Bioinformatics 3

V9 – Topologies and Dynamics of Gene Regulatory Networks

Dynamic Regulation: Boolean Networks

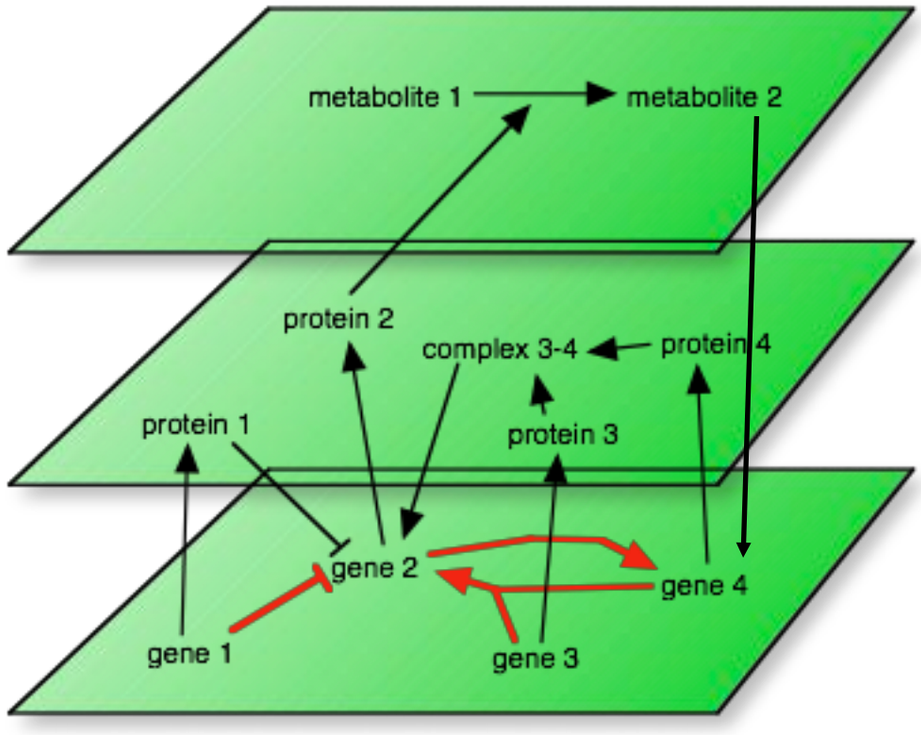
Fri, Nov 21, 2014

Gene Regulation Networks

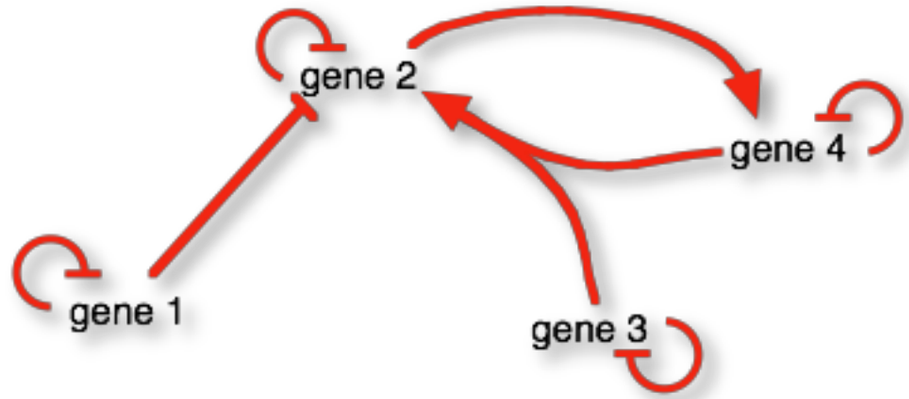
Biological regulation
via proteins and metabolites

<=>

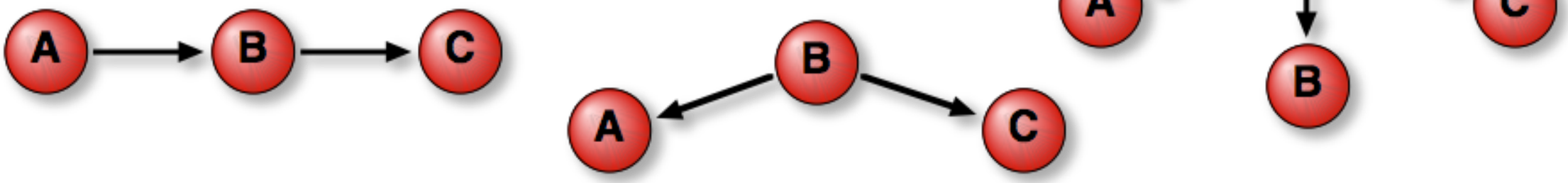
Projected regulatory network



<=>

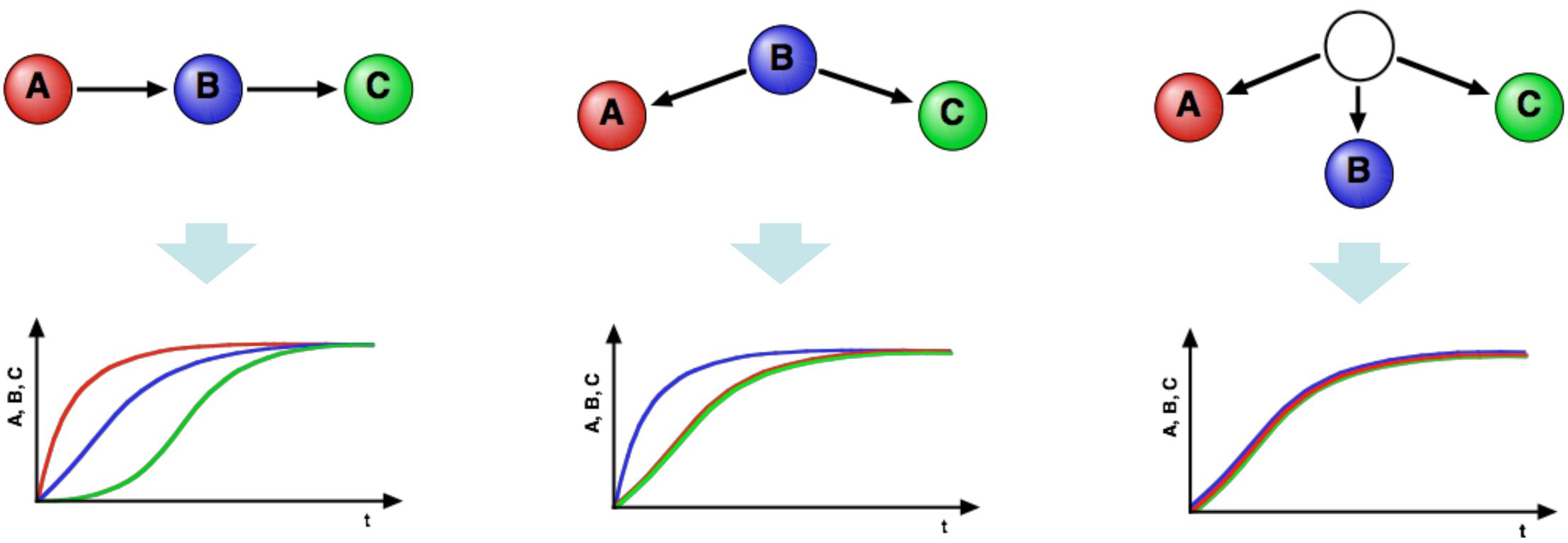


Reconstruction of static networks?



Dynamic Reconstruction

Different network topologies → different time series



Model large networks efficiently → simplified descriptions
(processes + numerics)

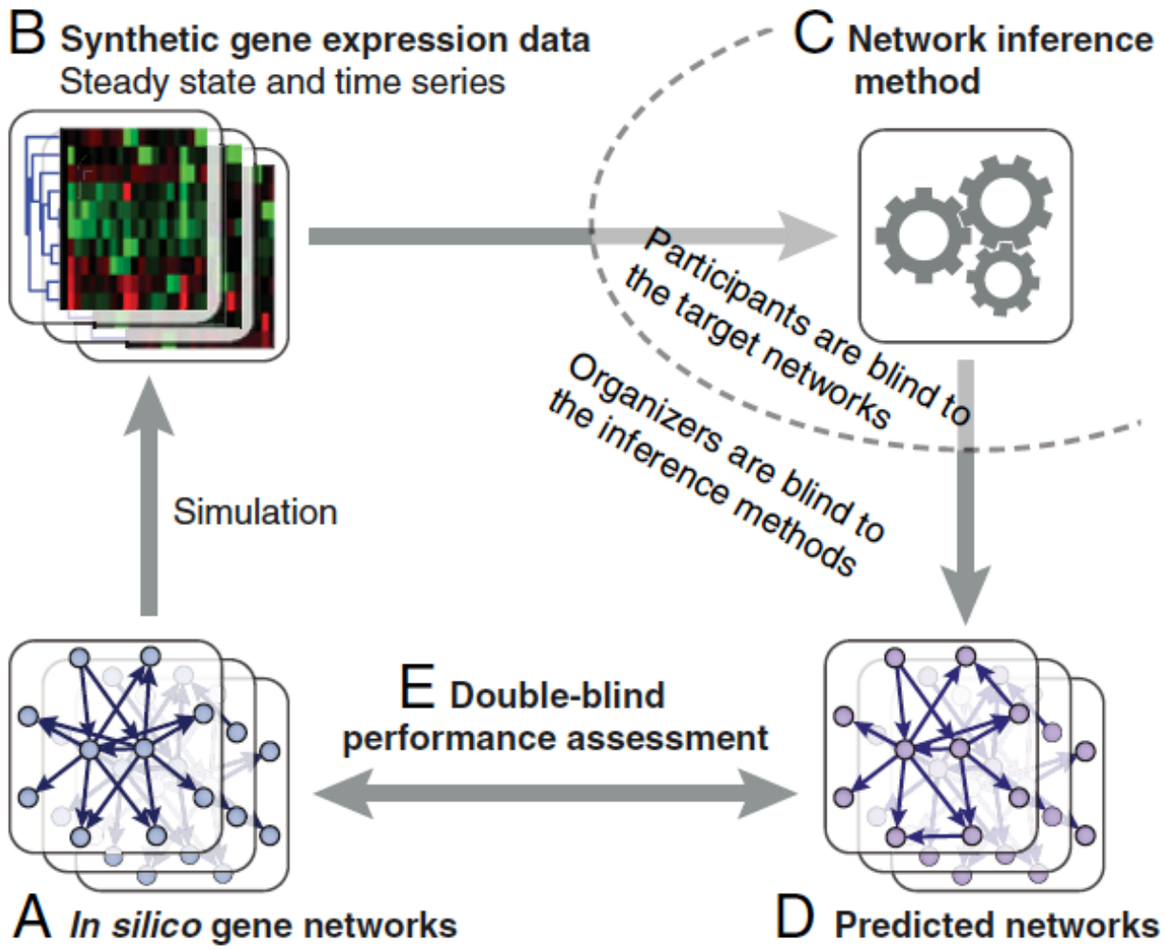
Regulatory Networks

DREAM: Dialogue on Reverse Engineerging Assessment and Methods

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

<u>Problem</u>:
correct answer is typically not known for real biological networks

<u>Approach</u>:
generate synthetic data



Gustavo Stolovitzky/IBM

Marbach et al. PNAS 107, 6286 (2010)

Generation of Synthetic Data

Transcriptional regulatory networks are modelled consisting of genes, 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 transcription rate (activation) of genes (no epigenetics, microRNAs etc.).

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

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

d by a **system of differential equations**

m_i : maximum transcription rate,

r_i the translation rate,

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

f_i(.) 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 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 *S1*) or it is 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

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:

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

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

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

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

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

Excursion: the Hill equation

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

Substituting [LM] by [L] [M] / K_D gives (rearranged fro $K_D = \frac{[L][M]}{[LM]}$)

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

Back to our case about TF binding to DNA. TF then takes the role of M. Divide eq by K_D.

The probability $P(S_1)$ that the gene *i* is in state *S1* at a particular moment is given by the *fractional satur*

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

where k_{ij} is the dissociation constant for TF *j* at the promoter of gene *i* and is n_{ij} the Hill coefficient (that describes 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 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.

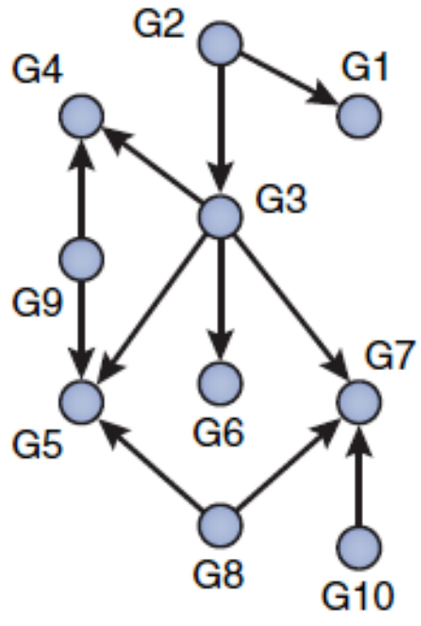
A gene that is controlled by N TFs has 2^N states: each of the TFs can be bound or not bound.

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 0, **gene knockdowns** by dividing it by 2.

Time-series experiments were simulated by integrating the dynamic evolution of the network ODEs using different initial conditions.

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

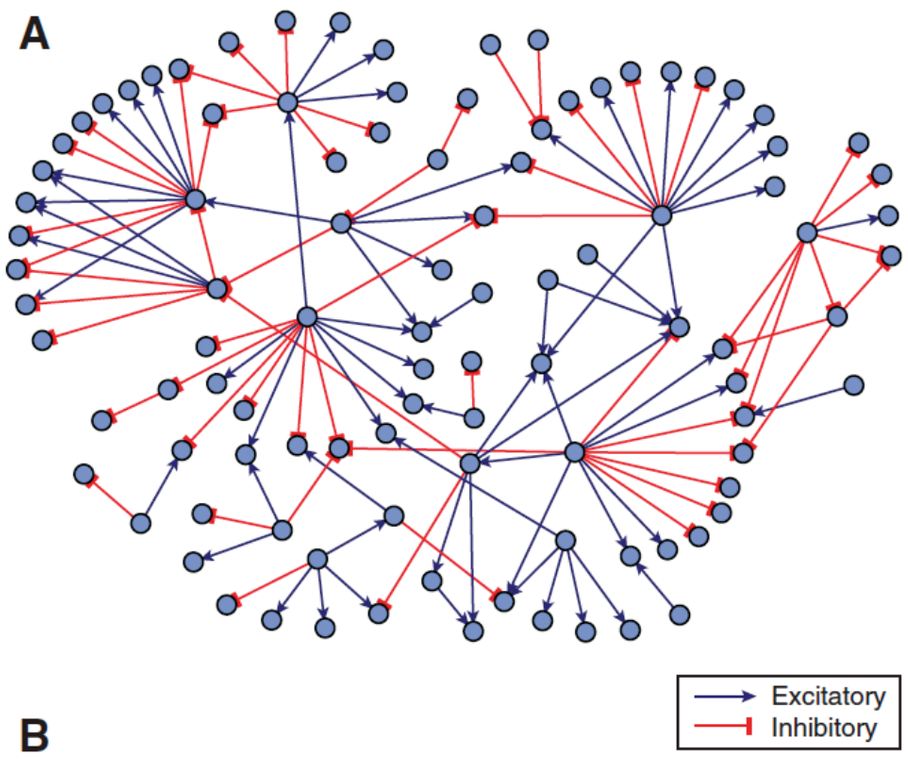
Marbach et al. PNAS 107, 6286 (2010)

Synthetic networks

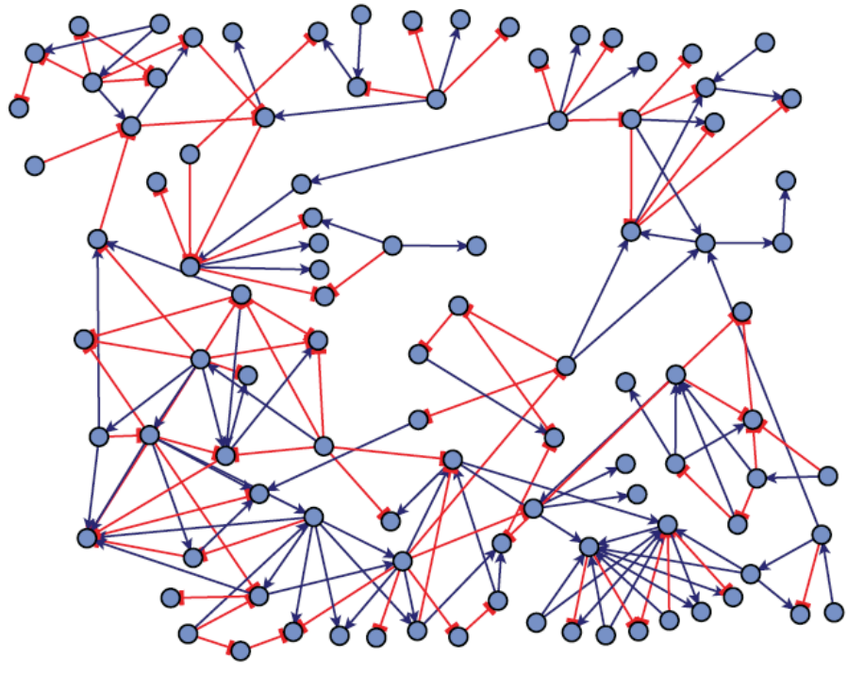
The challenge was structured as 3separate subchallenges with networks of 10, 50, and 100 genes, respectively. For each size, 5 in silico networks were generated.

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

Example network *E.coli*



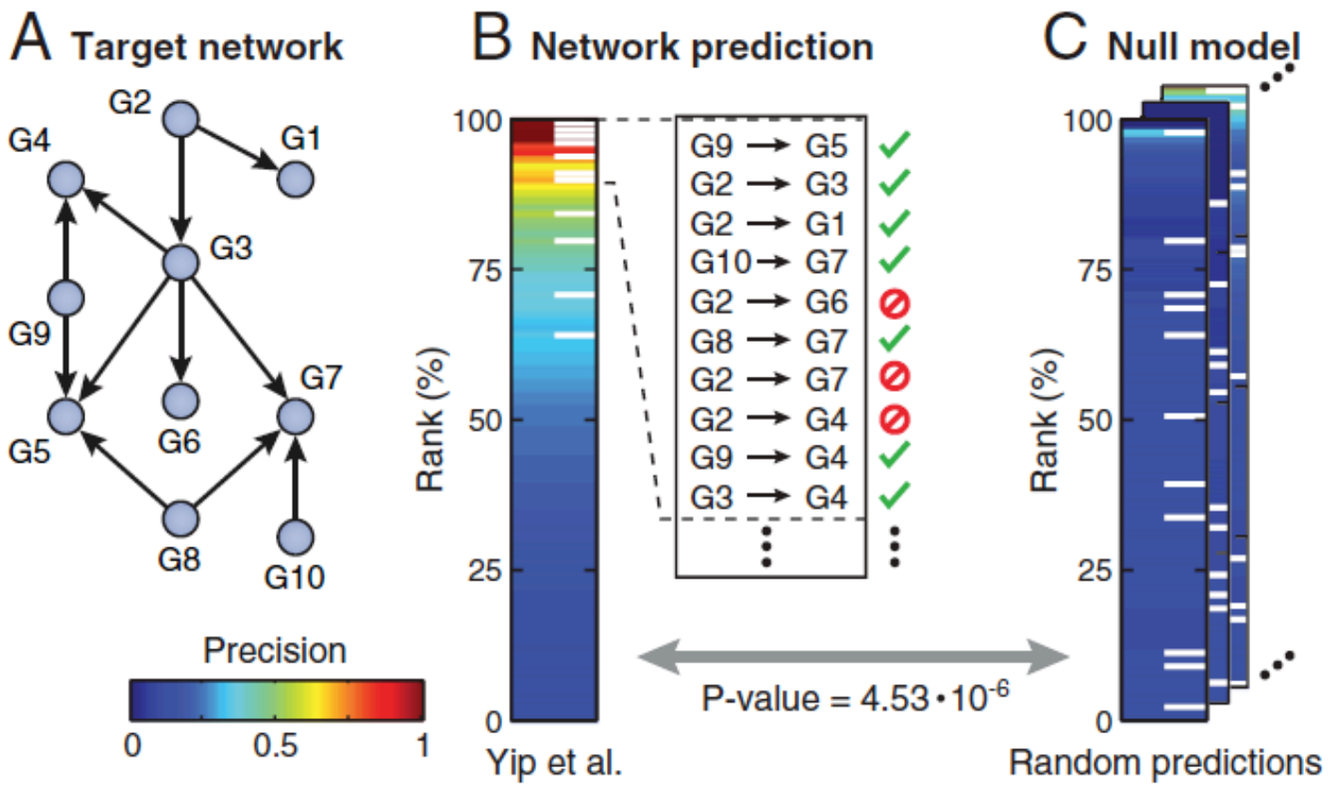
Example network yeast



Marbach et al. PNAS 107, 6286 (2010)

Evaluation of network predictions

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

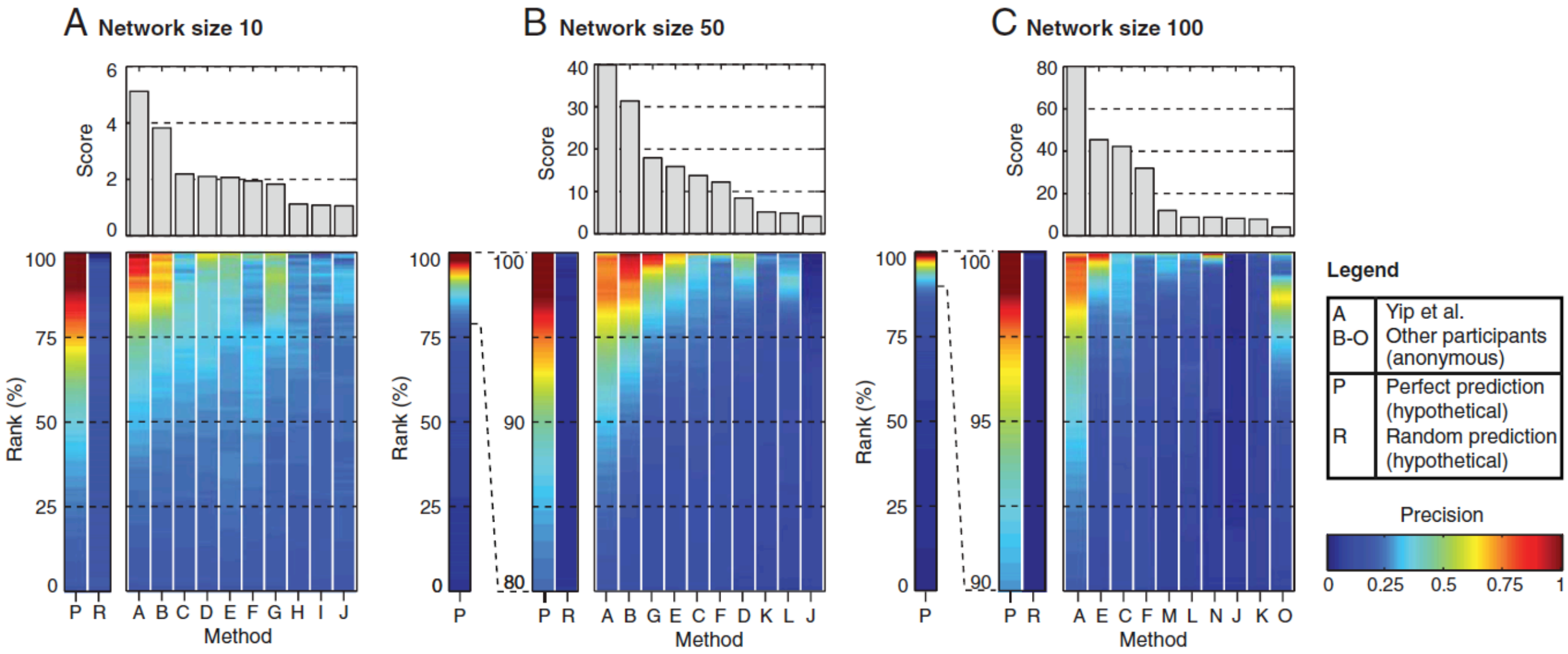


(C) The network prediction is evaluated by computing a P-value that indicates its statistical significance compared to random network predictions.

(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 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)

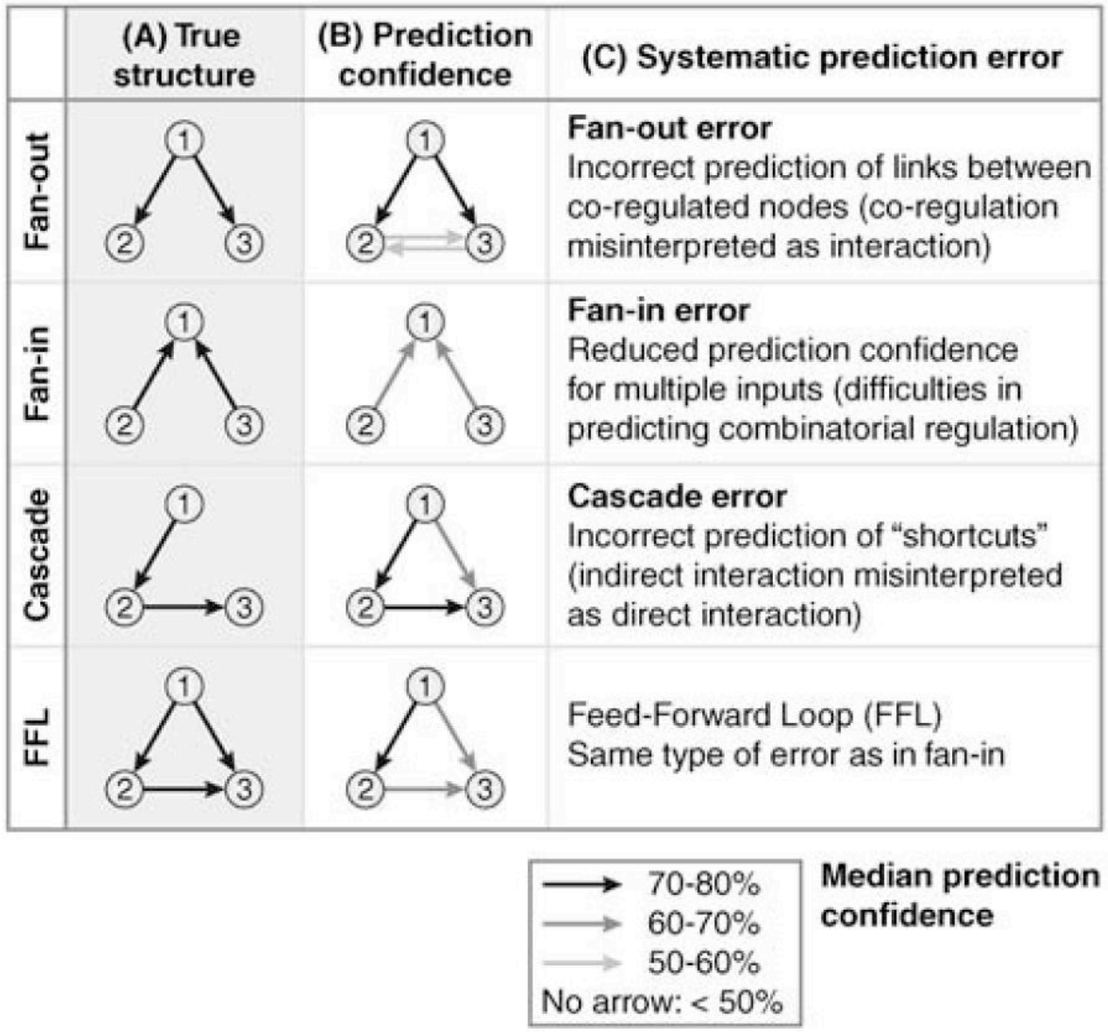
Similar performance on different network sizes

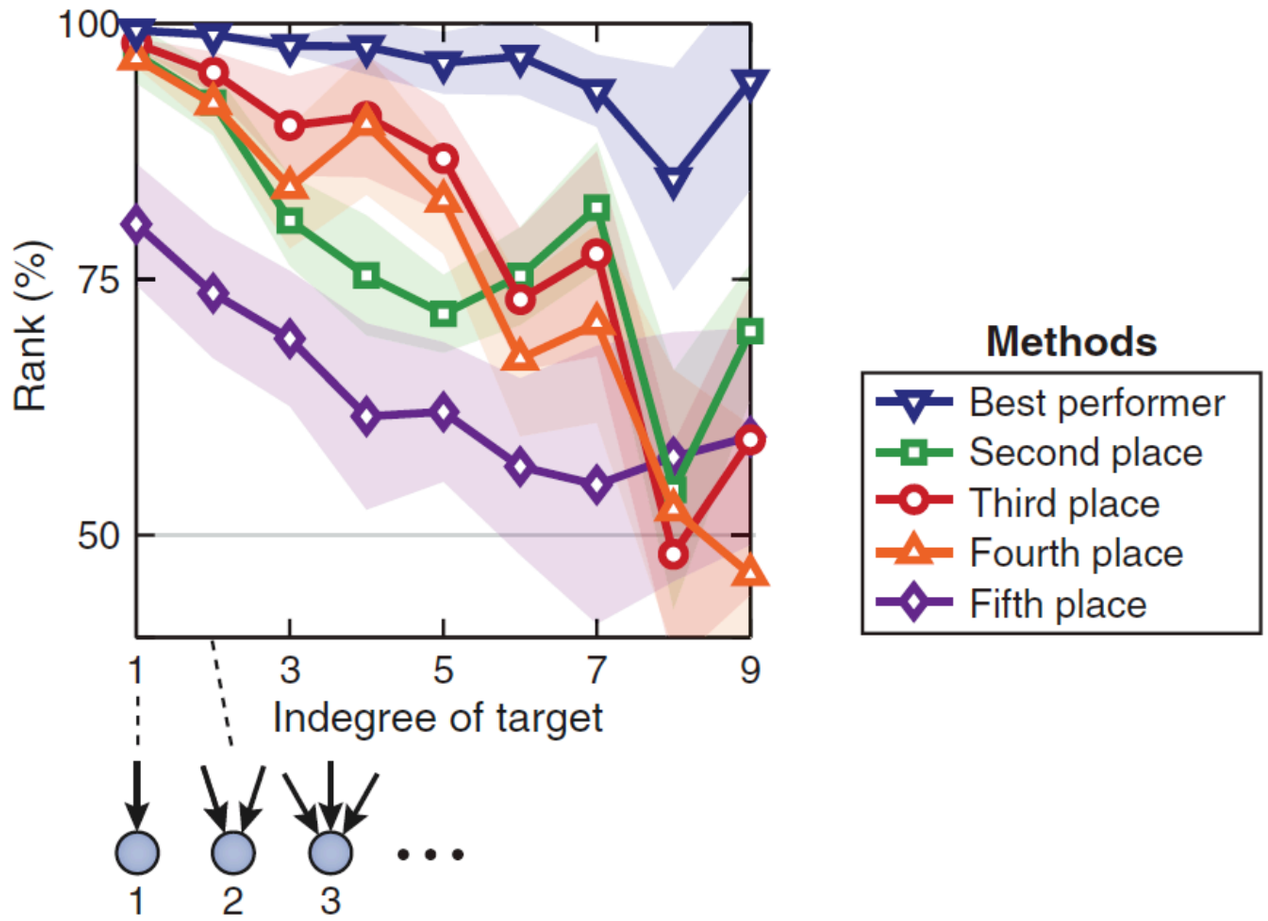


The method by Yip et al. (method A) gave the best results for all 3 network sizes.

Marbach et al. PNAS 107, 6286 (2010)

Error analysis





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 the 3 models were combined.

Mark Gerstein/Yale

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

Cumulative distribution function

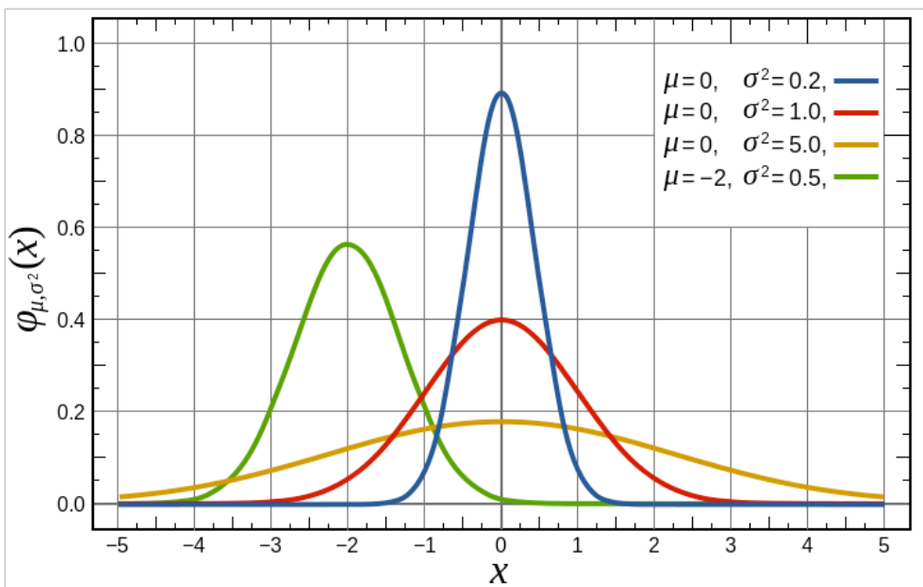
The cumulative distribution function (CDF) describes the probability that a real-valued random variable *X* with a given probability distribution *P* will be found at a value less than or equal to *x*.

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

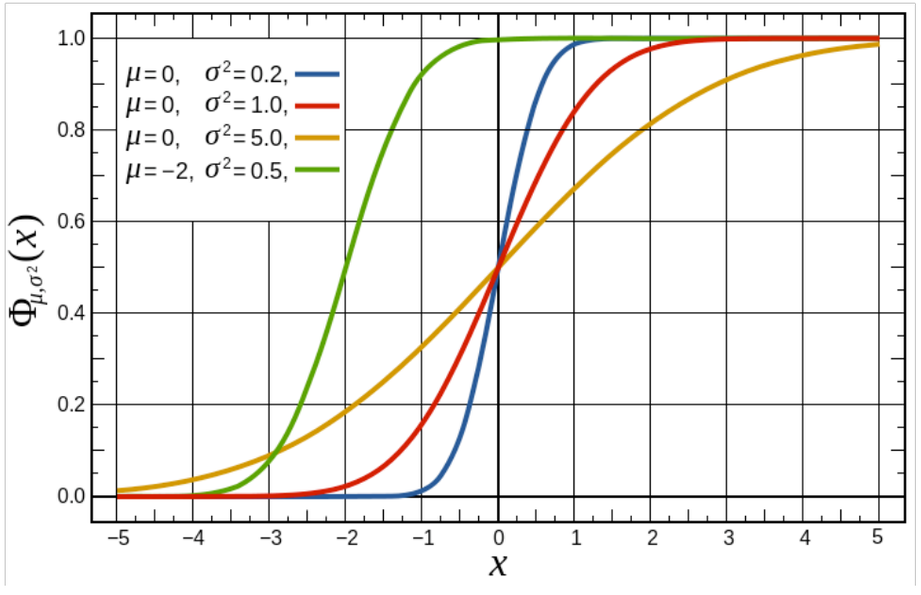
$$F(x) = \int_{-\infty}^{x} f(t)\,dt.$$

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

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



Different normal distributions



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,
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 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.

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

Noise model

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

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

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 $p_{b \to a}$ 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 $b \to a$ 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 $p_{b \square a}$ 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, we use differential equations to model the gene expression rates. The general form is as follows:

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

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

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

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

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

b_{i1} : 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)

Learning ODE models from perturbation time series data

• Batch 1 contains the most confident predictions: all predictions with probability of regulation $p_{b \to a} > 0.99$ 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

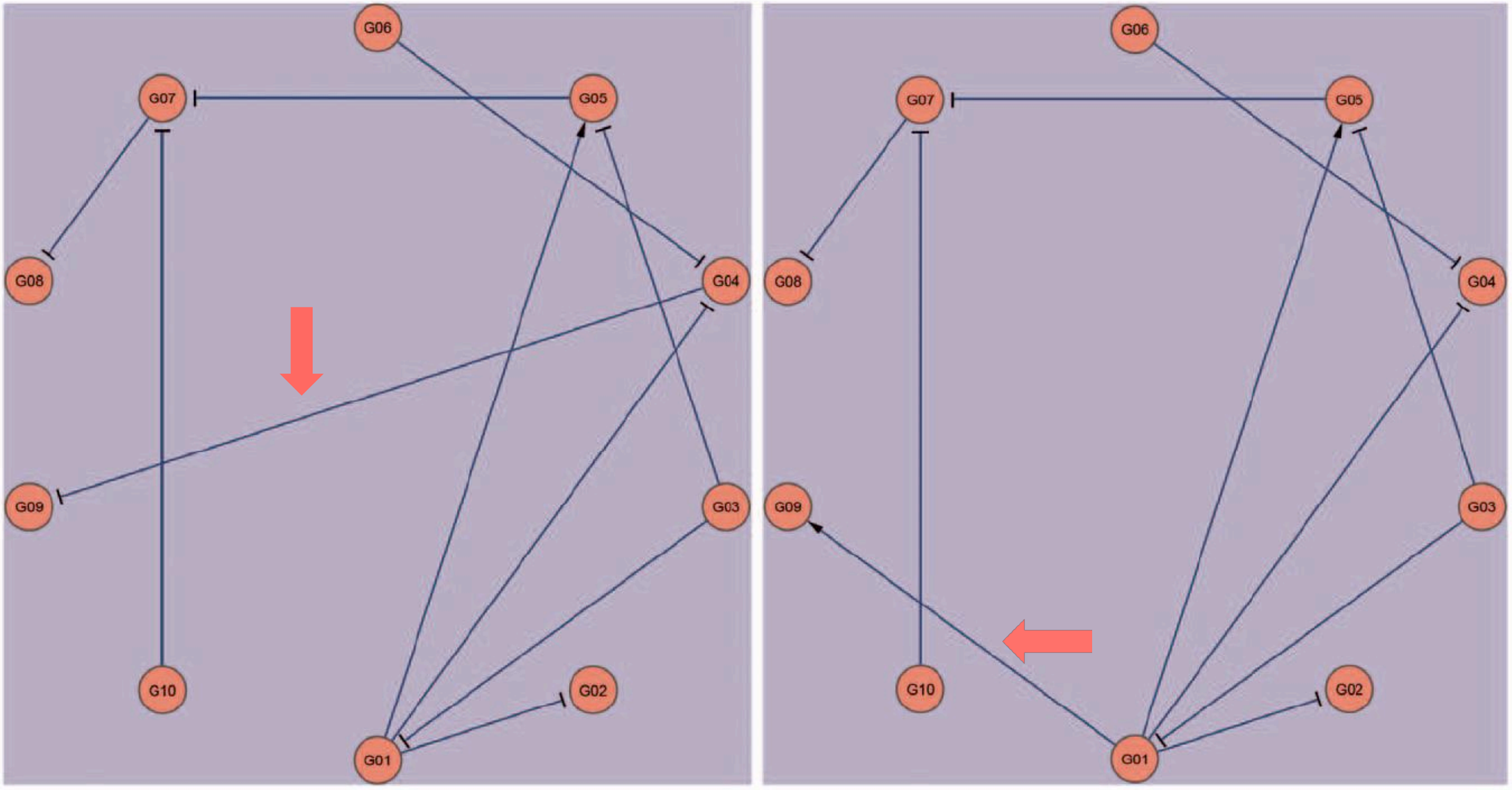


Figure 1. The Yeast1-size10 network. (a) The actual network. (b) Our top-10 predictions.

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

Learning ODE models from perturbation time series data

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:

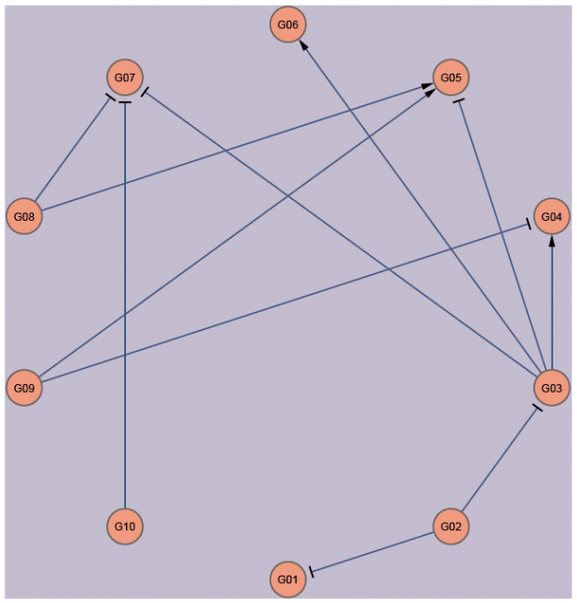
A network with 10 nodes has 10 x 9 possible edges

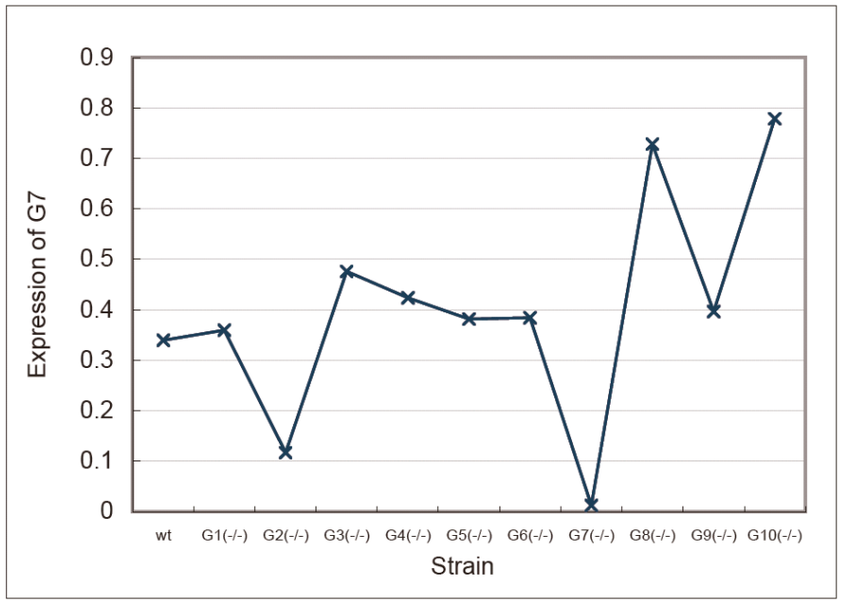
Batch 1 already contains many of the correct edges (7/11 – 8/22). The majority of the high-confidence predictions are correct (7/11 – 8/12).

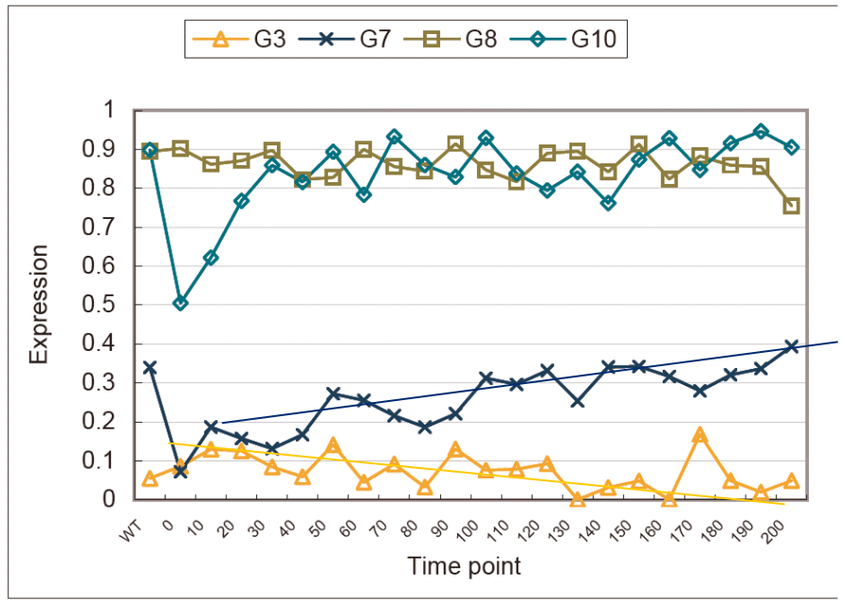
Batch 7 contains only 1 correct edge for the E.coli-like network, but 9 or 10 correct edges for the Yeast-like network.

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

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 G
which is difficult to detect.

However the right panel suggests that the increased expression of G7 over time
anti-correlated with the decreased level of G3
→ This link was detected by the ODE-models in batch 2

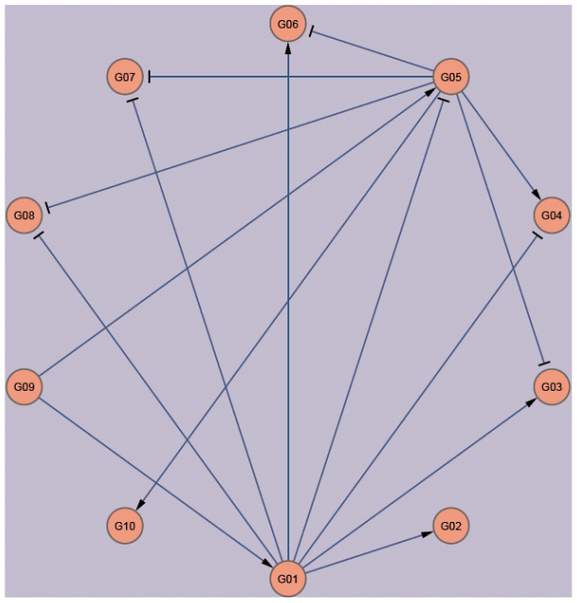
Yip et al. PloS ONE 5:e8121 (2010)

Learning ODE models from perturbation time series data

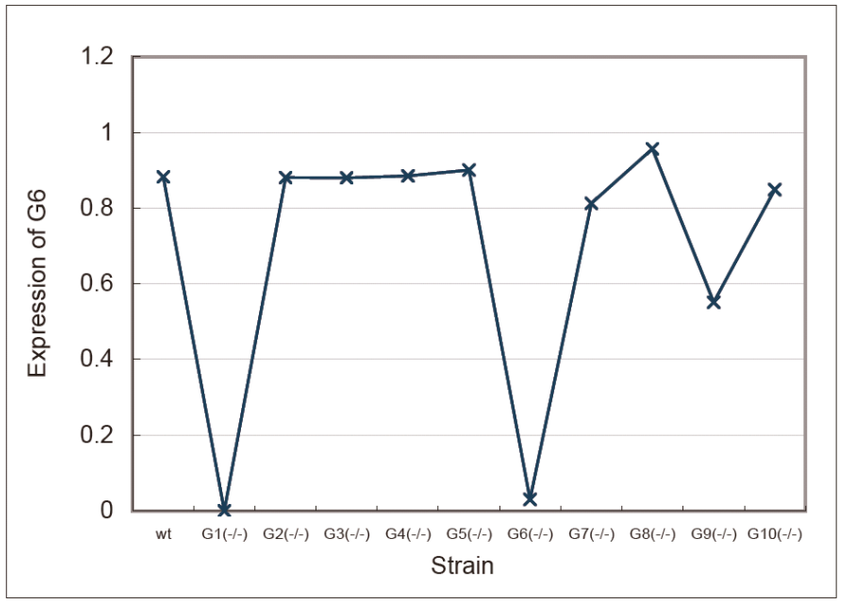
Another case:
Left: G6 is activated by G1 and suppressed by G5. G1 also suppresses G5. G1 therefore has 2 functions on G6. When G1 is expressed, deleting G5 (midd has no effect.

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

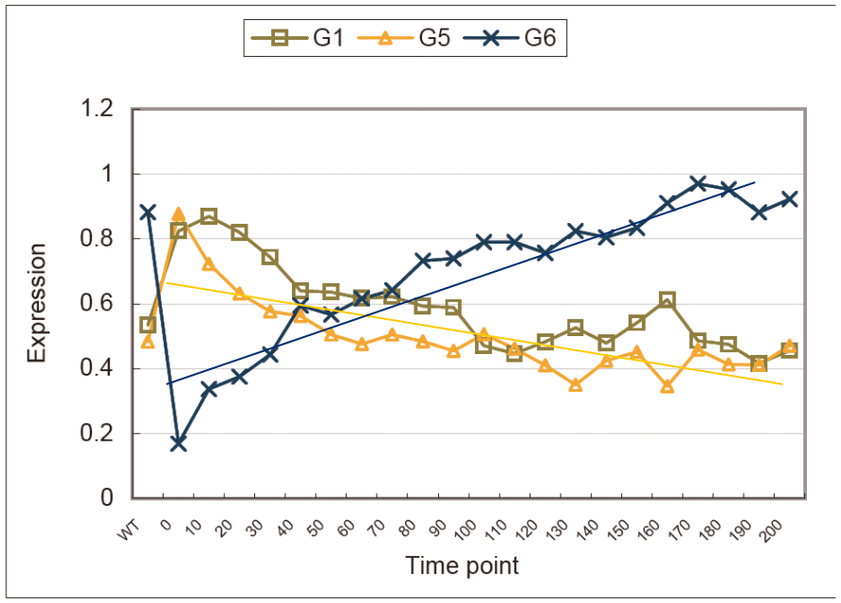
But G5 is also anti-correlated with G6 $\rightarrow$ evidence for inhibitory role of G5.



(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)

Quorum sensing of *Vibrio fischeri*

This luminescent bacterium exists in small amounts in the ocean and in large amount in isolated areas such as the light organs of squid.

When in small concentrations of cells, *V. fischeri* does not give off light,
but in high cell density these bacteria emit a blue-green light.

This cell density-dependent control of gene expression is activated by auto-induction that involves the coupling of a transcriptional activator protein with a signal molecule (autoinducer) that is released by the bacteria into its surrounding environment.

In the ocean, the population density of *V. fischeri* is only about 10^2 cells/ml.

Exporting the autoinducer from the bacteria into this low concentration
of cells is not enough to cause the luminescence genes to be activated.

However, inside the light organ of a squid for example,
the cell concentration is about 10^{10} cells/ml.

At such high concentrations, the autoinducer causes the bacteria to emit light

https://www.bio.cmu.edu/courses/03441/TermPapers/99TermPapers/Quorum/vibrio_fischeri.html

Quorum sensing of *Vibrio fischeri*

V. fischeri has a microbial **symbiotic relationship** with the squid Euprymna scolopes. The light organ of the squid provides the bacteria all of the nutrients that they need to survive. The squid benefits from the bacteria's quorum sensing and bioluminescence abilities.

During the day, the squid keeps the bacteria at lower concentrations by expelling some of them into the ocean during regular intervals.
At night however, the bacteria are allowed to accumulate to about 10^{10} cells/ml so that they will emit blue-green light.

This is perfect for the squid because it is a night feeder.

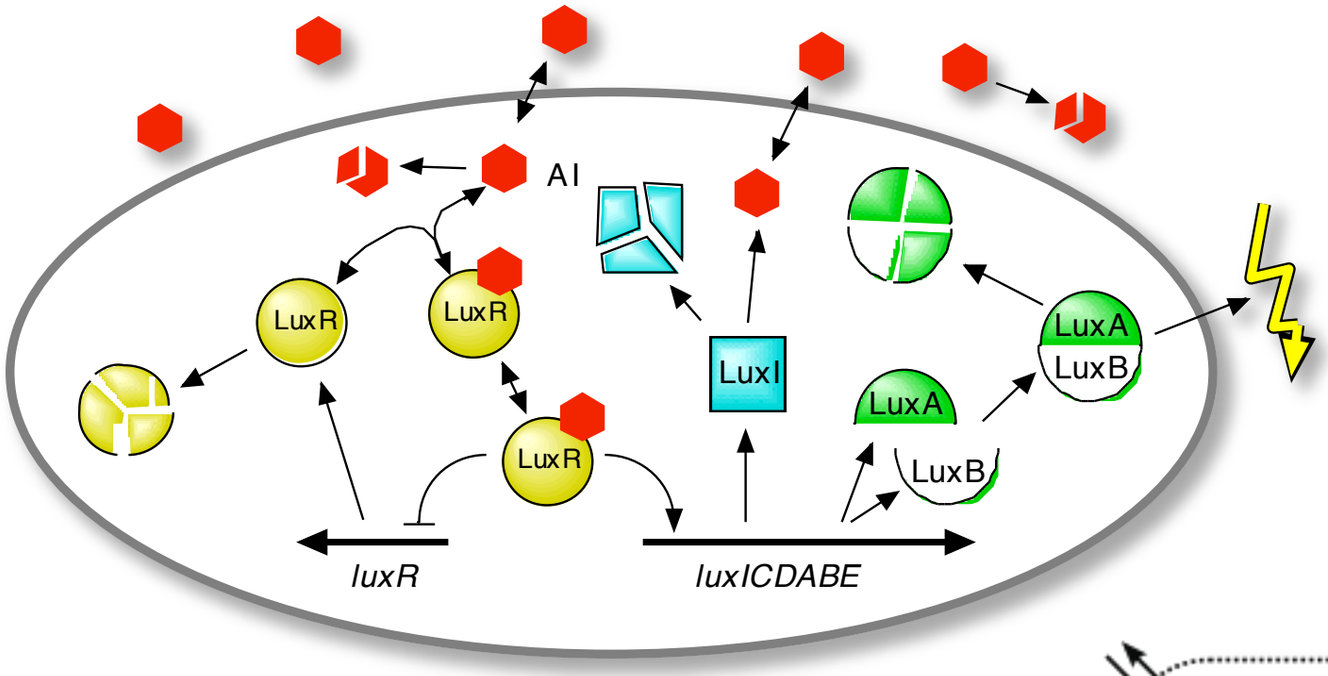
In the **moonlight**, the swimming squid would normally cast a **shadow** beneath itself making it a perfect target for squid-eating organisms.

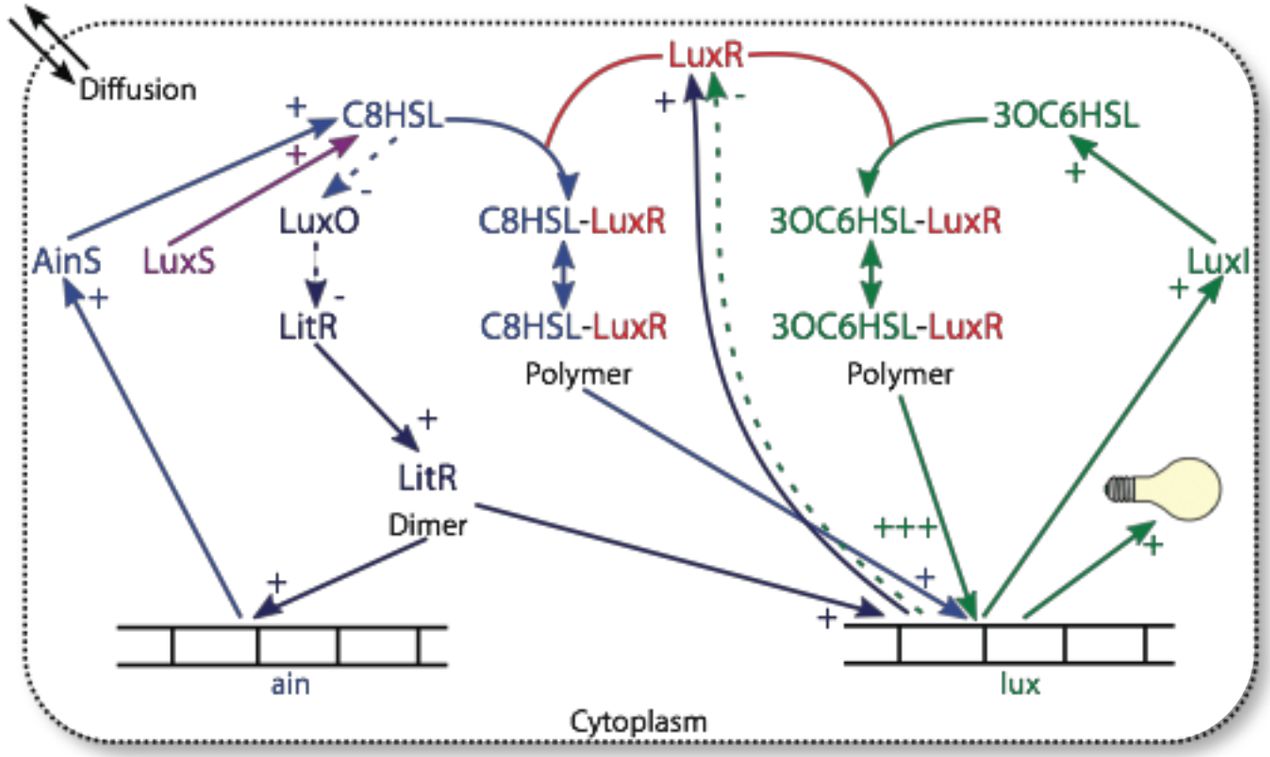
However, the bacterial glow will counter the shadowing effect the moon makes and mask the squid from its predators.

In the morning, the squid expels some bacteria into the ocean to a concentration where they will not generate light anymore so as to conserve energy.

http://www.mocha.edu/courses/c3441STermPapers/Quorum/vibrio_fischeri.html

Quorum sensing of *Vibrio fischeri*





Boolean Networks

"Blackboard explanations" often formulated as **conditional transitions**

- "If LuxI is present, then AI will be produced…"
- "If there is AI and there's no LuxR:AI bound to the genome, then LuxR will be expressed and complexes can form…"
- "If LuxR:AI is bound to the genome, then LuxI is expressed…"

Simplified mathematical **description** of the dependencies:

Densities of the species	<=>	discrete states: on/off, 1/0
Network of dependencies	<=>	condition tables
Progress in time	<=>	discrete propagation steps

Boolean Networks II

State of the system: described by **vector** of **discrete** values

$$S_i = \{0, 1, 1, 0, 0, 1, \ldots\}$$

$$S_i = \{x_1(i),\ x_2(i),\ x_3(i),\ \ldots\}$$

fixed number of species with **finite number** of states each

→ finite number of system states

→ periodic trajectories

→ **periodic** sequence of states = **attractor**

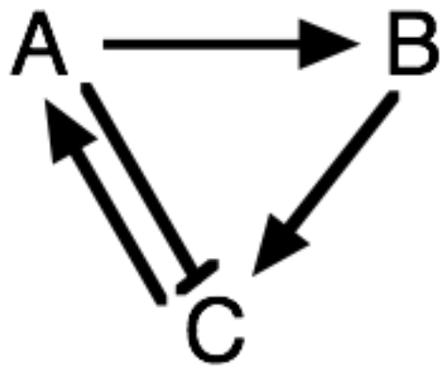
→ all states leading to an attractor = **basin of attraction**

Propagation:

$$S_{i+1} = \{x_1(i+1),\ x_2(i+1),\ x_3(i+1),\ \ldots\}$$

$$x_1(i+1) = f_1(x_1(i), x_2(i), x_3(i), \ldots)$$ with f_i given by condition tables

A Small Example



State vector $S = \{A, B, C\} \rightarrow$ 8 possible states

Conditional evolution:

A is on if C is on

A_{i+1}	C_i
0	0
1	1

A activates B

B_{i+1}	A_i
0	0
1	1

C is on if (B is on && A is off)

C_{i+1}	A_i	B_i
0	0	0
1	0	1
0	1	0
0	1	1

assume here that inhibition through A is stronger than activation via B

Start from {A, B, C} = {1, 0, 0}

#	S_i	A	B	C
0	S_0	1	0	0
1	S_1	0	1	0
2	S_2	0	0	1
3	$S_3 = S_0$	1	0	0

periodic orbit of length 3

Test the Other States

Test the other states

#	A	B	C
0	1	1	1
1	1	1	0
2	0	1	0
3	0	0	1
4	1	0	0
5	0	1	0

A_{i+1}	C_i
0	0
1	1

B_{i+1}	A_i
0	0
1	1

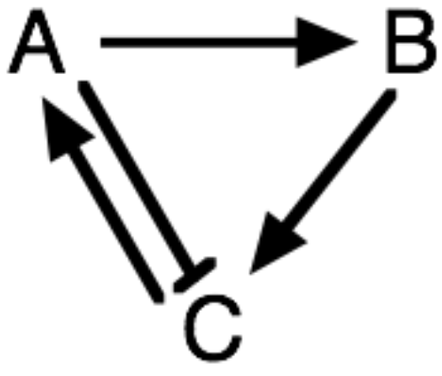
C_{i+1}	A_i	B_i
0	0	0
1	0	1
0	1	0
0	1	1

#	A	B	C
0	1	0	1
1	1	1	0

#	A	B	C
0	0	1	1
1	1	0	1

Same attractor as before:
100 → 010 → 001 → 100

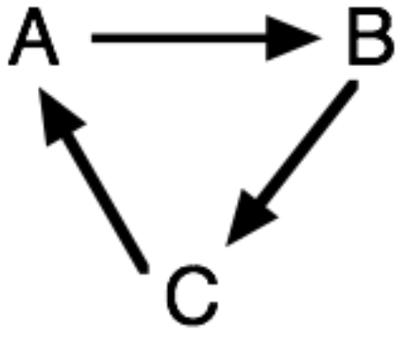
also reached from:
110, 111, 101, 011



#	A	B	C
0	0	0	0
1	0	0	0

→ **Either all off or stable oscillations**

A Knock-out Mutant



A_{i+1}	C_i
0	0
1	1

B_{i+1}	A_i
0	0
1	1

C_{i+1}	B_i
0	0
1	1

Attractors:

#	A	B	C
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0

#	A	B	C
0	1	1	0
1	0	1	1
2	1	0	1
3	1	1	0

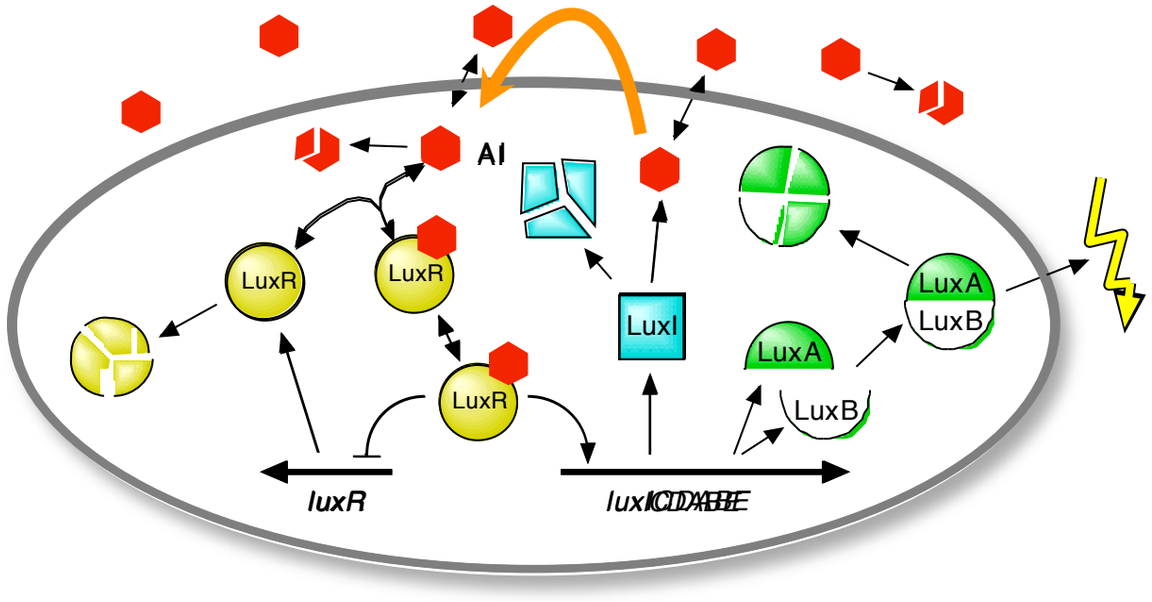
#	A	B	C
0	1	1	1
1	1	1	1

#	A	B	C
0	0	0	0
1	0	0	0

no feedback

→ no stabilization, network just "rotates"

Boolean Network of QS



Minimum set of species:

LuxR, AI, LuxR:AI, LuxR:AI:genome, LuxI

Here: Light signal (LuxAB) α LuxI

Condition tables: describe the state of a species in the next step given the current states of all relevant species.

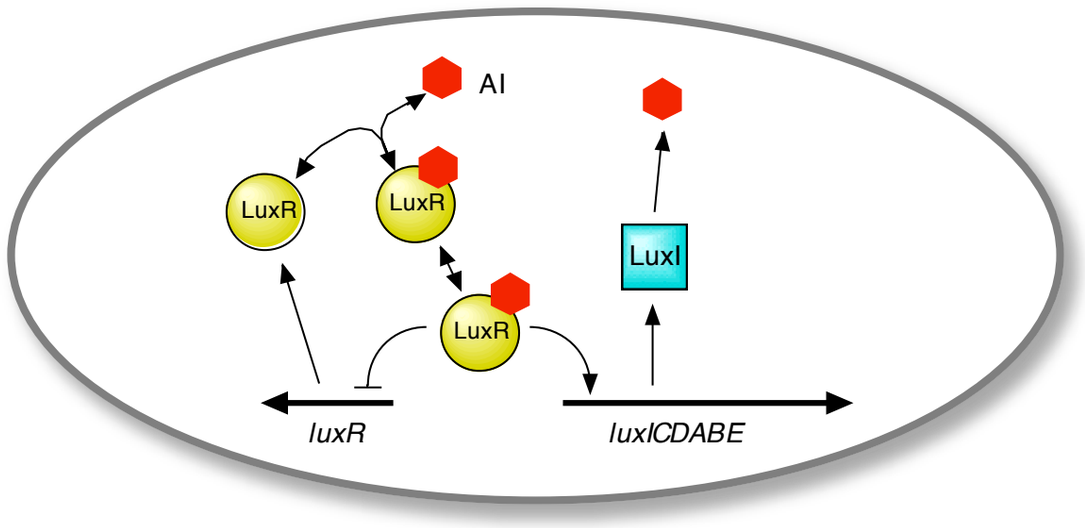
LuxI	LuxR:AI:Genome
0	0
1	1

How does LuxI depend on LuxR:AI:Genome?

LuxR:AI:Genome	LuxR:AI
0	0
1	1

How does LuxR:AI:Genome depend on LuxR:AI?

Condition Tables for QS II



LuxR	LuxR	AI	LuxR:AI:Genome	
1	0	0	0	When LuxR:AI:Genome is empty, LuxR is produced in next step
1	1	0	0	
1	0	1	0	
1	1	1	0	
0	0	0	1	
1	1	0	1	Comment: LuxR present, no AI available
0	0	1	1	
0	1	1	1	LuxR present, binds AI in next step, no LuxR is produced because LuxR:AI:Genome inhibits LuxR productio

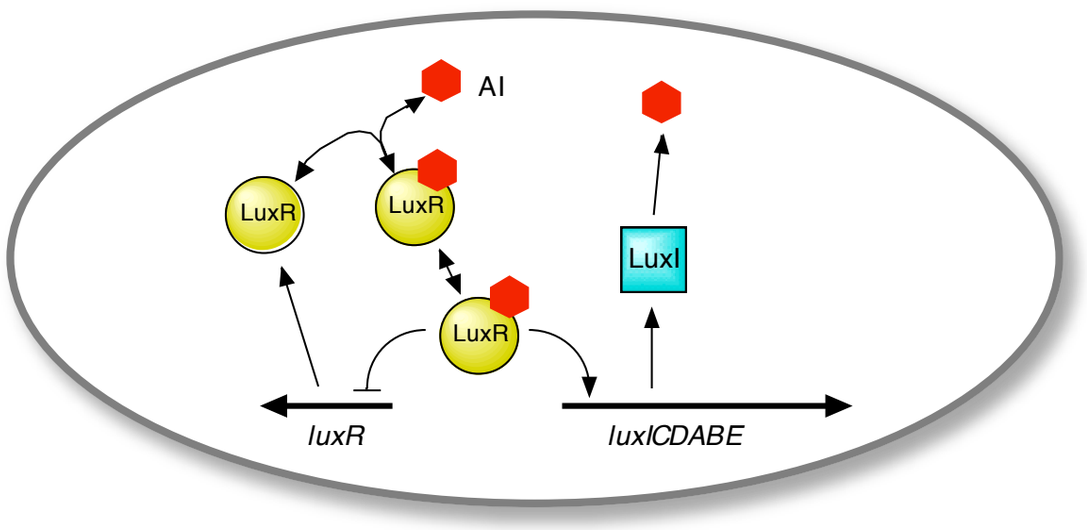
LuxR:AI	LuxR	AI	LuxR:AI:Genome
0	0	0	0
0	1	0	0
0	0	1	0
1	1	1	0
0	0	0	1
0	1	0	1
0	0	1	1
1	1	1	1

→

LuxR:AI	LuxR	AI	LuxR:AI:Genome
0	x	x	x
1	1	1	x

Note: no dissociation
(LuxR:AI:Genome → LuxR:AI + Genome)
only degradation of AI in this model
LuxR:AI:Genome → LuxR + Genome

Condition tables for QS III



AI	LuxR	AI	LuxI
0	0	0	0
0	1	0	0
1	0	1	0
0	1	1	0
1	0	0	1
1	1	0	1
1	0	1	1
1	1	1	1

→

AI	LuxR	AI	LuxI
1	x	x	1
0	x	0	0
1	0	1	0
0	1	1	0

Scanning for Attractors

States of *V. fischeri* QS system mapped onto integers

{LuxR (LR), LuxR:AI (RA), AI, LuxR:AI:Genome (RAG), LuxI (LI)}

= {1, 2, 4, 8, 16} - current state can be interpreted as binary number!

For each **attractor**:

- periodic orbit and its length (period)
- basin of attraction and its relative size (32 states in total)

→ how likely will the system end in each of the attractors?

Attractor 1: orbit: 1 → period 1

states: 0, 1 → size 2, 2/32 = 6.25 %

start from state 0:

#	LR	RA	AI	RAG	LI	-	state
0	.	.	.	.	.	-	0
1	X	.	.	.	.	-	1
2	X	.	.	.	.	-	1

<= attractor

Scanning for Attractors II

Attractor 2: orbit: 3, 9, 17, 5 → period 4

states: 2, 3, 5, 8, 9, 16, 17 → size 7, 21.9 %

start from state 8:

#	LR	RA	AI	RAG	LI	-	state
0	.	.	.	X	.	-	8
1	.	.	.	.	X	-	16
2	X	.	X	.	.	-	5
3	X	X	.	.	.	-	3
4	X	.	.	X	.	-	9
5	X	.	.	.	X	-	17
6	X	.	X	.	.	-	5

attractor

averaged occupancies in this periodic orbit:

LR	RA	AI	RAG	LI
4/4 = 1	1/4 = 0.25	1/4 = 0.25	1/4 = 0.25	1/4 = 0.25

Attractors III

Attractor 3: period 4, basin of 16 states → 50 %

#	LR	RA	AI	RAG	LI	– state0
	.	X	X	.	.	- 61
	.	X	X	X	.	- 142
	.	.	X	X	X	- 283
	.	.	X	.	X	- 20

Attractor 4: period 4, basin of 4 states → 12.5 %

#	LR	RA	AI	RAG	LI	- state0
	X	X	X	.	.	- 71
	X	X	.	X	.	- 112
	X	.	.	X	X	- 253
	X	.	X	.	X	- 21

Attractor 5: period 2, basin of 3 states → 9.4 %

#	LR	RA	AI	RAG	LI	- state0
	X	.	X	X	.	- 131
	.	X	.		X	- 18

Classifying the Attractors

→ Interpret the system's behavior from the properties of the attractors

Attractor	period	basin size	<LuxR>	<LuxR:AI>	<AI>	<LuxR:AI:Gen>	<LuxI>
1	1	6.25 % (2)	1	0	0	0	0
2	4	21.9% (7)	1	0.25	0.25	0.25	0.25
3	4	50 % (16)	0	0.5	1	0.5	0.5
4	4	12.5 % (4)	1	0.5	0.5	0.5	0.5
5	2	9.4% (3)	0.5	0.5	0.5	0.5	0.5

Three **regimes**:

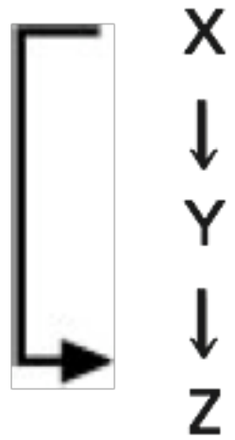
dark: LuxI = 0 — free LuxR, no AI

intermediate: LuxI = 0.25 — free LuxR + little AI

bright: LuxI = 0.5 — little free LuxR (0.24) + much AI (0.85)

The Feed-Forward-Loop

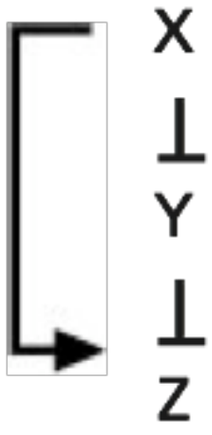
External signal determines state of X
→ response Z for short and long signals X



condition tables:

Y	X
0	0
1	1

Z	X	Y
0	0	0
0	0	1
0	1	0
1	1	1



Y	X
1	0
0	1

Z	X	Y
0	0	0
0	0	1
1	1	0
0	1	1

Signal propagation
Left column: external signal

Response to signal X(t)

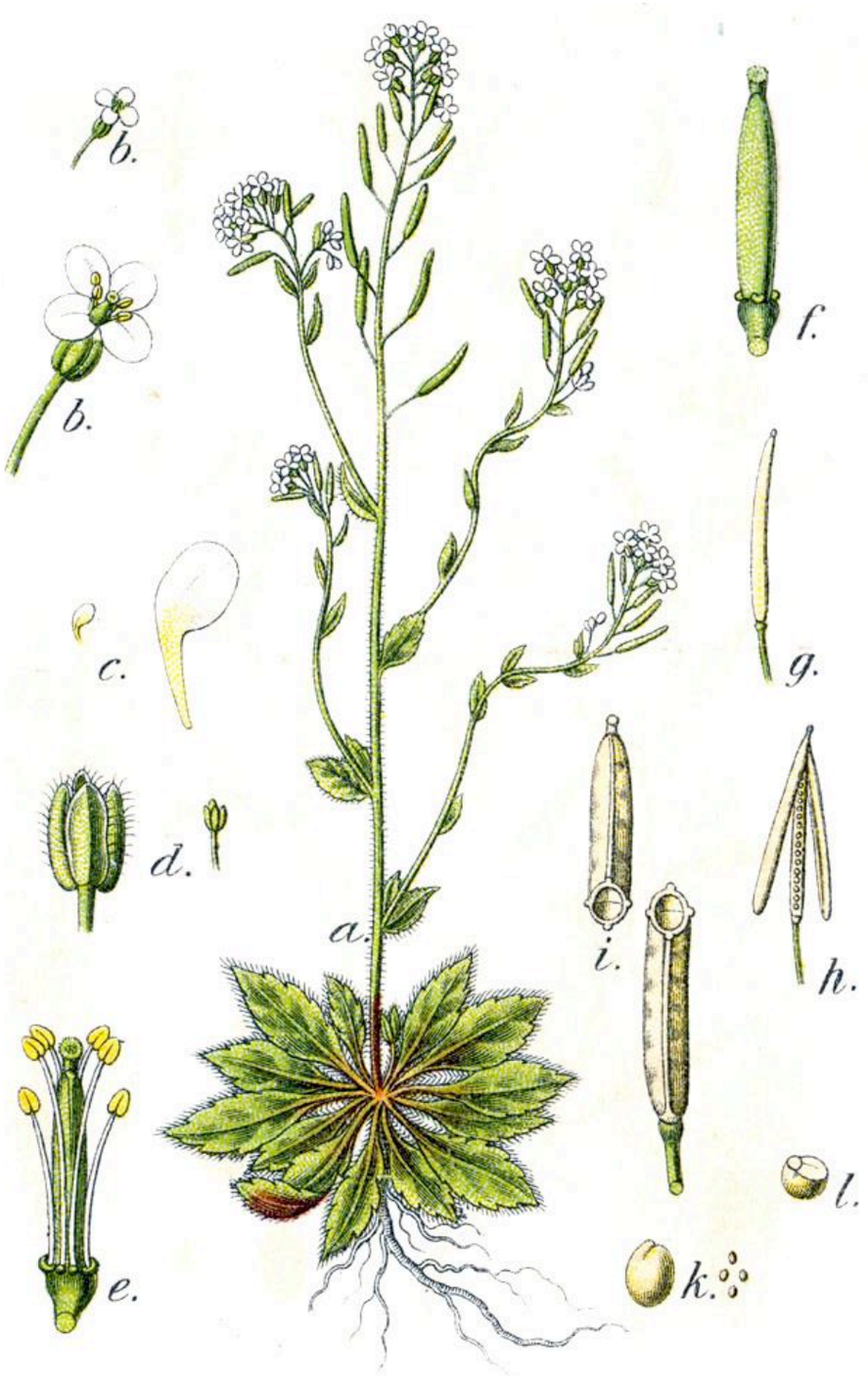
X	Y	Z
0	0	0
1	0	0
0	1	0
0	0	0
1	0	0
1	1	0
1	1	**1**
0	1	**1**
0	0	**0**
0	0	0

Short Signal

Long signal

X	Y	Z
0	1	0
1	1	0
0	0	0
0	1	0
1	1	0
1	0	0
1	0	**1**
0	0	**1**
0	1	**1**
0	1	0

The *A. thaliana* Flowering Network



Model organism in genomics:

- small, convenient to grow
- completely sequenced (2000): 125 Mbp
- can be easily mutated

also see: Arabidopsis Information Resource (TAIR)@ www.arabidopsis.org/

images from wikimedia

Dynamics of the Genetic Regulatory Network for *Arabidopsis thaliana* Flower Morphogenesis

LUIS MENDOZA AND ELENA R. ALVAREZ-BUYLLA*

Lab. de Genética Molecular y Evolución, Instituto de Ecología, UNAM, Ap. Postal 70-275, *Coyoacán, D.F. CP*04510, *Mexico*

(*Received on* 26 *September* 1997, *Accepted in revised form on* 3 *March* 1998)

We present a network model and its dynamic analysis for the regulatory relationships among 11 genes that participate in *Arabidopsis thaliana* flower morphogenesis. The topology of the network and the relative strengths of interactions among these genes were based from published genetic and molecular data, mainly relying on mRNA expression patterns under wild type and mutant backgrounds. The network model is made of binary elements and we used a particular dynamic implementation for the network that we call semi-synchronic. Using this method the network reaches six attractors; four of them correspond to observed patterns of gene expression found in the floral organs of *Arabidopsis* (sepals, petals, stamens and carpels) as predicted by the ABC model of flower morphogenesis. The fifth state corresponds to cells that are not competent to flowering, and the sixth attractor predicted by the model is never found in wild-type plants, but it could be induced experimentally. We discuss the biological implications and the potential use of this network modeling approach to integrate functional data of regulatory genes of plant development.

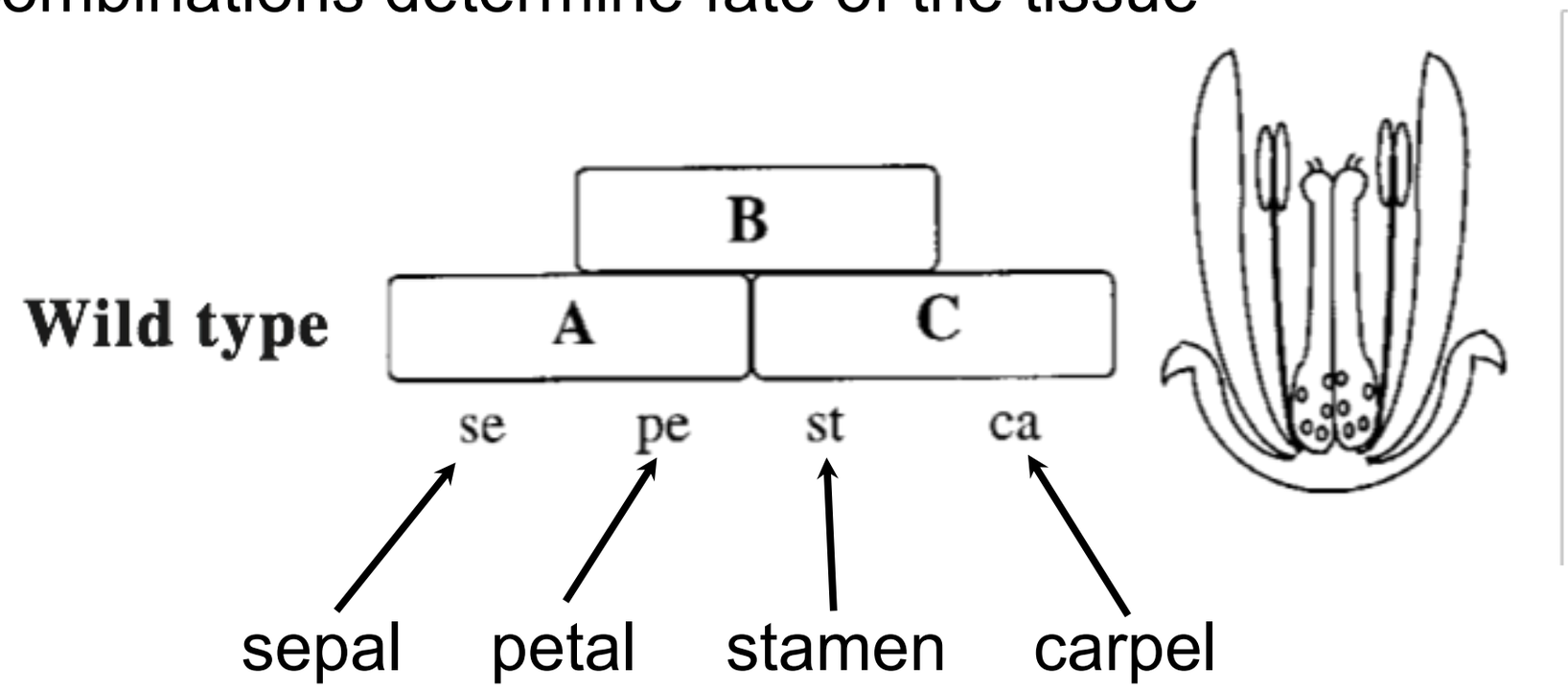
J. theor Biol. **193** (1998) 307

The ABC Model

Coen, Meyerowitz (1991):

three different activities A, B, and C, active in two adjacent whorls,
mutual inhibition of A and C

→ combinations determine fate of the tissue



Related genes:

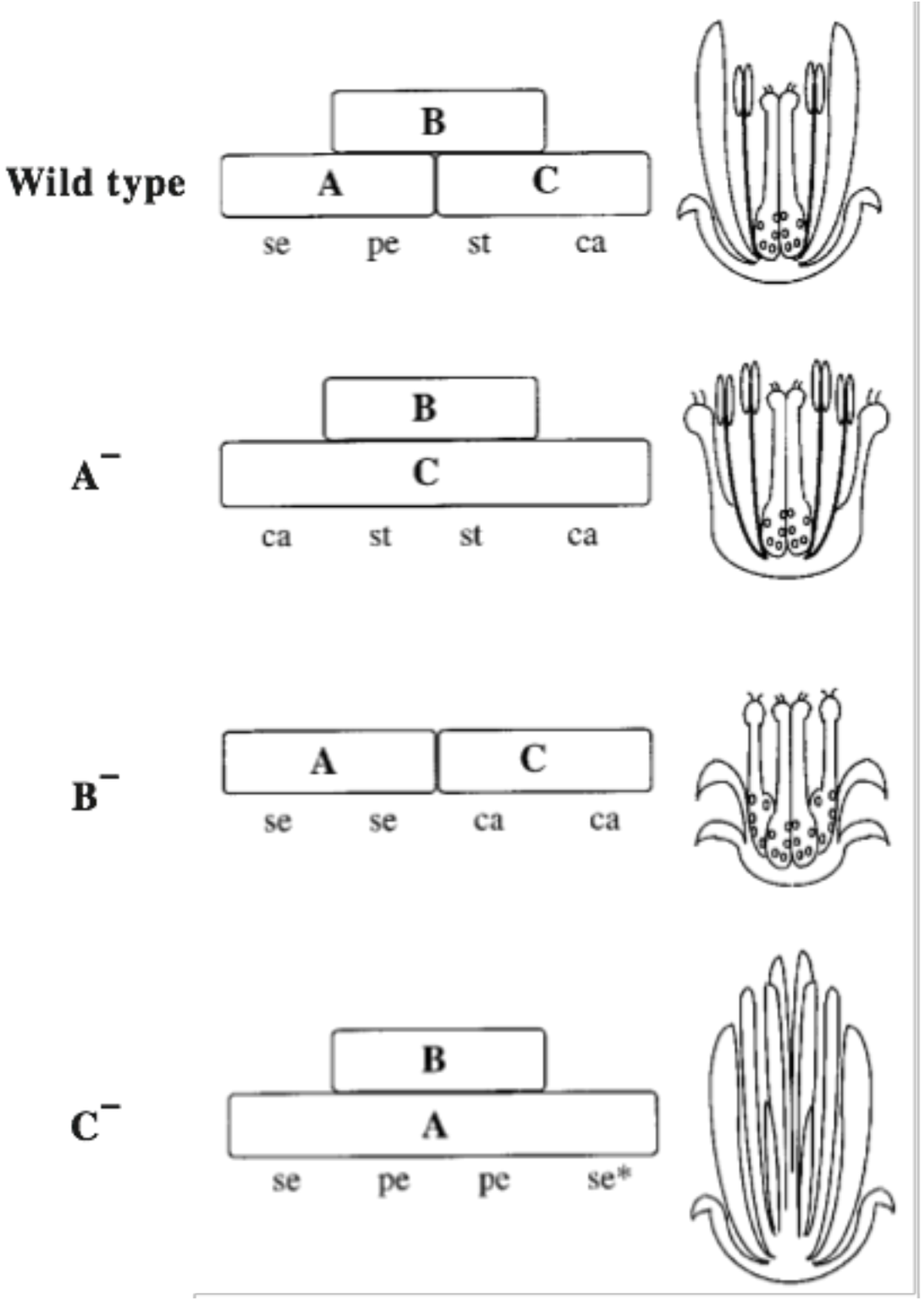
A:
APETALA1 (*AP1*)

B:
APETALA3 (*AP3*),
PISTILATA (*PI*)

C:
AGAMOUS (*AG*)

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

ABC Mutants

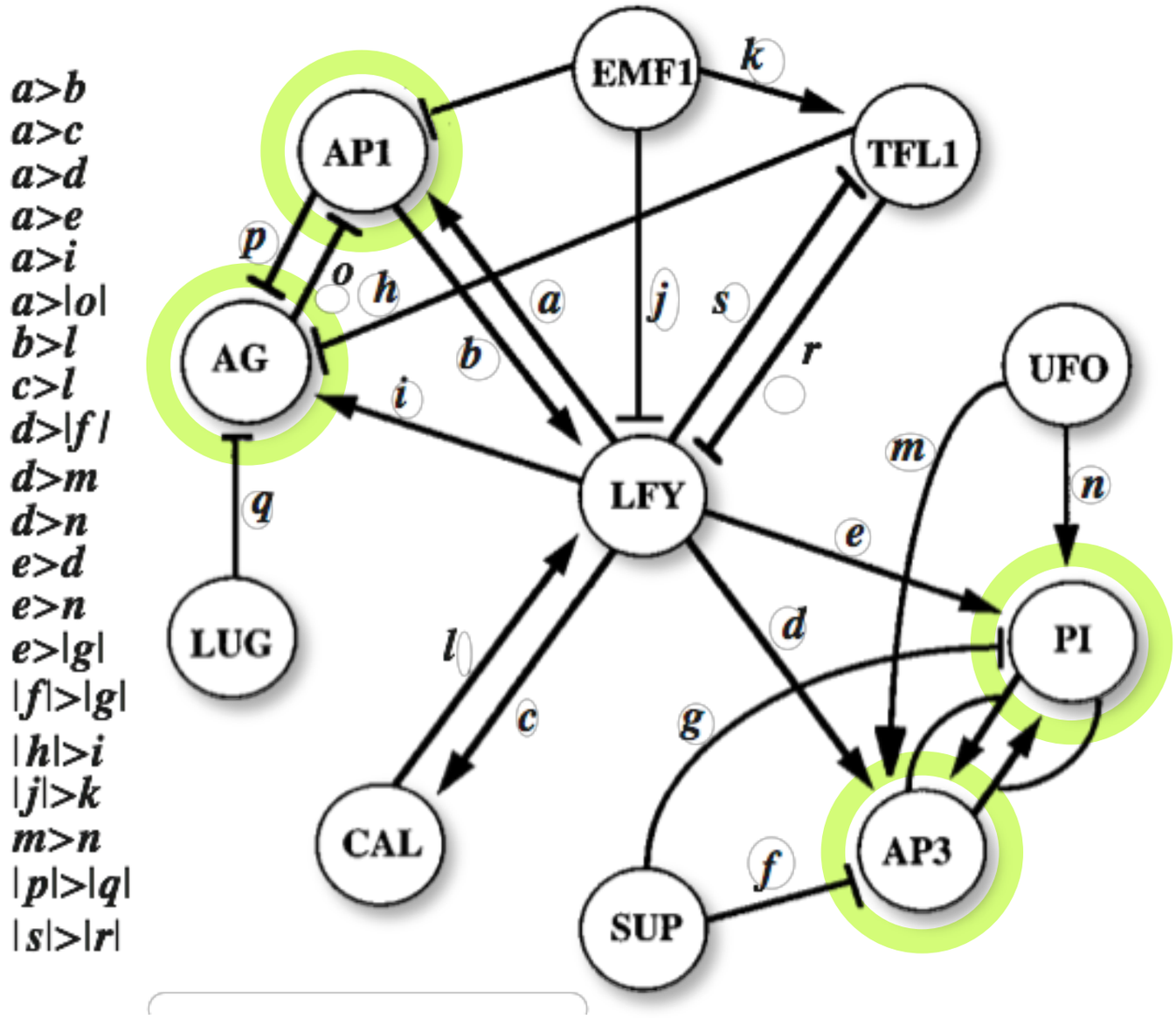


If any of the 3 functions (activities) is missing, the flowers have different tissue combinations.

se = sepals,
pe = petals,
st = stamens,
ca = carpels,
se* = se, pe, pe

The Network Model

11 genes (including the four ABC genes)



inequalities denote the relative weights of the interactions

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

Model Implementation

Here: Boolean model with **weighted** interactions

Propagate state vector $\boldsymbol{x} = \{x_1, x_2, \ldots, x_{11}\}$ by:

$$x_i(t+1) = \mathbf{H}\left(\sum_{j=1}^{N} w_{ij} x_j(t) - \theta_i\right)$$

Heavyside step function: $\mathbf{H}(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x \leqslant 0 \end{cases}$

Weights w_{ij} and threshold θ_i are not known exactly
→ choose integers for simplicity
→ positive for activation, negative for inhibition

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

The Numbers

*EMF*1, *TFL*1, *LFY*, *AP*1, *CAL*, *LUG*, *UFO*, *BFU*,*AG*, *AP*3, *PI* and *SUP*.

$$\mathbf{W} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & -1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & -2 & 1 & -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & -2 \\
0 & 0 & 4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix}
0 \\ 0 \\ 3 \\ -1 \\ 1 \\ 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0
\end{bmatrix}.$$

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

Synchronous vs. Asynchronous

Synchronous propagation (Kauffman (1969)):

→ update all species **simultaneously**

→ biological problem: do all genes respond at exactly the same time?

Asynchronous propagation (Thomas (1991)):

→ update one species after the other **in chosen order**

→ order of update may influence dynamic gene activation patterns

Semi-synchronic propagation (Mendoza (1998)):

→ split genes in groups:

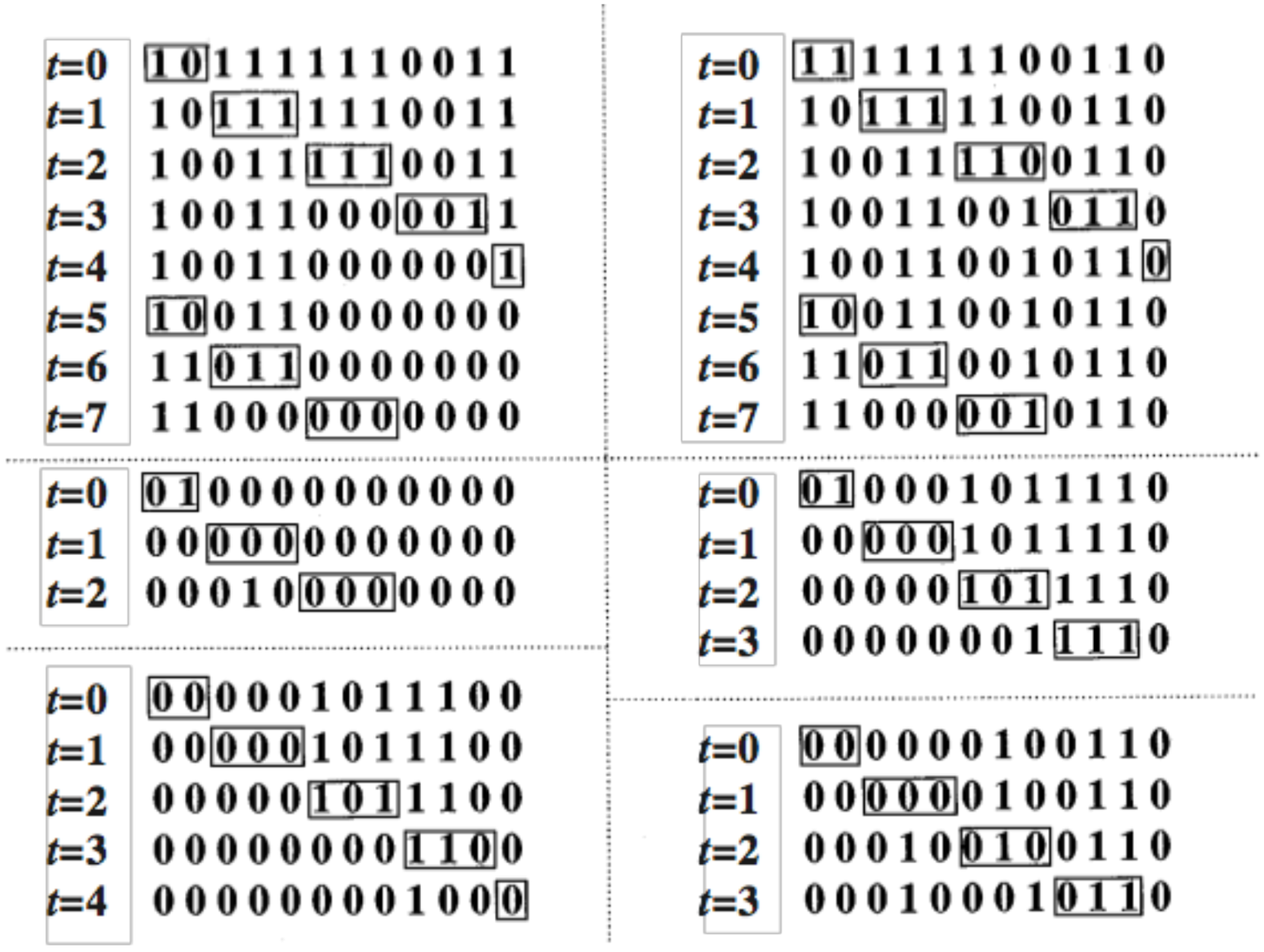
→ synchronous within group, one group after the other

→ base order of groups upon experimental data (it's still a "choice")

*EMF*1, *TFL*1 → *LFY, AP*1, *CAL* → *LUG, UFO, BFU* → *AG, AP*3, *PI* → *SUP*

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

Some Example Patterns



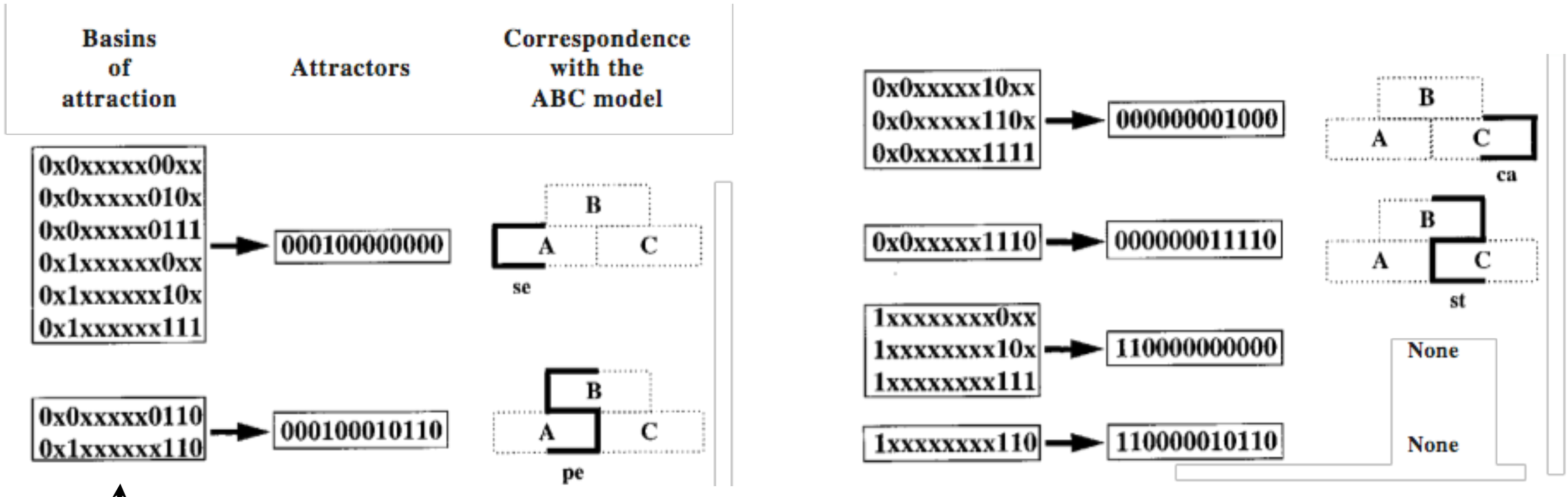
Exhaustive search: start from **all** 2^{12} = 4096 possible **initial** states,

run for t = 200 steps

→ **six stationary** patterns (attractors of size 1)

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

The Attractors



*EMF*1, *TFL*1, *LFY*, *AP*1, *CAL*, *LUG*, *UFO*, *BFU*,*AG*, *AP*3, *PI* and *SUP*.

A C B

From gene **activation patterns** in the attractors:

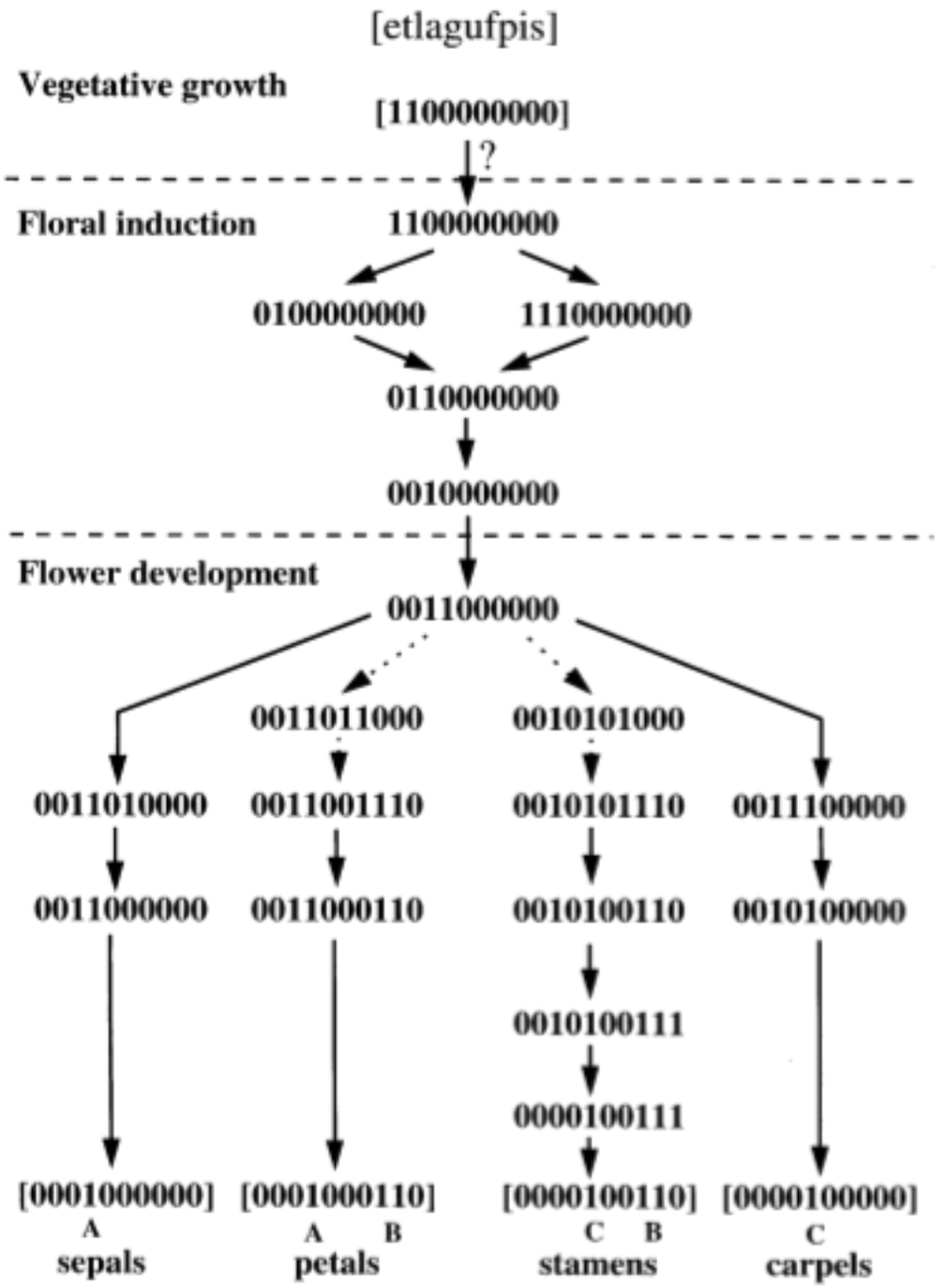
→ identify the **four floral** tissue **types** of the ABC model

→ one attractor with floral **inhibitors** *EMF*1, *TFL*1
(characteristic for cells that are not part of the flowers)

→ one yet **unidentified** state

Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

Possible Pathways



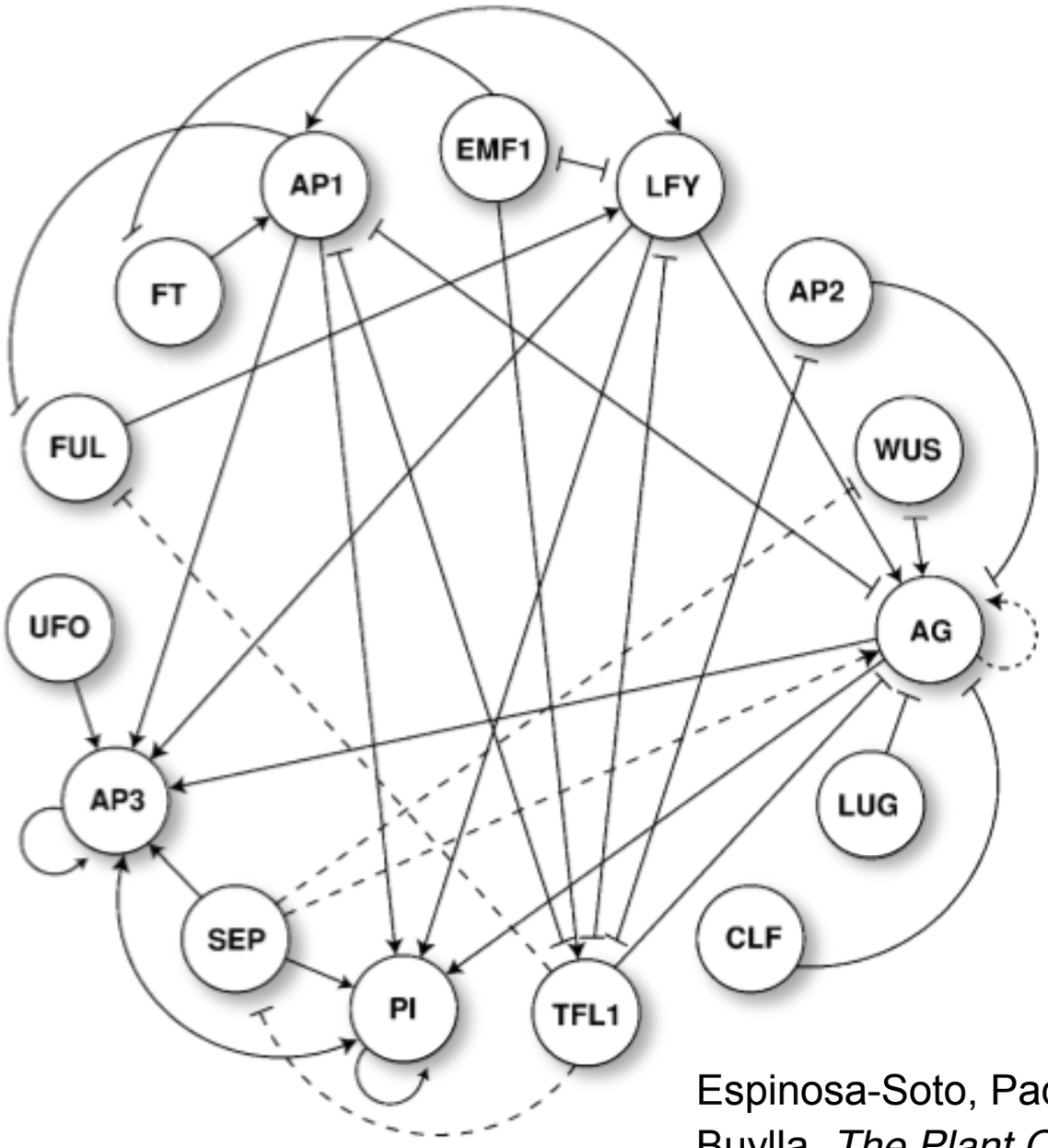
Note: the model does not include temporal and spatial information required to predict where and when which genes are activated or repressed ("signals")

→ these pathways are a "proposal" only

Mendoza et al, *Bioinformatics* **15** (1999) 593

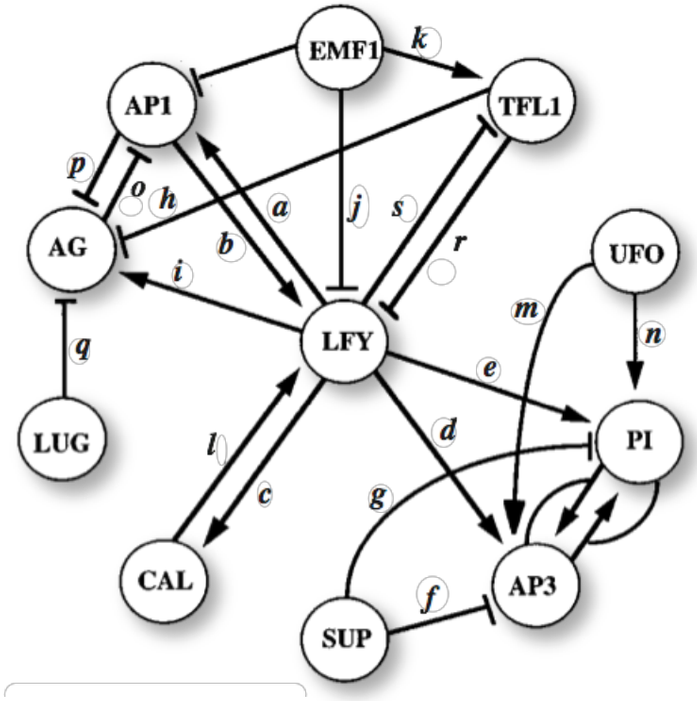
Sophistication of Networks

A few years later: additional genes and predicted interactions (- - -)



2004

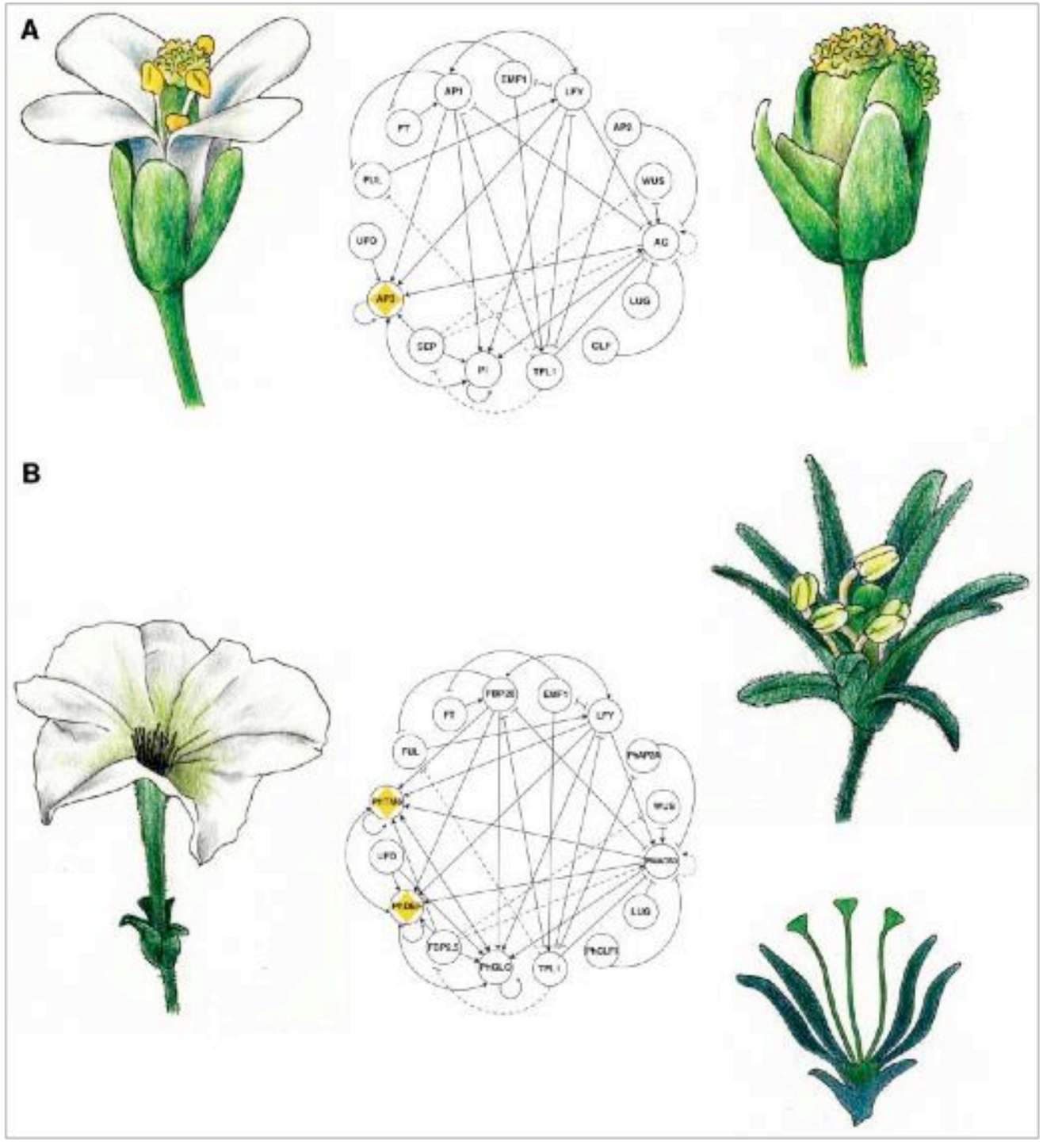
1998



Mendoza, Alvarez-Buylla, *J. theor Biol.* **193** (1998) 307

Espinosa-Soto, Padilla-Longoria, Alvarez-Buylla, *The Plant Cell* **16** (2004) 2923

Predictions for Petunia



From *A. thaliana* predict/understand *green petals* mutant phenotype for petunia.

Espinosa-Soto, Padilla-Longoria, Alvarez-Buylla, *The Plant Cell* **16** (2004) 2923

What is it Worth?

Generally: → quality of the **results** depends on the quality of the **model**

→ quality of the model depends on the quality of the **assumptions**

Assumptions for the Boolean network description:

(• subset of the species considered → reduced system state space)

- only discrete density levels
- conditional yes–no causality

→ dynamic balances lost,
reduced to oscillations
→ no continuous processes

- discretized propagation steps

→ timing of concurrent paths?

"You get what you pay for"