Benchmarking

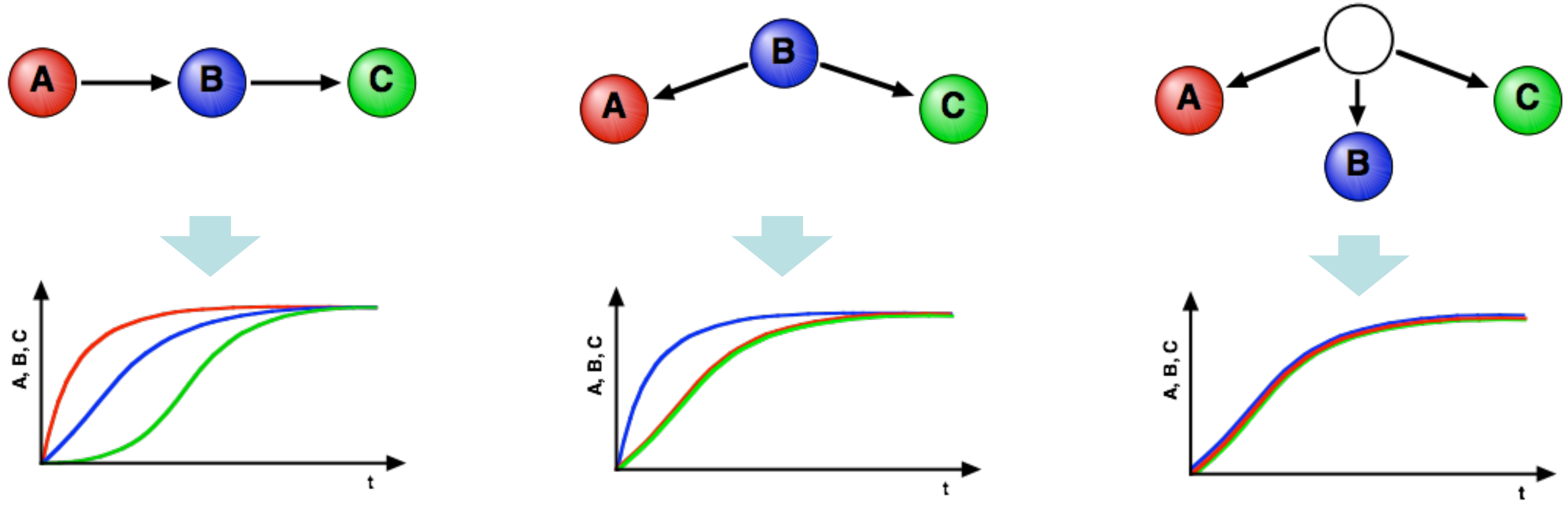
Tue, Dec 3, 2019

Static vs. Dynamic Reconstruction

Reconstruction of static networks?

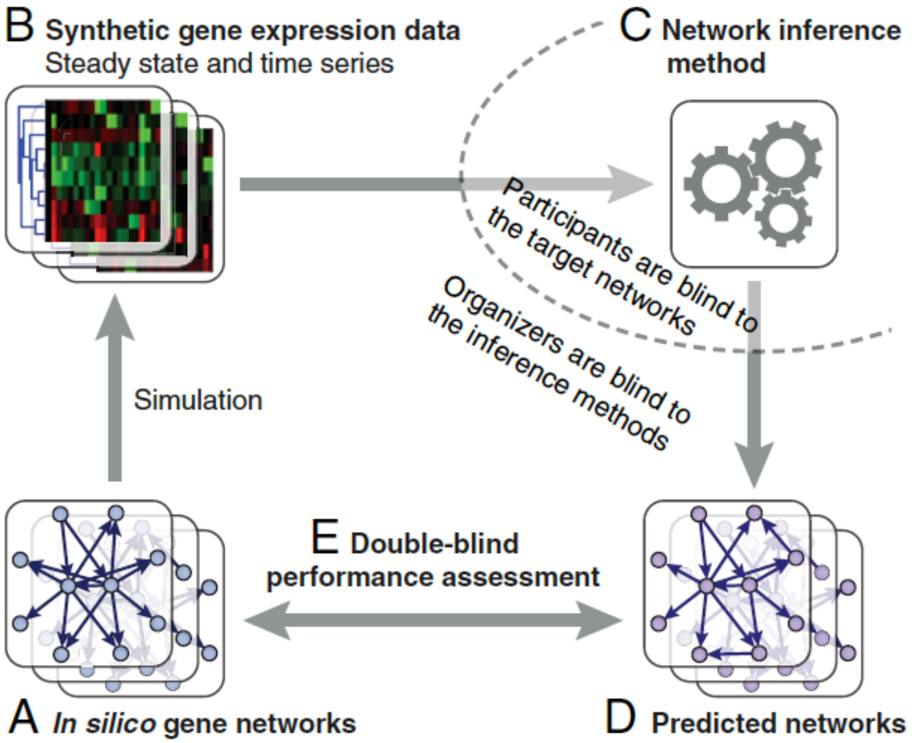


Different network topologies → different time series



Mathematical reconstruction of Gene Regulatory Networks

DREAM: Dialogue on Reverse Engineering Assessment and Methods


B Synthetic gene expression data
Steady state and time series
C Network inference method
Participants are blind to the target networks
Organizers are blind to the inference methods
Simulation
E Double-blind performance assessment
A *In silico* gene networks
D Predicted networks

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

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

Approach:
generate **synthetic data**

Gustavo Stolovitzky/IBM

Marbach et al. PNAS 107, 6286 (2010)

Generation of Synthetic Data

Model transcriptional regulatory networks consisting of mRNA and proteins.

Current **state** of network :

vector of **mRNA concentrations x** and **protein concentrations y**.

Considered is only transcriptional regulation, where regulatory proteins (TFs) control the activation of genes; no epigenetics, microRNAs etc.

The gene network is modeled by a **system of differential equations** (equivalent to V11, slide 24).

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

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

m_i : maximum **transcription rate**,

r_i : **translation rate**,

$f_i(.)$: so-called **input function** of gene *i*.

λ_i^{RNA} , λ_i^{Prot} : mRNA and protein **degradation rates**

Marbach et al. PNAS 107, 6286 (2010)

The input function $f_i()$

The input function describes the relative activation of a gene given the transcription-factor (TF) concentrations **y**.
Its value is between 0 (gene shut off) and 1 (gene maximally activated).

We assume that **binding of TFs** to cis-regulatory sites on the DNA is in **quasi-equilibrium**, since TF binding is orders of magnitudes faster than transcription and translation (which take minutes).

In the **simplest 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 *S1*) or not bound (state *S0*).

The probability $P(S_1)$ that the gene *i* is in state *S1* at a particular moment is given by the **fractional saturation**, which depends on the TF concentration y_j

Marbach et al. PNAS 107, 6286 (2010)

Excursion: the Hill equation (see V9, slide 33)

Let us consider the binding reaction of two molecules *L* and *M:*

$$L + M \leftrightarrows LM$$

The **dissociation equilibrium constant** K_D is defined as:

$$K_{\mathrm{D}} = \frac{[L][M]}{[LM]}$$

where [*L*], [*M*], and [*LM*] are the molecular concentrations of *L* and *M* and of the complex *LM*.

In equilibrium, we may take *T* as the total concentration of molecule *L*

$$T = [L] + [LM]$$

y is the **fraction** of molecules *L* **that have reacted (bound)**

$$y = \frac{[LM]}{[LM] + [L]}$$

Goutelle et al. Fundamental & Clinical Pharmacology 22 (2008) 633–648

Excursion: the Hill equation (see V9, slide 34)

$$y = \frac{[LM]}{[LM] + [L]}$$

Substituting [*LM*] by [*L*] [*M*] / K_D gives (rearranged from $K_{\mathrm{D}} = \frac{[L][M]}{[LM]}$)

$$y = \frac{([L]\,[M]\,/K_D)}{([L]\,[M]\,/K_D) + [L]} = \frac{([M]\,/K_D)}{([M]\,/K_D) + 1}$$

Back to our case about TF binding to DNA. **(slightly different from V9)**
TF *j* then takes the role of *M*. Its concentration is y_j.

The probability $P(S_1)$ that the gene *i* is in state *S1* at a particular moment is given by the *fractional saturation*, which depends on the TF concentration y_j

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

k_{ij} : dissociation constant for TF *j* at the promoter of gene *i*
n_{ij} : Hill coefficient (describing cooperativity) for this binding equilibrium.

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 y_j of TF j is large
and if the dissociation constant k_{ij} is small (strong binding).

The bound TF either activates or represses the expression of the gene.

In state S_0 the **relative activation** is α_0. In state S_1 it is α_1.

The **input function** $f_i(y_j)$ is obtained from $P(S_1)$ and its complement $P(S_0)$.

$$P\left(S_0\right) = 1 - \frac{\chi_j}{1+\chi_j} = \frac{1+\chi_j-\chi_j}{1+\chi_j} = \frac{1}{1+\chi_j}$$

The input function describes 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 generalized
to an **arbitrary number** of regulatory inputs.

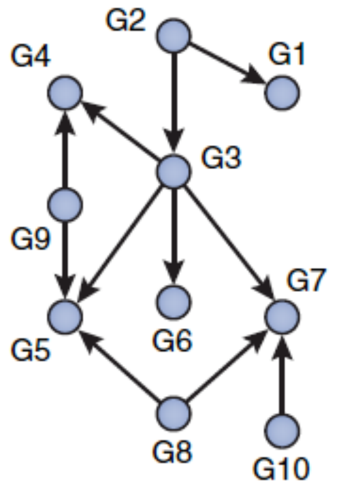
A gene that is controlled by *N* TFs has 2^N states:
each of the TFs can be bound or not bound.

Thus, the input function for *N* regulators is

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

Marbach et al. PNAS 107, 6286 (2010)

Synthetic gene expression data



Gene knockouts were simulated for the DREAM competition by setting the maximum transcription rate of the deleted gene to 0.

Gene knockdowns were simulated by dividing it by 2.

Time-series experiments were simulated by integrating the dynamic evolution of the network ODEs.

For networks of **size** 10, 50, and 100,
4, 23, and 46 different **time series** of 21 time points were provided.

Marbach et al. PNAS 107, 6286 (2010)

Synthetic gene expression data

For each time series, a different **random initial condition** was used for the mRNA and protein concentrations.

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

White noise (with zero auto-correlation) with a standard deviation of 0.05 was added after the simulation to the generated gene expression data.

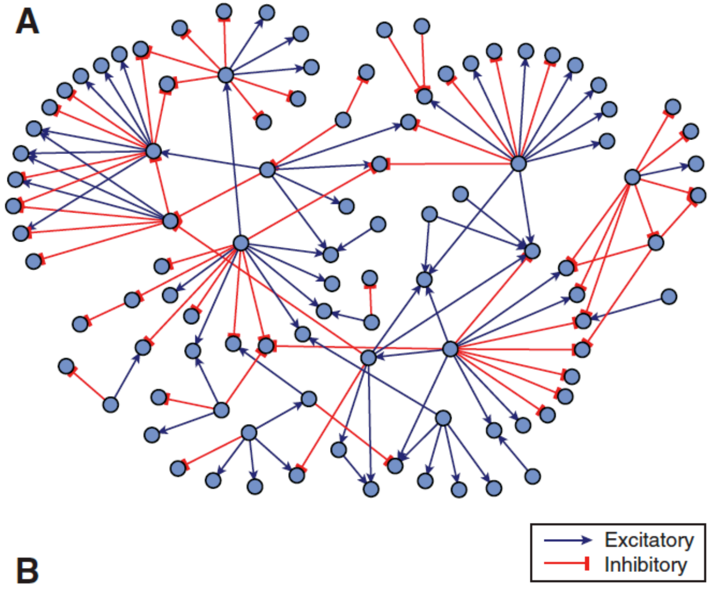
Marbach et al. PNAS 107, 6286 (2010)

Synthetic networks

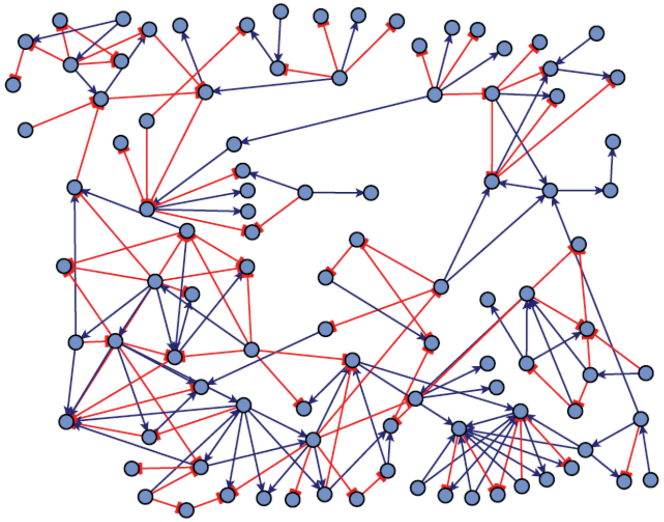
The challenge was structured as 3 separate subchallenges with networks of 10, 50, and 100 genes, respectively.

For each size, 5 in silico networks were generated that resembled realistic network structures by extracting modules from the known transcriptional regulatory network for *E. coli* (2x) and for yeast (3x).

Example network *E.coli*



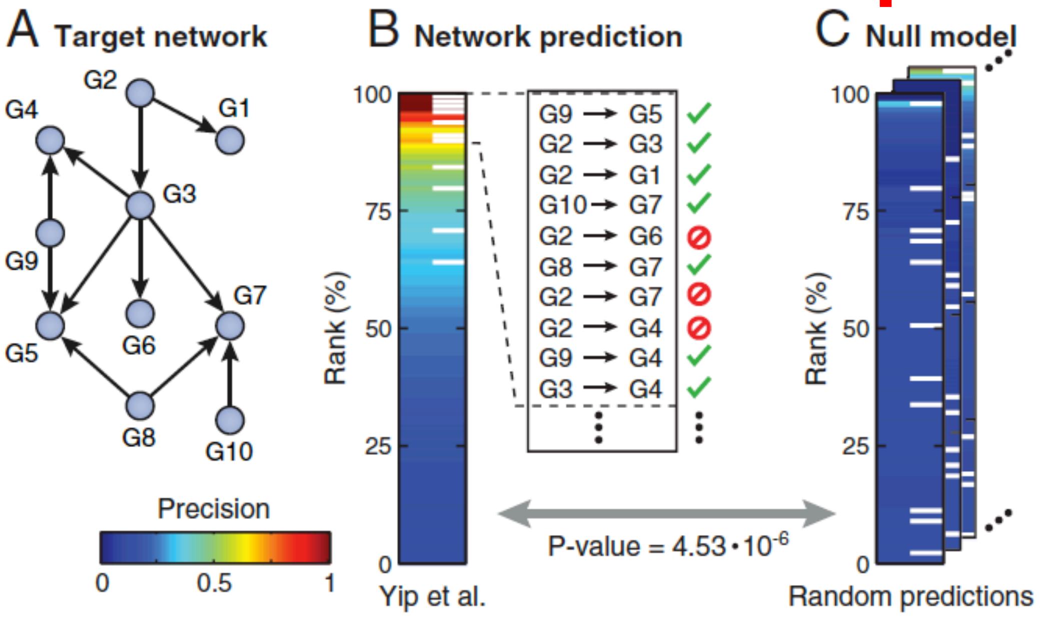
Example network yeast



Marbach et al. PNAS 107, 6286 (2010)

Evaluation of network predictions

(A) 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.

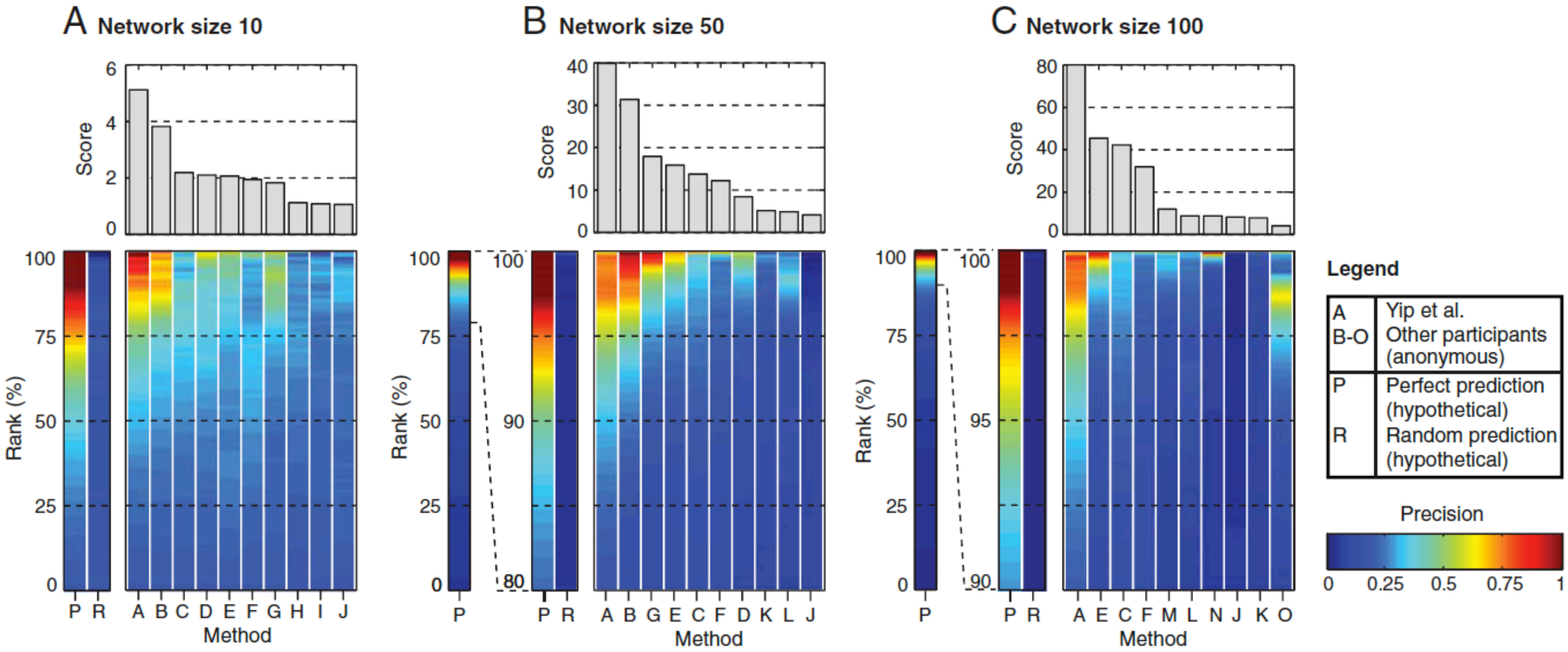
The format is a ranked list of predicted edges, represented here by the vertical colored bar.

White stripes : true edges of the target network. A perfect prediction would have all white stripes at the top of the list.

Inset shows 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)

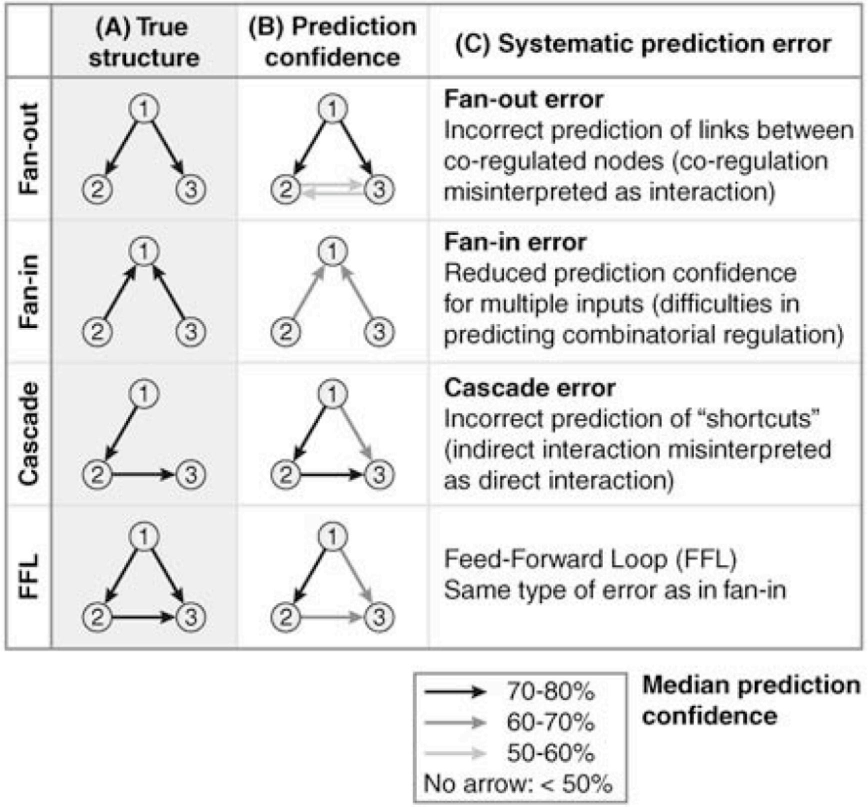
Similar performance on different network sizes



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

Marbach et al. PNAS 107, 6286 (2010)

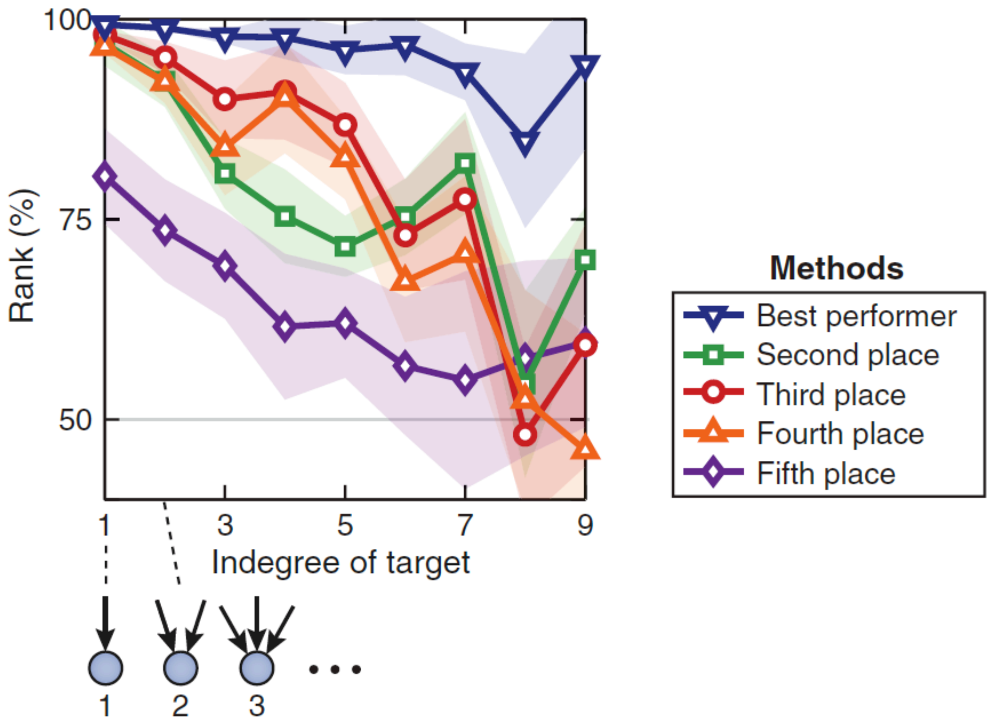
Error analysis



	(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

Median prediction confidence

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

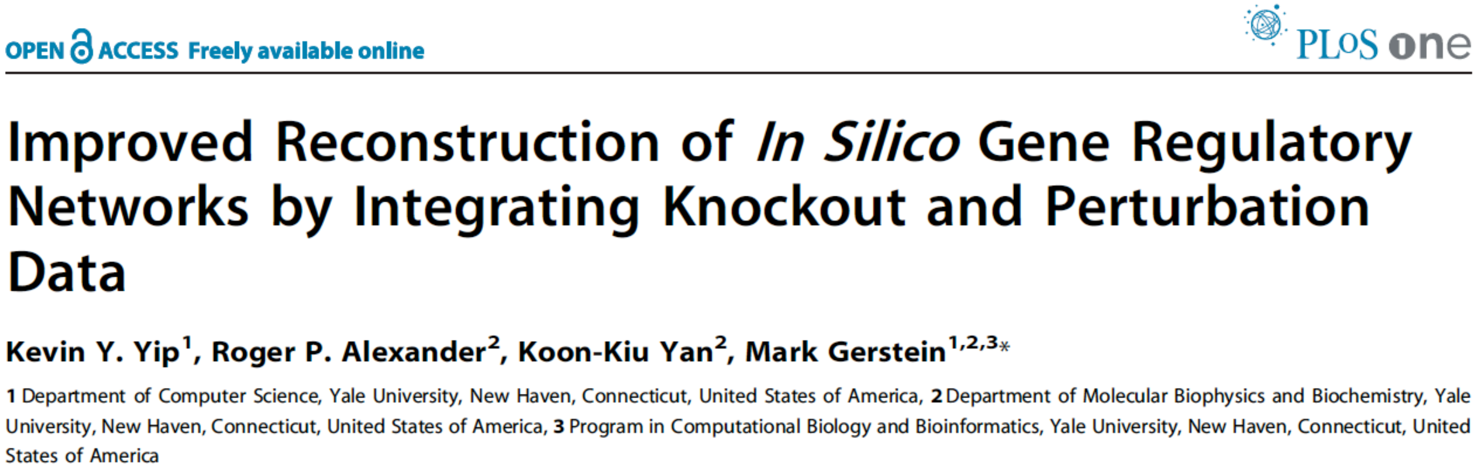


Left: 3 typical errors made in predicted networks.

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)

Synthetic networks



OPEN ACCESS Freely available online

PLoS one

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

Kevin Y. Yip[1], Roger P. Alexander[2], Koon-Kiu Yan[2], 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 3 models were combined.

Mark Gerstein/Yale

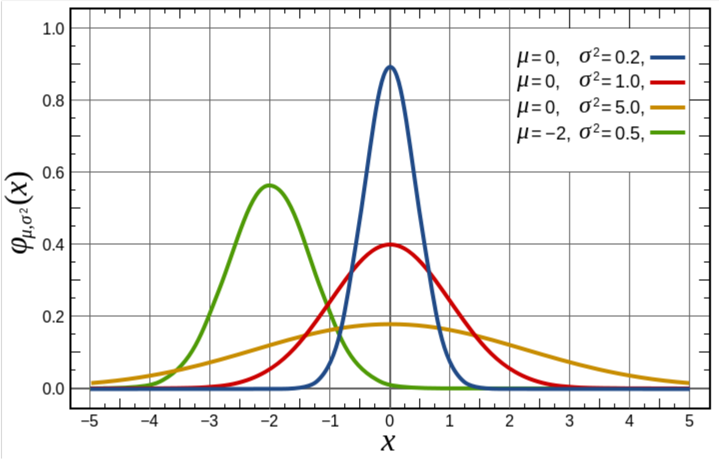
Yip et al. PloS ONE 5:e8121 (2010)

Cumulative distribution function

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

$$F_X(x) = \mathrm{P}(X \leq x),$$

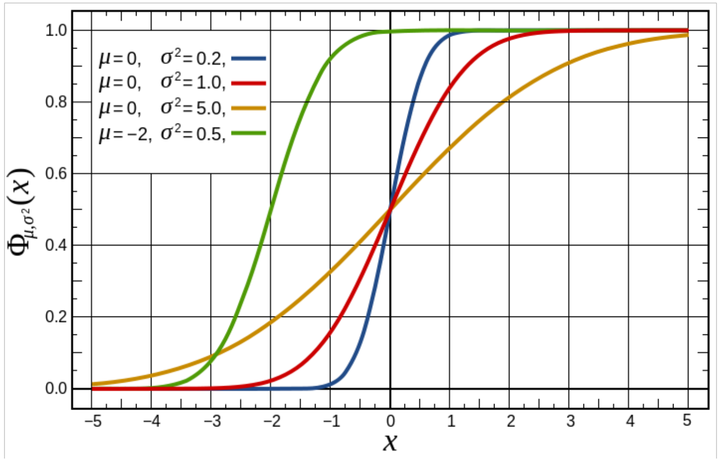
$$F(x) = \int_{-\infty}^{x} f(t)\,dt.$$



Different normal distributions

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

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



CDF of the normal distribution

www.wikipedia.org

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) expression levels, 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 (one on the left and right each).

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

Noise model

The complement of the above equation

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

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

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

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

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

Noise model

Two **difficulties** exist:

(1) the **set of genes** *a* that are **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 subject to **random noise**.

Thus, it cannot be used as the gold-standard reference point x_a^{wt*} in the calculations

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

Noise model

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

First, assume 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*, the 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.

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

Noise model

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 regulations:
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

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

After the iterations, the **probability of regulation** $p_{b \to a}$ is computed using the final estimate of the reference points 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, ODEs are used to model the gene expression rates.

The general form is:

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

with x_i : expression level of gene *i* ,

f_i *(…):* 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 (slide omitted)

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_{ii} : 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 I S I + 2 parameters a_{ij}.

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

Learning ODE models from perturbation time series data (slide omitted)

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 often seem to exhibit nonlinear characteristics. The second model assumes a **sigmoidal relationship** between the regulators and the target

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

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

The sigmoidal model contains I S I + 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)

Group predicted interactions into classes

- Batch 1 contains the **most confident predictions** ($p_{b \to a} > 0.99$) 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 $p_{b \to a} > 0.95$ 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

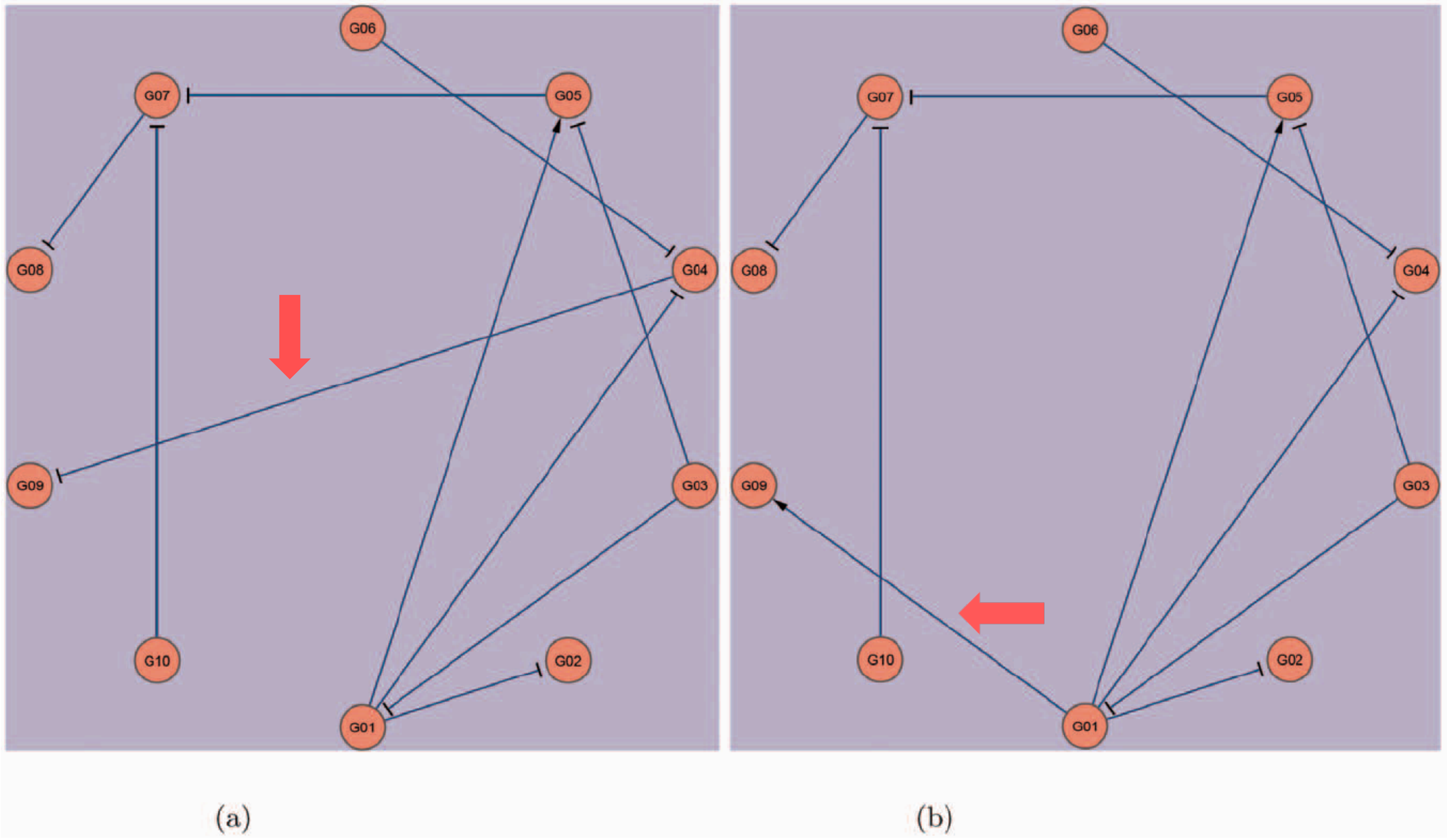


Figure 1. The Yeast1-size10 network. (a) The actual network. (b) Our top-10 predictions.

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

Prediction accuracy

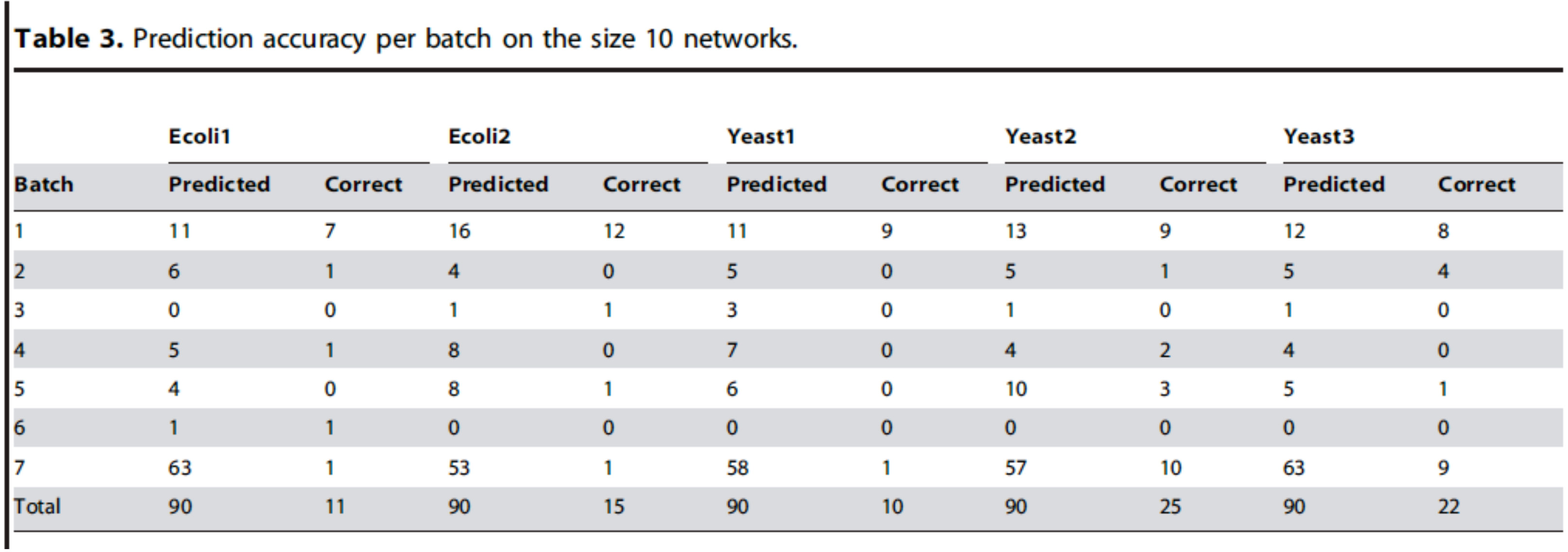


Table 3. Prediction accuracy per batch on the size 10 networks.

	Ecoli1		Ecoli2		Yeast1		Yeast2		Yeast3	
Batch	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 x 9 possible edges

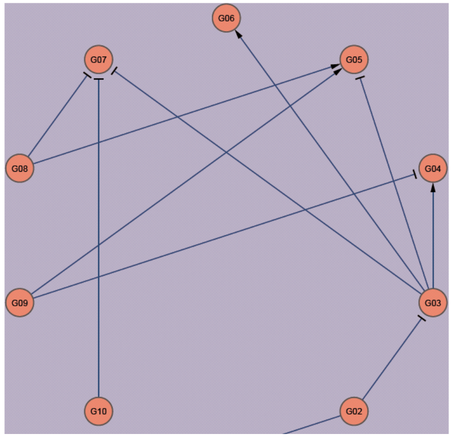
Batch 1 already contains many of the correct edges (7/11 – 8/22).

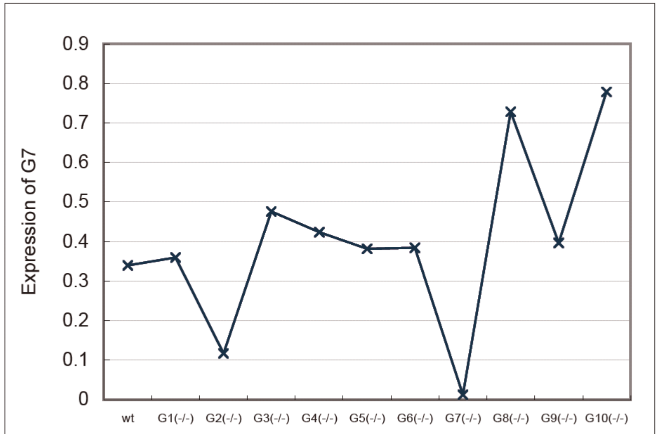
The majority of the high-confidence predictions are correct (7/11 – 8/12).

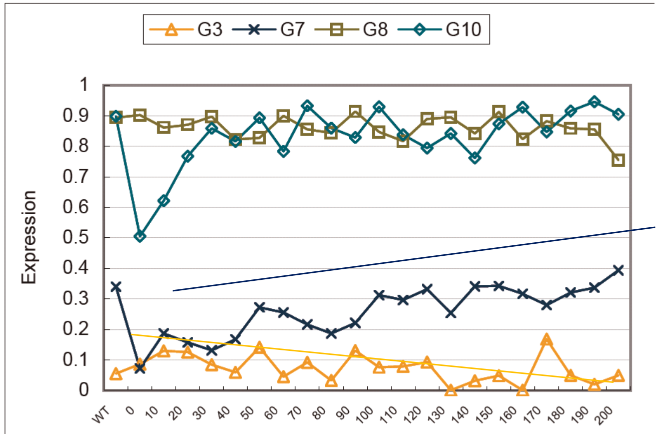
Batch 7 contains only 1 correct edge for the *E.coli*-like network, but 9 or 10 correct edges for the Yeast-like network.

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

Can all regulations be predicted equally well?







Not all regulatory 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)

Problematic dependencies (II)

Another case:

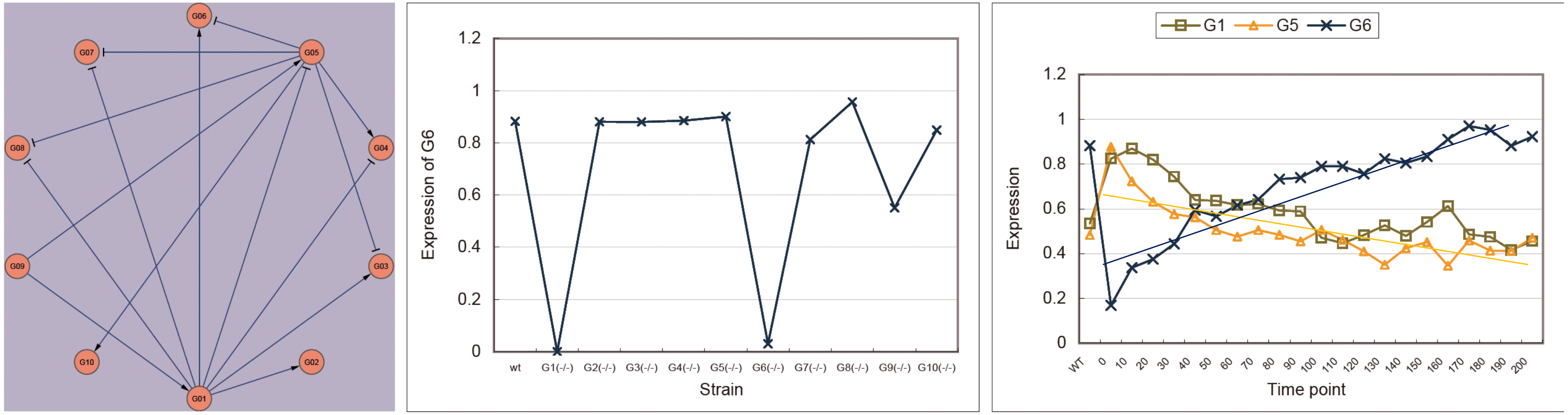
Left: G6 is activated by G1 and suppressed by G5. G1 also suppresses G5.

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.



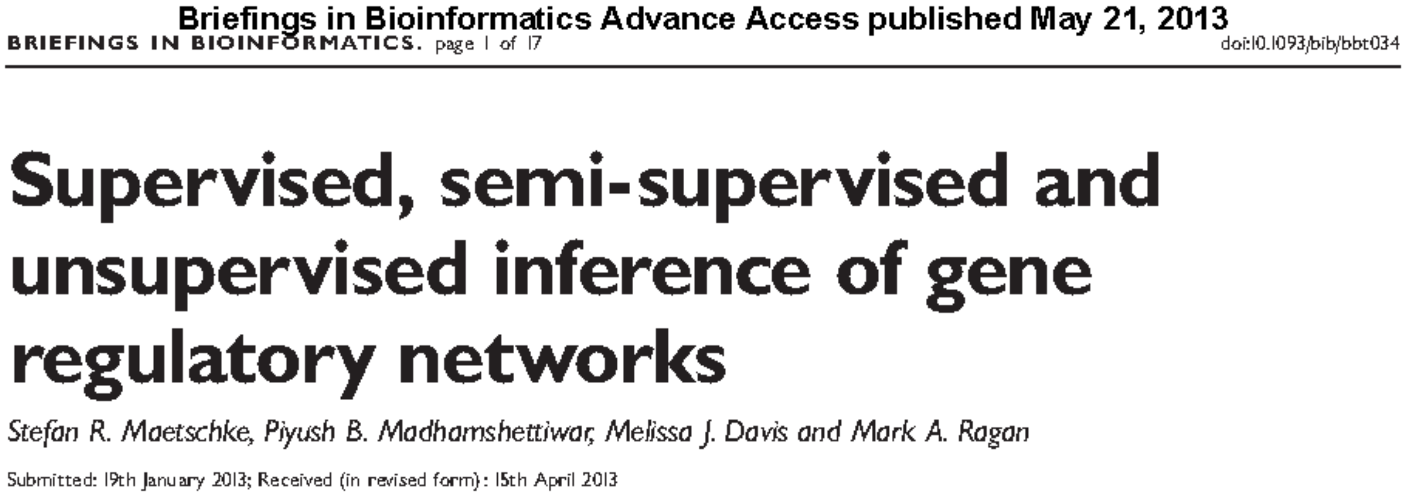
(d) (e) (f)

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

How does one generate GRNs?

(1.) „by hand“ based on individual experimental observations

(2) Infer GRNs by computational methods from gene expression data (see reference below)
Unsupervised methods are either based on **correlation** or on **mutual information**. (We will not cover supervised methods here).



Briefings in Bioinformatics Advance Access published May 21, 2013

BRIEFINGS IN BIOINFORMATICS. page 1 of 17 doi:10.1093/bib/bbt034

Supervised, semi-supervised and unsupervised inference of gene regulatory networks

Stefan R. Maetschke, Piyush B. Madhamshettiwar, Melissa J. Davis and Mark A. Ragan

Submitted: 19th January 2013; Received (in revised form): 15th April 2013

Correlation-based unsupervised methods

Correlation-based network inference methods assume that correlated expression levels between two genes are indicative of a regulatory interaction (note however slide 42 in lecture V9).

Correlation coefficients range from -1 to 1.
A **positive** correlation coefficient indicates an **activating interaction**, whereas a **negative** coefficient indicates an **inhibitory interaction**.

The common correlation measure by **Pearson** is defined as

$$corr(X_i, X_j) = \frac{cov(X_i, X_j)}{\sigma(X_i) \cdot \sigma(X_j)}$$

where X_i and X_j are the expression levels of genes *i* and *j*,
cov(.,.) denotes the **covariance**, and σ is the **standard deviation**.

Rank-based unsupervised methods

Pearson's correlation measure assumes normally distributed values.
This assumption does not necessarily hold for gene expression data.

Therefore rank-based measures are frequently used.
The measures by Spearman and Kendall are the most common.

Spearman's method is simply Pearson's correlation coefficient for the ranked expression values

Kendall's τ coefficient :

$$\tau(X_i, X_j) = \frac{con(X_i^r, X_j^r) - dis(X_i^r, X_j^r)}{\frac{1}{2}n(n-1)}$$

where X^r_i and X^r_j are the ranked expression profiles of genes *i* and *j*.

Con(.) denotes the number of concordant value pairs (i.e. where the ranks for both elements agree). *dis(.)* is the number of disconcordant value pairs in X^r_i and X^r_j . Both profiles are of length *n*.

WGCNA

WGCNA is a modification of correlation-based inference methods that **amplifies high correlation coefficients** by raising the absolute value to the power of β (‘softpower’).

$$w_{ij} = |corr(X_i, X_j)|^{\beta}$$

with $\beta \geq 1$.

Because softpower is a nonlinear but monotonic transformation of the correlation coefficient, the prediction accuracy measured by AUC will be no different from that of the underlying correlation method itself.

Z-score

Z-SCORE is a network inference strategy by Prill *et al.* that assumes the availability of **knockout experiments** that lead to a change in other genes.

The assumption is that the knocked-out gene *i* in experiment *k* affects more strongly the genes that it regulates than the others.

The effect of gene *i* on gene *j* is captured with the Z-score z_{ij}:

$$z_{ij} = \left| \frac{x_{jk} - \mu_{X_j}}{\sigma_{X_j}} \right|$$

assuming that the *k*-th experiment is a knockout of gene *i*, μ_{Xj} and σ_{Xj} are respectively the mean and standard deviation of the empirical distribution of the expression values x_{jk} of gene *j*.

Unsupervised methods based on mutual information

Relevance networks (RN) introduced by Butte and Kohane measure the **mutual information (MI)** between gene expression profiles to infer interactions.

The MI between discrete variables (here: expression levels of genes) X_i and X_j is defined as

$$M_{ij} = \sum_{X_i} \sum_{X_j} p(X_i, X_j) \log_2 \frac{p(X_i, X_j)}{p(X_i)p(X_j)}$$

where $p(X_i, X_j)$ is the **joint probability distribution** of X_i and X_j
(both variables fall into given ranges) and
$p(X_i)$ and $p(X_j)$ are the **marginal probabilities** of the two variables
(ignoring the value of the other one).

RELNET

The RELNET is the simplest method based on **mutual information**.

For each pair of genes, the mutual information M_{ij} is estimated and
the edge between genes i and j is created
if the mutual information is above a threshold.

Although mutual information is more general than Pearson correlation,
in practice both give similar results.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

CLR

The Context Likelihood or Relatedness network (CLR) method is an extension of RELNET.

CLR derives a score that is associated to the empirical distribution of the mutual information values.

The score between gene *i* and gene *j* is:

$$c_{ij} = \sqrt{c_i^2 + c_j^2}, \text{ with } c_i = \max\left(0, \frac{M_{ij} - \mu_{M_i}}{\sigma_{M_i}}\right) \text{ and}$$

$$c_j = \max\left(0, \frac{M_{ji} - \mu_{M_j}}{\sigma_{M_j}}\right).$$

with the mean μ_{Mi} and standard deviation σ_{Mi} of the empirical distribution of the mutual information between these genes and other genes,

$$\mu_{M_i} = \frac{1}{G}\sum_{l=1}^{G} M_{il}, \quad \sigma_{M_i} = \sqrt{\frac{1}{G-1}\sum_{l=1}^{G}(M_{il} - \mu_{M_i})^2}$$

Bellot *et al. BMC Bioinformatics* (2015) 16:312

ARACNE

Motivation behind the
"Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE)":
imany similar measures between variables may be due to indirect effects.

In order to avoid such indirect effects, the algorithm relies on the
"Data Processing Inequality" (DPI).

In every triplet of genes,
DPI removes the weakest edge having the lowest mutual information

Bellot *et al. BMC Bioinformatics* (2015) 16:312

PCIT

The "Partial Correlation coefficient with Information Theory (PCIT)" algorithm combines the concept of **partial correlation coefficients** with information theory to identify significant gene-to-gene associations.

Similarly to ARACNE, PCIT extracts **all possible interaction triangles** and applies DPI to filter indirect connections, but instead of mutual information it uses first-order partial correlation as interaction weights.

The partial correlation tries to eliminate the effect of a third gene *l* on the correlation of genes *i* and *j*.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

C3NET

The Conservative Causal Core NETwork (C3NET) consists of 2 main steps.

(1) Pairwise mutual information is computed.
Then, non-significant connections are eliminated, according to a chosen significance level α, between gene pairs.

(2) One selects the **most significant edge** for each gene: it has the highest mutual information value among the neighboring connections for each gene.

→ the highest possible number of connections that can be reconstructed by C3NET is equal to the number of genes under consideration.

C3NET does not aim at reconstructing the entire network underlying gene regulation but mainly tries to recover the core structure.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

Feature selection approaches

A GRN reconstruction problem can also be seen as a feature selection problem.
For every gene, the goal is to discover its true regulators among all other genes or candidate regulators. This approach can integrate knowledge about genes that are not TFs and therefore reduce the search space.

Typically, this approach only focuses on designing a significance score *s(i, j)* that leads to a good ranking of the candidate regulations, such that true regulations tend to be at the top of the list since an edge is assigned between *i* and *j* if the evidence *s(i, j)* is larger than a threshold.

With the feature selection approach, the scores *s(i, j)* for all the genes are jointly estimated with a method that is able to capture the fact that a large score for a link *(i, j)* is not needed if the apparent relationship between *i* and *j*
is already explained by another and more likely regulation.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

MRNET

The Minimum Redundancy NETworks (MRNET) method reconstructs a network using the feature selection technique known as Minimum Redundancy Maximum Relevance (MRMR), which is based on a mutual information measure.

In order to generate a network, the algorithm performs a feature selection for each gene ($i \in [1, G]$) on the set of remaining genes ($j \in [1, G] \setminus i$).

The MRMR procedure returns a ranked list of features that maximize the mutual information with the target gene (maximum relevance) and, at the same time, such that the selected genes are mutually dissimilar (minimum redundancy).

Bellot *et al. BMC Bioinformatics* (2015) 16:312

MRNET

For every gene, the MRMR feature selection provides a score of potential connections where the higher scores should correspond to direct interactions.

The indirect interactions should have lower scores because they are redundant with the direct ones.

Then, a threshold is computed as in the RELNET method.

The MRNET reconstructs a network using a forward selection strategy, which leads to subset selection that is strongly conditioned by the first selected variables.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

Genie3

The GEne Network Inference with Ensemble of trees (Genie3) algorithm uses the **random forests** feature selection technique to solve a regression problem for each of the genes in the network.

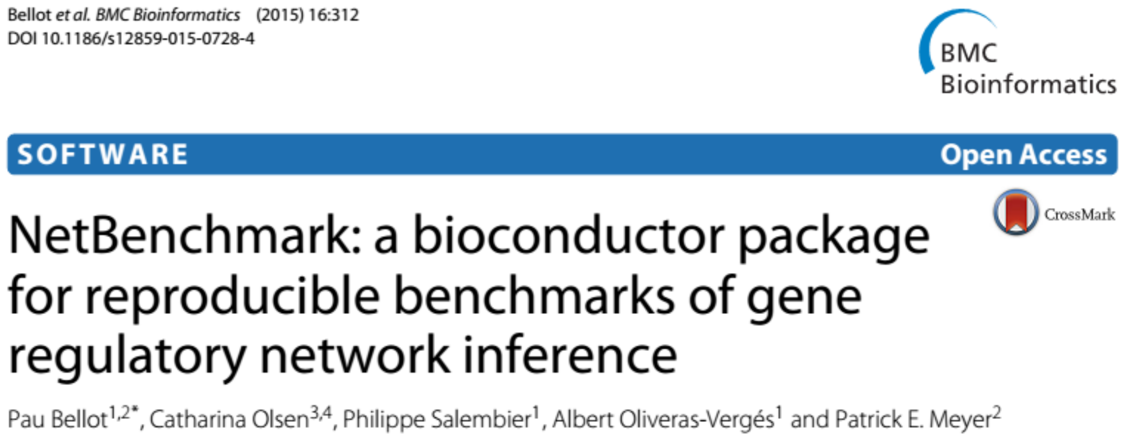
In each of the regression problems, the expression pattern of the target gene should be predicted from the expression patterns of all TFs.

The importance of each TF in the prediction of the target gene is taken as an indication of an apparent regulatory edge.

Then these candidate regulatory connections are aggregated over all genes to generate a ranking for the whole network.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

GRN benchmark



Bellot *et al. BMC Bioinformatics* (2015) 16:312
DOI 10.1186/s12859-015-0728-4

BMC Bioinformatics

SOFTWARE Open Access

CrossMark

NetBenchmark: a bioconductor package for reproducible benchmarks of gene regulatory network inference

Pau Bellot[1,2*], Catharina Olsen[3,4], Philippe Salembier[1], Albert Oliveras-Vergés[1] and Patrick E. Meyer[2]

Real data suffers from drawbacks.

(1) the different algorithms are tested based on only partial knowledge of the underlying network, where a false positive could be a still undiscovered true positive.

(2) the intensity of noise is uncontrollable → assessing a method's robustness to varying intensities of noise cannot be done easily with real data.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

Workflow

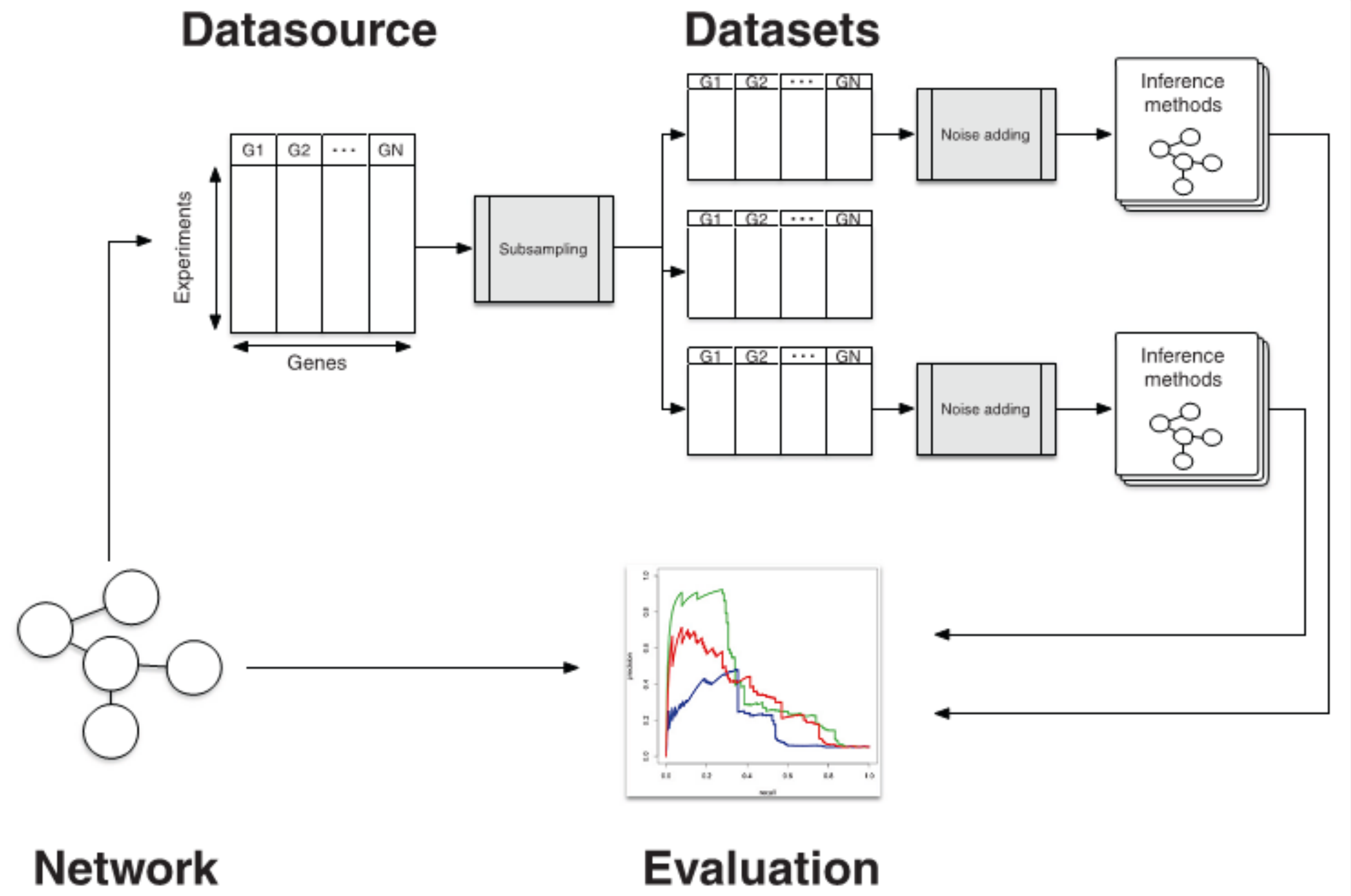


Fig. 1 Workflow of the network evaluation process

Bellot *et al. BMC Bioinformatics* (2015) 16:312

Generation of synthetic data

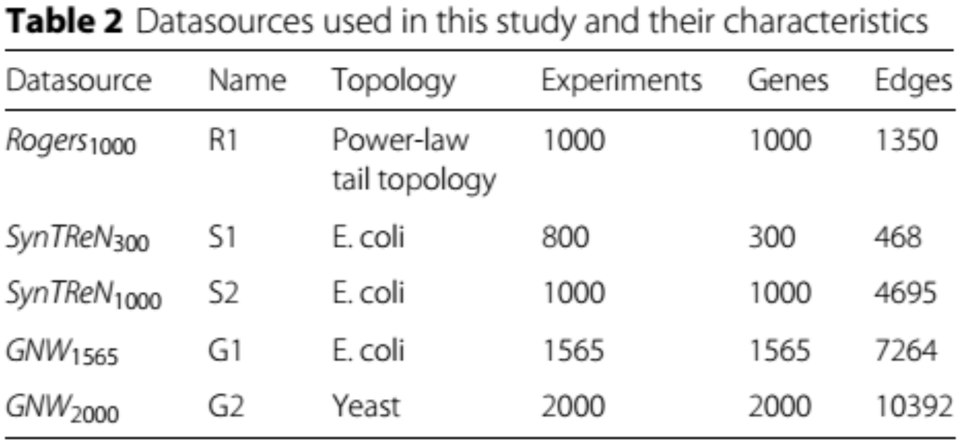


Table 2 Datasources used in this study and their characteristics

Datasource	Name	Topology	Experiments	Genes	Edges
Rogers$_{1000}$	R1	Power-law tail topology	1000	1000	1350
SynTReN$_{300}$	S1	E. coli	800	300	468
SynTReN$_{1000}$	S2	E. coli	1000	1000	4695
GNW$_{1565}$	G1	E. coli	1565	1565	7264
GNW$_{2000}$	G2	Yeast	2000	2000	10392

GNW The GNW simulator generates network structures by extracting parts of known real GRN structures capturing several of their important structural properties. To produce gene expression data, the simulator relies on a system of non-linear ODEs.

SynTReN The SynTReN simulator generates the underlying networks by selecting sub-networks from *E. coli* and *Yeast* organisms. Then the experiments are obtained by simulating equations based on Michaelis-Menten and Hill kinetics under different conditions.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

Computational runtimes

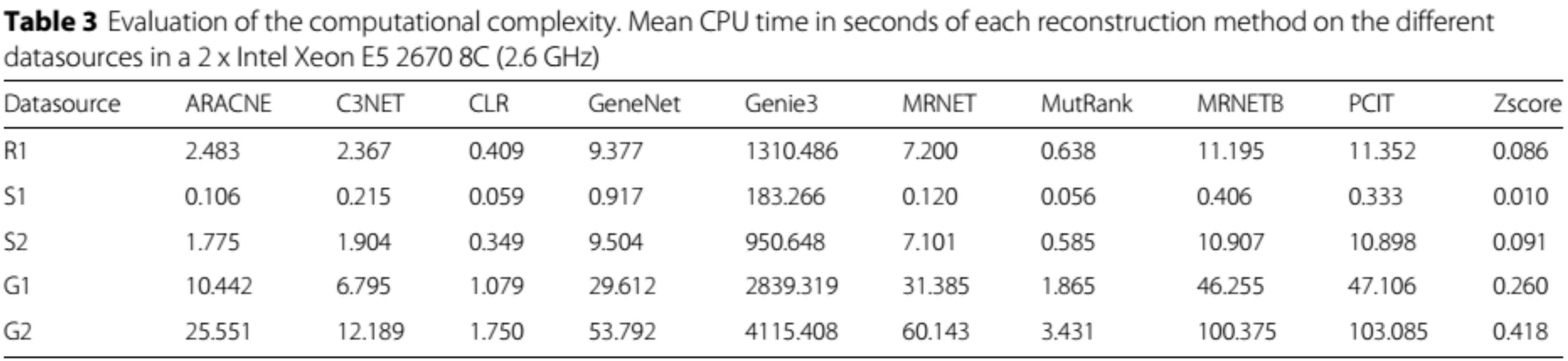


Table 3 Evaluation of the computational complexity. Mean CPU time in seconds of each reconstruction method on the different datasources in a 2 x Intel Xeon E5 2670 8C (2.6 GHz)

Datasource	ARACNE	C3NET	CLR	GeneNet	Genie3	MRNET	MutRank	MRNETB	PCIT	Zscore
R1	2.483	2.367	0.409	9.377	1310.486	7.200	0.638	11.195	11.352	0.086
S1	0.106	0.215	0.059	0.917	183.266	0.120	0.056	0.406	0.333	0.010
S2	1.775	1.904	0.349	9.504	950.648	7.101	0.585	10.907	10.898	0.091
G1	10.442	6.795	1.079	29.612	2839.319	31.385	1.865	46.255	47.106	0.260
G2	25.551	12.189	1.750	53.792	4115.408	60.143	3.431	100.375	103.085	0.418

Different methods have very different runtimes.

Genie3 is the slowest method.

Z-score is the fastest method, followed by CLR.

Bellot *et al. BMC Bioinformatics* (2015) 16:312

Methods generate at most 18% correct links

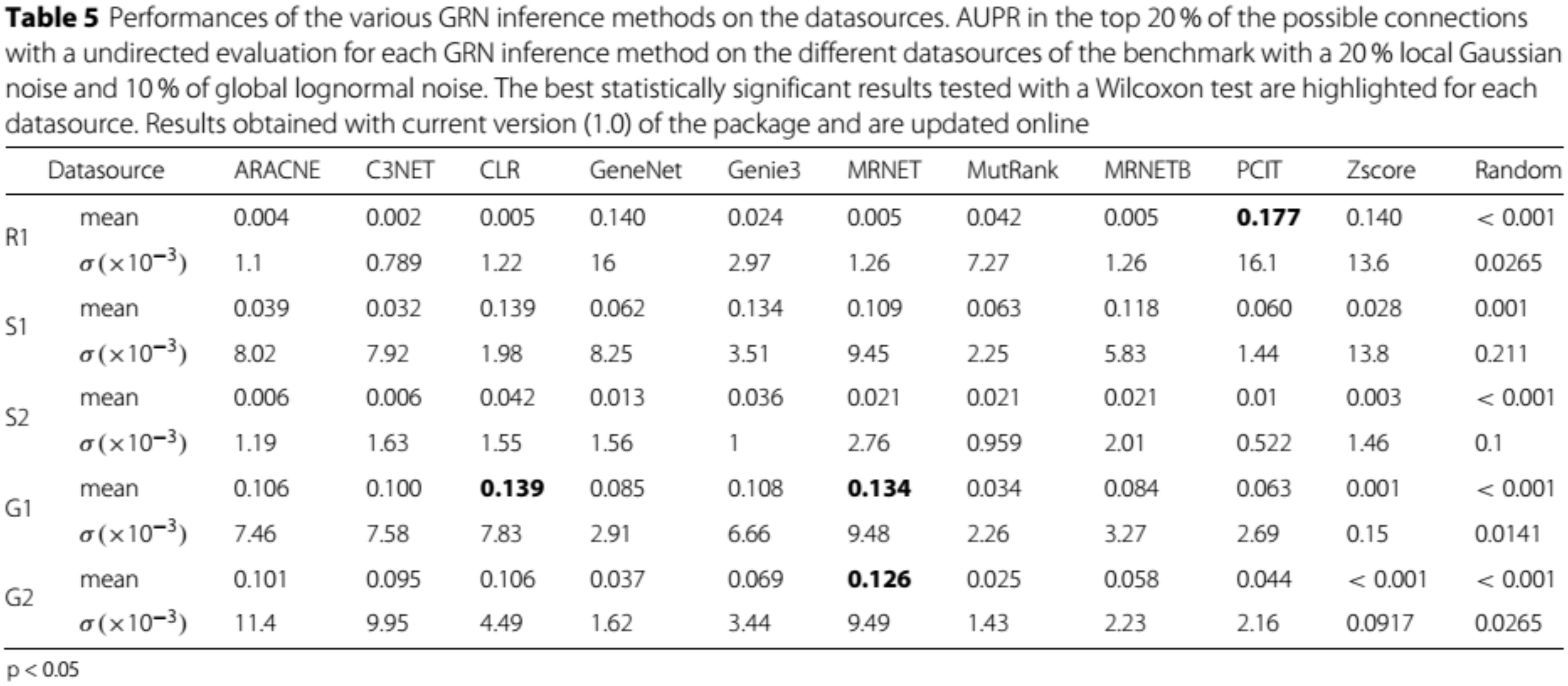


Table 5 Performances of the various GRN inference methods on the datasources. AUPR in the top 20 % of the possible connections with a undirected evaluation for each GRN inference method on the different datasources of the benchmark with a 20 % local Gaussian noise and 10 % of global lognormal noise. The best statistically significant results tested with a Wilcoxon test are highlighted for each datasource. Results obtained with current version (1.0) of the package and are updated online

Datasource		ARACNE	C3NET	CLR	GeneNet	Genie3	MRNET	MutRank	MRNETB	PCIT	Zscore	Random
R1	mean	0.004	0.002	0.005	0.140	0.024	0.005	0.042	0.005	**0.177**	0.140	< 0.001
	$\sigma(\times 10^{-3})$	1.1	0.789	1.22	16	2.97	1.26	7.27	1.26	16.1	13.6	0.0265
S1	mean	0.039	0.032	0.139	0.062	0.134	0.109	0.063	0.118	0.060	0.028	0.001
	$\sigma(\times 10^{-3})$	8.02	7.92	1.98	8.25	3.51	9.45	2.25	5.83	1.44	13.8	0.211
S2	mean	0.006	0.006	0.042	0.013	0.036	0.021	0.021	0.021	0.01	0.003	< 0.001
	$\sigma(\times 10^{-3})$	1.19	1.63	1.55	1.56	1	2.76	0.959	2.01	0.522	1.46	0.1
G1	mean	0.106	0.100	**0.139**	0.085	0.108	**0.134**	0.034	0.084	0.063	0.001	< 0.001
	$\sigma(\times 10^{-3})$	7.46	7.58	7.83	2.91	6.66	9.48	2.26	3.27	2.69	0.15	0.0141
G2	mean	0.101	0.095	0.106	0.037	0.069	**0.126**	0.025	0.058	0.044	< 0.001	< 0.001
	$\sigma(\times 10^{-3})$	11.4	9.95	4.49	1.62	3.44	9.49	1.43	2.23	2.16	0.0917	0.0265

$p < 0.05$

Listed are „Area Under Precision Recall" values obtained in an undirected evaluation on the top 20 % (*AUPR*20 %) of the total possible connections for each data source The *AUPR*20 % values have different ranges for each data source.

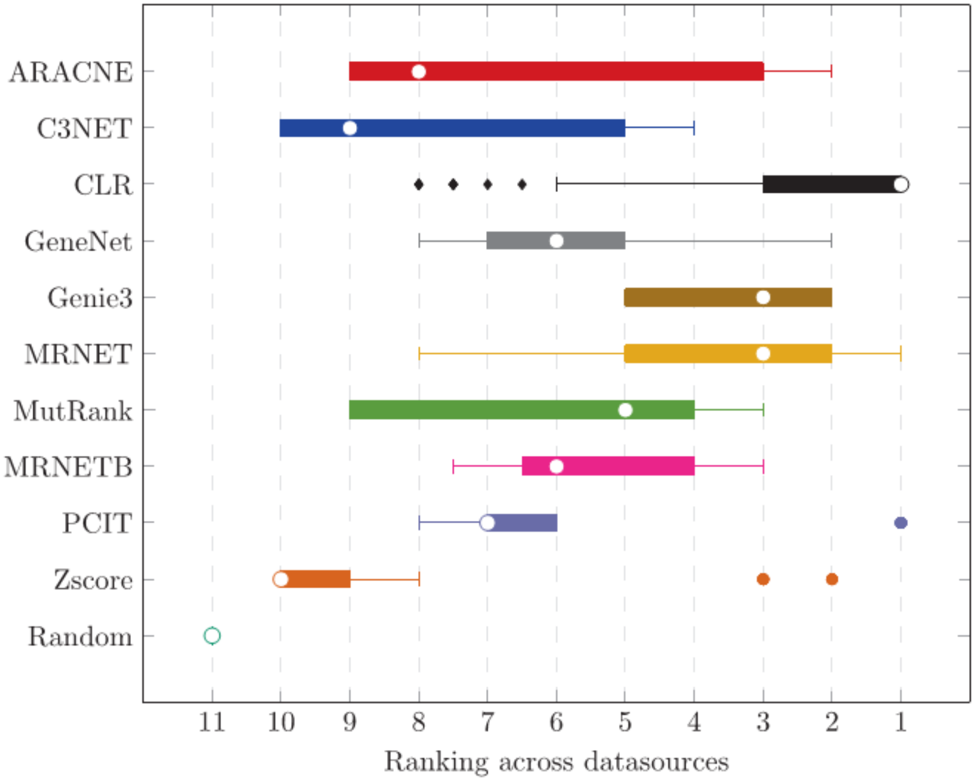
Bellot *et al. BMC Bioinformatics* (2015) 16:312

Aggregated ranking of methods

CLR is the best on the majority of the datasets, but it does not obtain the best results across all the different data sources and kinds of data.

In the case of complete knockout data, the best-performing methods are the Zscore followed by PCIT and GeneNet.

Genie3 and MRNET exhibit competitive performances. However, these methods are not as fast as CLR in terms of computation time.



Bellot *et al. BMC Bioinformatics* (2015) 16:312

Summary

Network inference is a very important active research field.

Inference methods allow to construct the topologies of gene-regulatory networks solely from expression data.
Also functional interpretation of exp. data, guiding inhibitor design etc.

Current GRN models are **limited** by
(1) incomplete knowledge about TF → target gene relations
(2) about the regulatory effects (activation vs. repression)

(3) Performance on real data is lower than on synthetic data
because regulation in cells is not only due to interaction
of TFs with genes,
but also depends on epigenetic effects (DNA methylation,
chromatin structure/histone modifications, and miRNAs).