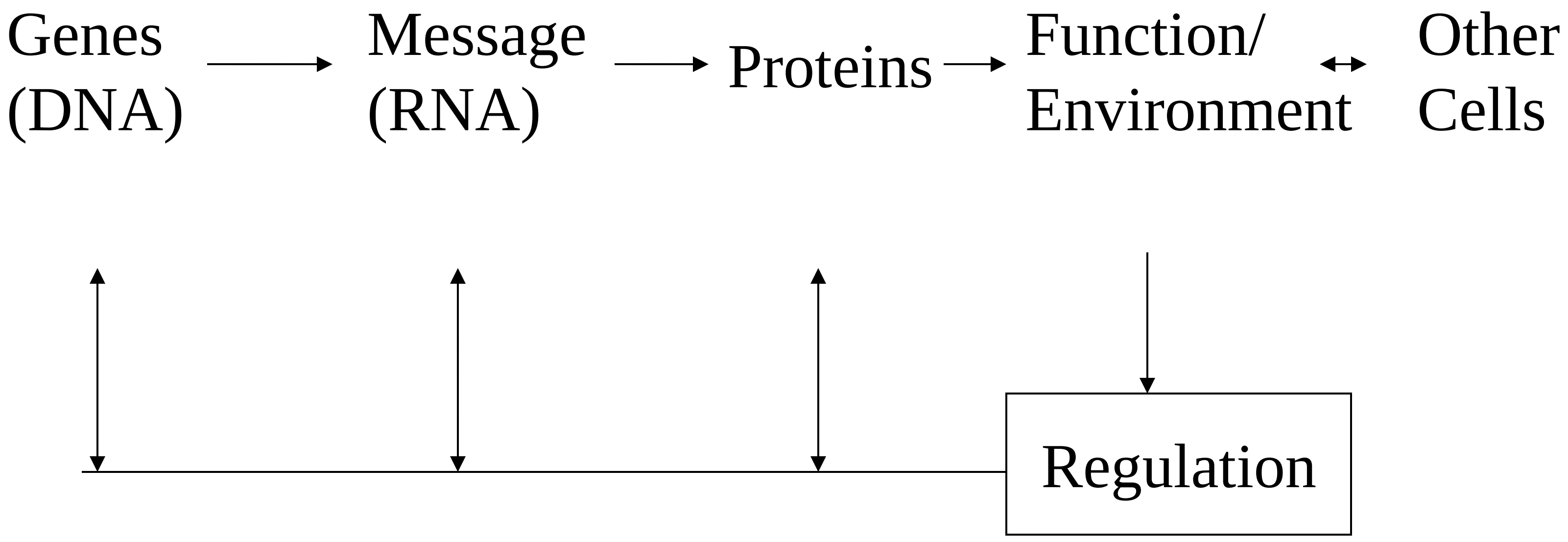
GLOBEX Bioinformatics (Summer 2015)

Genetic networks and gene expression data

Gene Networks

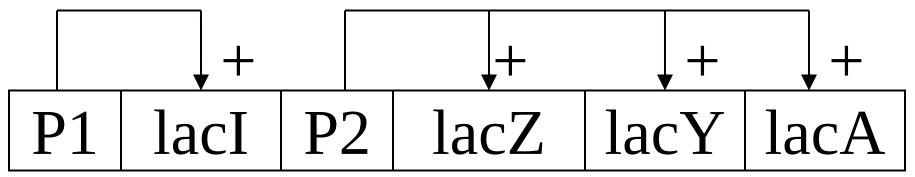
- **Definition**: A gene network is a set of molecular components, such as genes and proteins, and interactions between them that collectively carry out some cellular function. A genetic regulatory network refers to the network of controls that turn on/off gene transcription.
- **Motivation**: Using a known structure of such networks, it is sometimes possible to describe behavior of cellular processes, reveal their function and the role of specific genes and proteins
- **Experiments**
  - DNA microarray : observe the expression of many genes simultaneously and monitor gene expression at the level of mRNA abundance.
  - Protein chips: the rapid identification of proteins and their abundance is becoming possible through methods such as 2D polyacrylamide gel electrophoresis.
  - 2-hybrid systems: identify protein-protein interactions
    - (Stan Fields' lab http://depts.washington.edu/sfields/)

Regulation

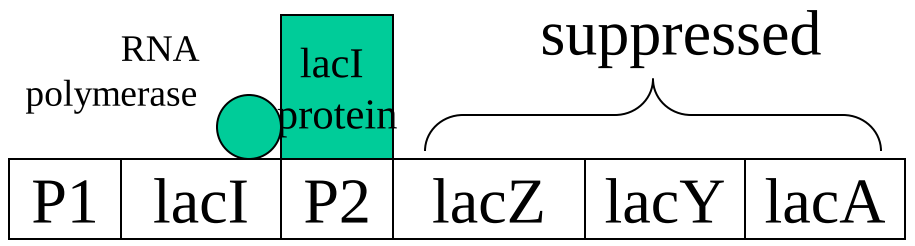


Genes (DNA)
Message (RNA)
Proteins
Function/ Environment
Other Cells
Regulation

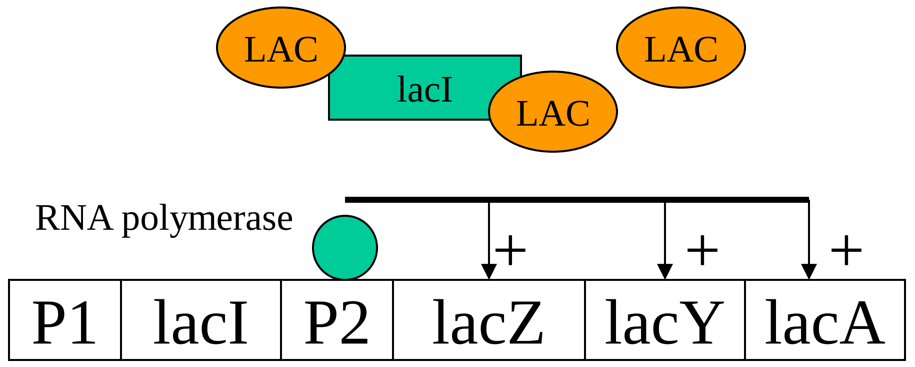
Operon



lac operon on *E. coli*



Repressor protein coded by lacI, bind to P2 preventing transcription of lacZ, lacY and lacA



Lactose binds with lacI, allowing RNA polymerase to bind to P2 and transcribe the structural genes

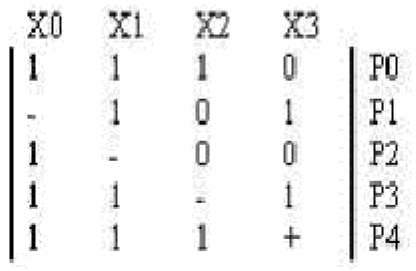
Genetic Network Models

- **Linear Model**: expression level of a node in a network depends on linear combination of the expression levels of its neighbors.
- **Boolean Model**: The most promising technique to date is based on the view of gene systems as a logical network of nodes that influence each other's expression levels. It assumes only two distinct levels of expression: 0 and 1. According to this model a value of a node at the next step is boolean function of the values of its neighbors.
- **Bayesian Model**: attempts to give a more accurate model of network behavior, based on Bayesian probabilities for expression levels.

Evaluation of Models

- Inferential power
- Predictive power
- Robustness
- Consistency
- Stability
- Computational cost

Boolean Networks: An example



	X0	X1	X2	X3
P0	1	1	1	0
P1	-	1	0	1
P2	1	-	0	0
P3	1	1	-	1
P4	1	1	1	+

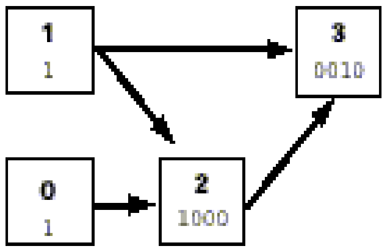
1: induced

0: suppressed

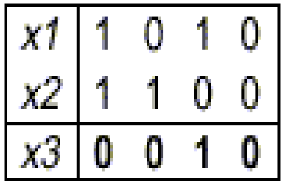
-: forced low

+: forced high

Interpreting data ↑ ↓ Reverse Engineering

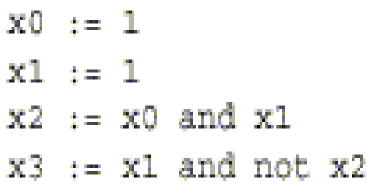


A A directed graph structure with numbered nodes connected by edges



x1	1	0	1	0
x2	1	1	0	0
x3	0	0	1	0

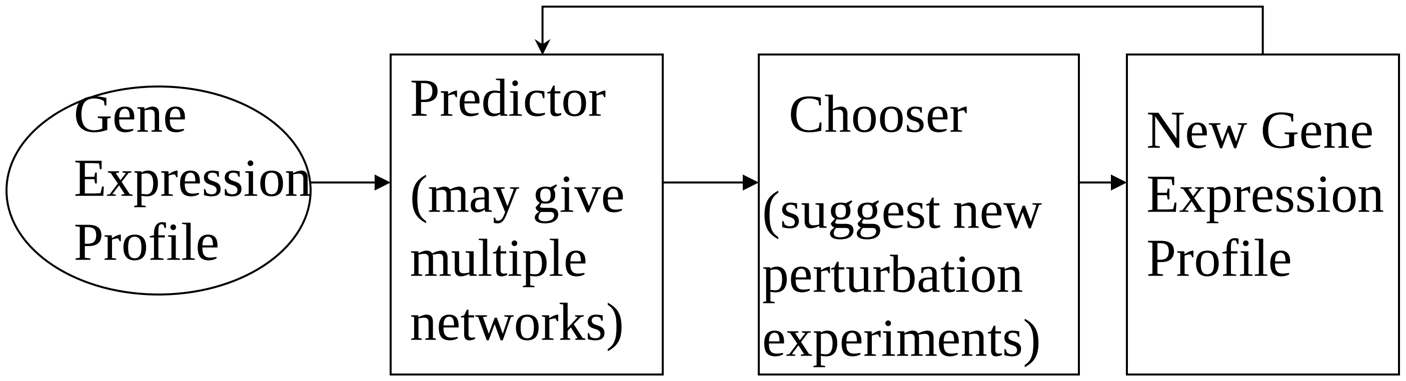
B The truth table (shown for node 3 only)



```
x0 := 1
x1 := 1
x2 := x0 and x1
x3 := x1 and not x2
```

C The logic equations for each node

Boolean networks: A Predictor/Chooser scheme


Gene Expression Profile
Predictor
(may give multiple networks)
Chooser
(suggest new perturbation experiments)
New Gene Expression Profile

Predictor

- A population of cells containing a target genetic network *T* is monitored in the steady state over a series of *M* experimental perturbations.
- In each perturbation p_m $(0 \le m < M)$ any number of nodes may be forced to a low or high level.

$$
\boldsymbol{E} = \begin{array}{c|cccc|l}
 & x_0 & x_1 & x_2 & x_3 & \\
 & 1 & 1 & 1 & 0 & p_0 \quad \longleftarrow \text{ Wild-type state} \\
 & - & 1 & 0 & 1 & p_1 \\
 & 1 & - & 0 & 0 & p_2 \\
 & 1 & 1 & - & 1 & p_3 \\
 & 1 & 1 & 1 & + & p_4
\end{array}
$$

-: forced low

+: forced high

Figure 2: Example expression matrix generated from the genetic network in fig. 1.

Step 1. For each gene x_n, find all pairs of rows (i, j) in $\boldsymbol{E}$ in which the expression level of x_n differs, excluding rows in which x_n was forced to a high or low value.

$$\boldsymbol{E} = \begin{array}{c|cccc|c} & x_0 & x_1 & x_2 & x_3 & \\ \hline & 1 & 1 & 1 & 0 & p_0 \\ & - & 1 & 0 & 1 & p_1 \\ & 1 & - & 0 & 0 & p_2 \\ & 1 & 1 & - & 1 & p_3 \\ & 1 & 1 & 1 & + & p_4 \end{array}$$

Figure 2: Example expression matrix generated from the genetic network in fig. 1.

For x_3, we find:

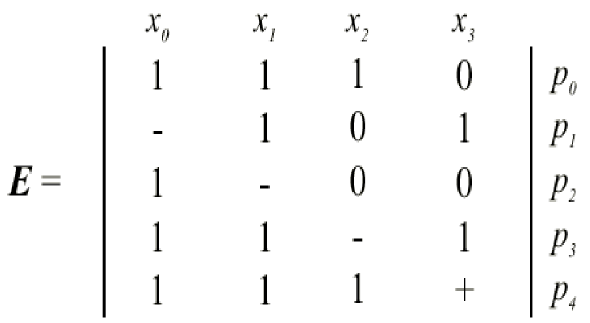
(p0, p1),

(p0, p3),

(p1, p2),

(p2, p3)

Step 2. For each pair (i,j), S_{ij} contains all other genes whose expression levels also differ between experiments i and j. Find the *minimum cover set* S_{min}, which contains at least one node from each set S_{ij}



$$E = \begin{array}{c|cccc|c} & x_0 & x_1 & x_2 & x_3 & \\ \hline & 1 & 1 & 1 & 0 & p_0 \\ & - & 1 & 0 & 1 & p_1 \\ & 1 & - & 0 & 0 & p_2 \\ & 1 & 1 & - & 1 & p_3 \\ & 1 & 1 & 1 & + & p_4 \end{array}$$

Figure 2: Example expression matrix generated from the genetic network in fig. 1.

Step 1:

(p0,p1),

(p0, p3),

(p1,p2),

(p2,p3)

Step 2:

(p0, p1)->$S_{01}=\{x_0, x_2\}$

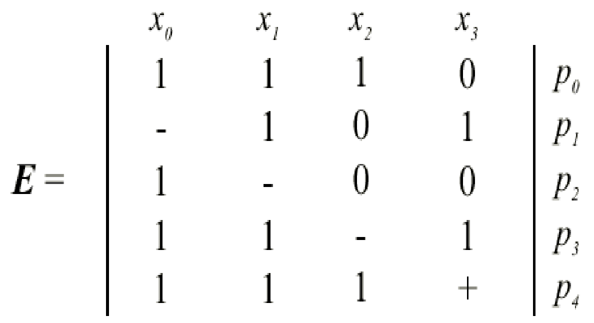
(p0, p3)->$S_{03}=\{x_2\}$

(p1, p2)-> $S_{12}=\{x_0, x_1\}$

(p2, p3)->$S_{23}=\{x_1)$

So, now the S_{min} is $\{x_1, x_2\}$

Step 3. use the nodes in S_{min} as input, x_n as output, build truth table to find out f_n (In this example, n=3)



$$E = \begin{array}{c|cccc|c} & x_0 & x_1 & x_2 & x_3 & \\ \hline & 1 & 1 & 1 & 0 & p_0 \\ & - & 1 & 0 & 1 & p_1 \\ & 1 & - & 0 & 0 & p_2 \\ & 1 & 1 & - & 1 & p_3 \\ & 1 & 1 & 1 & + & p_4 \end{array}$$

Figure 2: Example expression matrix generated from the genetic network in fig. 1.

Now the S_{min} is $\{x_1, x_2\}$

x_1	1 0 1 0
x_2	1 1 0 0
x_3	0 * 1 0

So $f_3 = 0 * 1 0$

* cannot be determined

Chooser

For L hypothetical equiprobable networks generated by the predictor, choose perturbation p that would best discriminate between the L networks, by maximizing entropy H_p as defined below.

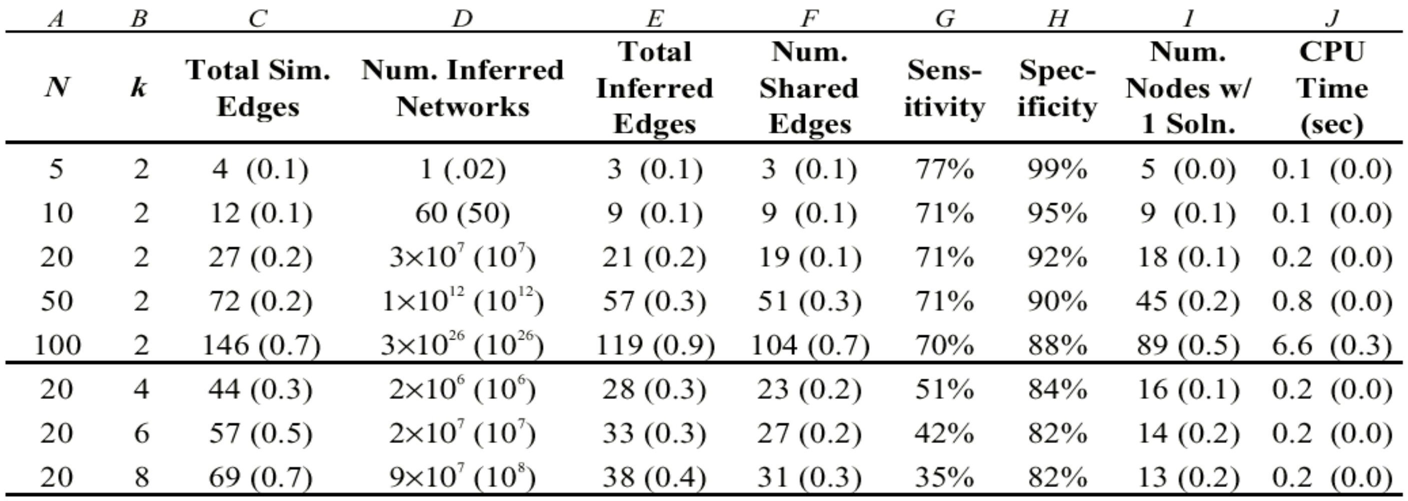
$$H_p = - \sum_{s=1}^{S} (l_s/L) \log_2 (l_s/L)$$

where l_s is the number of networks giving the state s

Note: $(1 \leq s \leq S)$, and $(1 \leq S \leq L)$

Result and Evaluation

- Evaluation of Predictor
- construct a target network *T:* size = *N*, and maximum in-degree = *k* (where the in-degree of a node is its number of incoming edges)
- *sensitivity* is defined as the percentage of edges in the target network that were also present in the inferred network, and *specificity* is defined as the percentage of edges in the inferred network that were also present in the target network.



A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
N	***k***	**Total Sim. Edges**	**Num. Inferred Networks**	**Total Inferred Edges**	**Num. Shared Edges**	**Sens-itivity**	**Spec-ificity**	**Num. Nodes w/ 1 Soln.**	**CPU Time (sec)**
5	2	4 (0.1)	1 (.02)	3 (0.1)	3 (0.1)	77%	99%	5 (0.0)	0.1 (0.0)
10	2	12 (0.1)	60 (50)	9 (0.1)	9 (0.1)	71%	95%	9 (0.1)	0.1 (0.0)
20	2	27 (0.2)	3×10^{7} (10^{7})	21 (0.2)	19 (0.1)	71%	92%	18 (0.1)	0.2 (0.0)
50	2	72 (0.2)	1×10^{12} (10^{12})	57 (0.3)	51 (0.3)	71%	90%	45 (0.2)	0.8 (0.0)
100	2	146 (0.7)	3×10^{26} (10^{26})	119 (0.9)	104 (0.7)	70%	88%	89 (0.5)	6.6 (0.3)
20	4	44 (0.3)	2×10^{6} (10^{6})	28 (0.3)	23 (0.2)	51%	84%	16 (0.1)	0.2 (0.0)
20	6	57 (0.5)	2×10^{7} (10^{7})	33 (0.3)	27 (0.2)	42%	82%	14 (0.2)	0.2 (0.0)
20	8	69 (0.7)	9×10^{7} (10^{8})	38 (0.4)	31 (0.3)	35%	82%	13 (0.2)	0.2 (0.0)

Discussions

- Incorporate pre-existing information
- Boolean to multi-levels
- Cyclic networks
- Noise tolerance

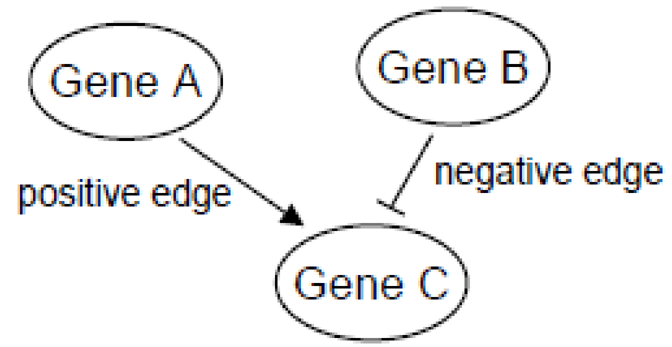
References

- Ideker, Thorsson, and Karp, PSB 2000, 5: 302-313.

Bayesian Networks

Biological processes are stochastic

- Data can be noisy as well.



§ Quantitative part:

Gene A	Gene B	P(C+|AB)	P(C-|AB)
+	+	0.6	0.4
-	+	0.01	0.99
+	-	0.99	0.01
-	-	0.4	0.6

This row indicates that when Gene A and Gene B are up-regulated, then Gene C has a 60% probability to be up-regulated and a 40% probability to be down-regulated.

$$\begin{array}{cc} & \text{conditions} \\ \begin{array}{c} \uparrow \\ \text{variables} \\ X_1,\ldots X_n \\ \text{(genes)} \\ \downarrow \end{array} & \underbrace{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & & x_{2m} \\ \vdots & & \ddots & \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}}_{\text{m independent (steady-state) observatio}} \end{array} \Longrightarrow$$

of the system $X_1,\ldots,X_N$

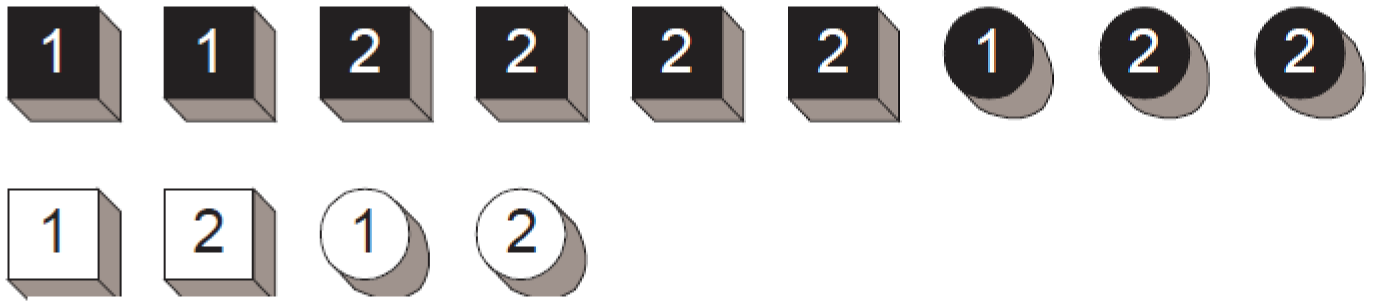
Join probability
$P(X_1, \ldots, X_n)$

e.g.,
$P(+,+,-, \ldots,+) = 0.003$
$P(-,+,+,\ldots, -) = 0.00015$
…
2^N

How many combinations?

Query/Inference: P(X1 | X6, X7) ?

Conditional Probability and Conditional Independence



$$\begin{aligned} P(\text{One}) &= \frac{5}{13} \\ P(\text{One}|\text{Square}) &= \frac{3}{8} \end{aligned}$$

$$\begin{aligned} P(\text{One}|\text{Black}) &= \frac{3}{9} = \frac{1}{3} \\ P(\text{One}|\text{Square} \cap \text{Black}) &= \frac{2}{6} = \frac{1}{3} \end{aligned}$$

$$\begin{aligned} P(\text{One}|\text{White}) &= \frac{2}{4} = \frac{1}{2} \\ P(\text{One}|\text{Square} \cap \text{White}) &= \frac{1}{2}. \end{aligned}$$

So One *and* Square *are not independent, but they are conditionally independent given* Black *and given* White.

Bayesian Network as an efficient way to factorize the Joint Probability

Factorization of join probability

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$$

# of parameters = $2^N - 1$

Example:

$$P(\mathrm{A,E,B,C,D}) = P(\mathrm{A})P(\mathrm{E|A})P(\mathrm{B|A,E})P(\mathrm{C|A,E,B}) \\ P(\mathrm{D|A,E,B,C})$$

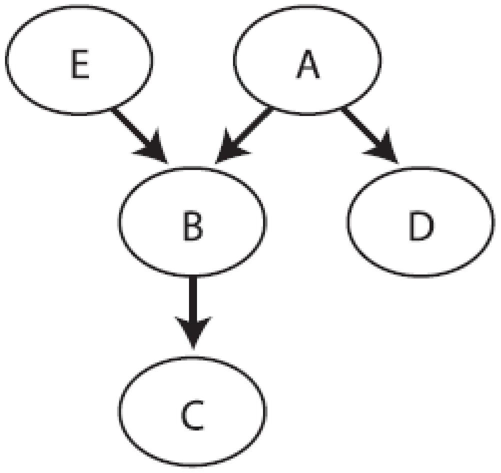
# of parameters = 1 + 2 + 4 + 8 + 16 = 31

Conditional independence

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa_i)$$

$$P(\mathrm{A,E,B,C,D}) = P(\mathrm{A})P(\mathrm{E})P(\mathrm{B|A,E})P(\mathrm{C|B})P(\mathrm{D|A})$$

Assuming max in-degree k, the number of parameters is reduced to $2^k N$



# of parameters = 1 + 1 + 4 + 1 + 1 = 10

A greedy Algorithm to Learn Bayesian Network from the data

Input

D // a data set
G_o // initial network structure

Output

G // final network structure

Greedy-structure-search

```
$G_{best} = G_o$
repeat // apply best possible operator to G in each iteration
    $G = G_{best}$
    foreach operator o // (each edge addition, deletion, or reversal on $G$)
        $G^o = o(G)$ // apply to $G$
        if $G^o$ is cyclic continue
        if scoreBDe($G^o : D$) > scoreBDe($G_{best} : D$)
            $G_{best} = G^o$

until $G == G_{best}$ // no change in structure improves score
```

Parameter Estimation

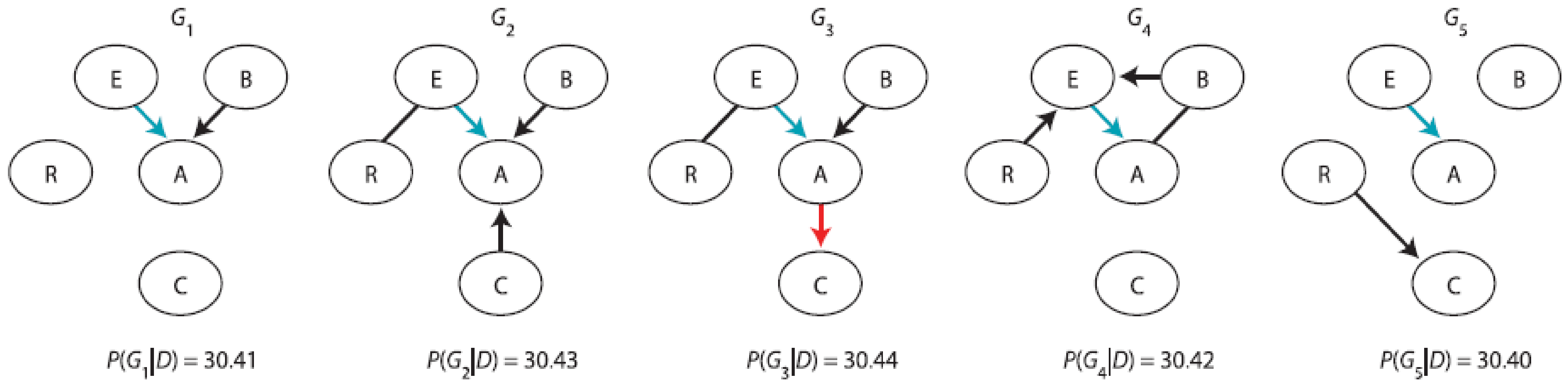
-Maximum Likelihood
-Bayesian approach
- Dirichlet priors are used for model parameters.

Structure evaluation

$$\begin{aligned}\text{BayesianScore}(M) &= \log[P(M \mid D)] \\ &= \log[P(M)] + \log[P(D \mid M)] + c\end{aligned}$$

- Where M = model, D = microarray data, c = constant

Model Averaging



$$P[f(G)|D] = \sum_G f(G)P(G|D)$$

Feature f: edge X→Y is in the network.

f(G) = 1, if G has the feature
= 0, otherwise.

How to compute P[f(G)|D]?
-Enumerate all high scored networks
- Sampling (MCMC)
- Bootstrap

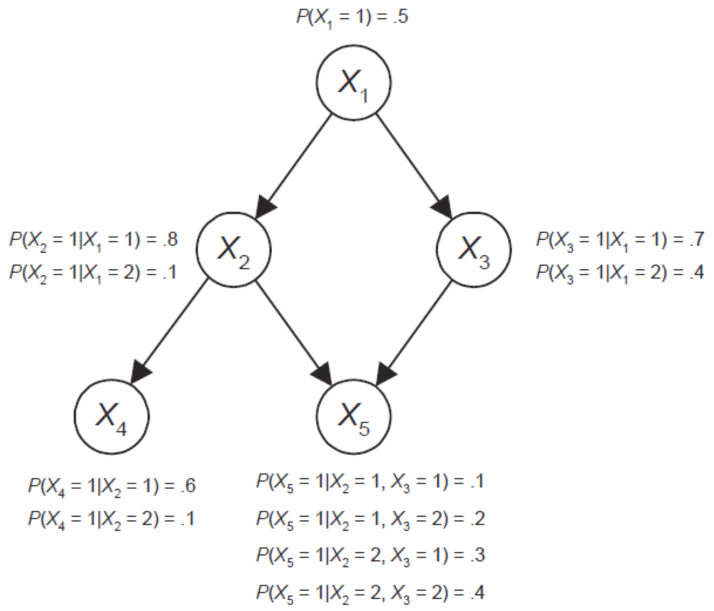
Bootstrap

- For $i = 1, \ldots, m$, construct a data set D_i by sampling, with replacement, M instances from D. Then, apply the learning procedure on D_i to induce a network structure G_i.
- For each feature f of interest, calculate

$$\mathrm{conf}(f) = \frac{1}{m} \sum_{i=1}^{m} f(G_i)$$

Inference

- Given $P(X_1, \ldots, X_N)$ as a BN, calculate $P(X_i \mid \text{evidence})$, where evidence is a subset of nodes that we know their values.



$P(X_1 = 1) = .5$
X_1
$P(X_2 = 1|X_1 = 1) = .8$
$P(X_2 = 1|X_1 = 2) = .1$
X_2
X_3
$P(X_3 = 1|X_1 = 1) = .7$
$P(X_3 = 1|X_1 = 2) = .4$
X_4
X_5
$P(X_4 = 1|X_2 = 1) = .6$
$P(X_4 = 1|X_2 = 2) = .1$
$P(X_5 = 1|X_2 = 1, X_3 = 1) = .1$
$P(X_5 = 1|X_2 = 1, X_3 = 2) = .2$
$P(X_5 = 1|X_2 = 2, X_3 = 1) = .3$
$P(X_5 = 1|X_2 = 2, X_3 = 2) = .4$

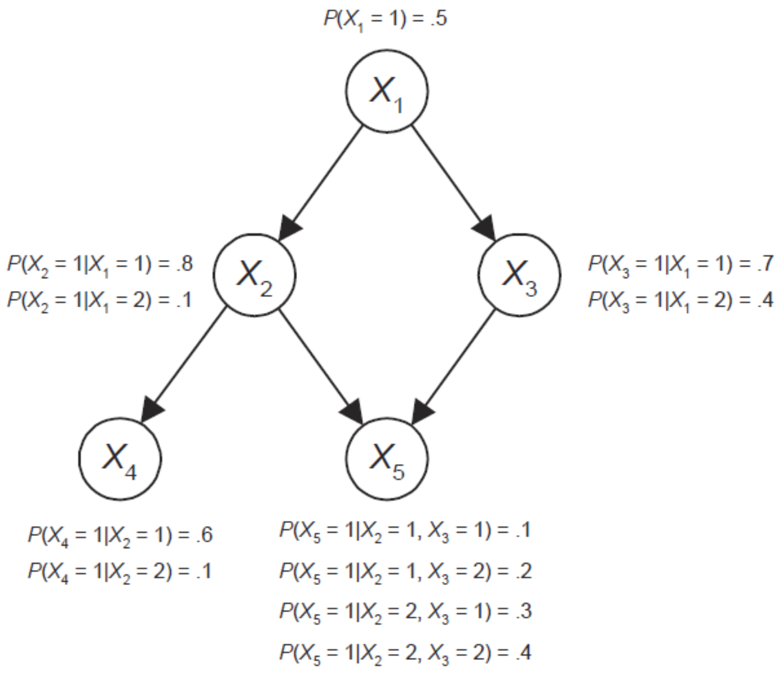
e.g., $P(X_2 \mid X_3, X_4) =?$

- **Exact inference** is NP-hard (Cooper, 1990)

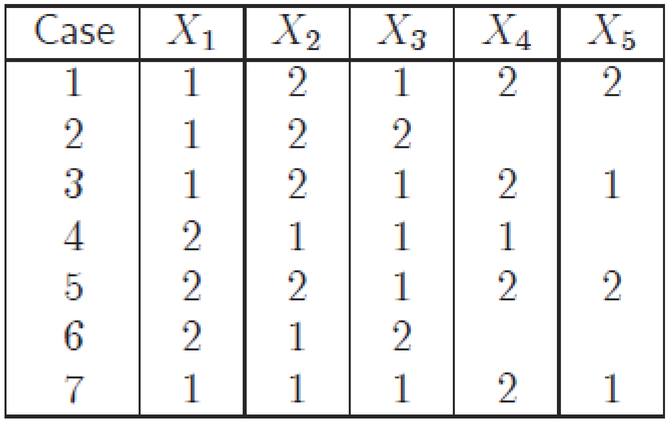
$$P(X_i \mid \text{evidence}) = \sum_{Y \in V - \{Xi, \text{evidence}\}} P(X_i \mid Y)$$

Inference by Sampling

- Direct sampling, e.g. $P(X_5)$
- Rejection sampling
- Weighted (likelihood) sampling
- Gibbs sampling



$P(X_1 = 1) = .5$
X_1
$P(X_2 = 1|X_1 = 1) = .8$
$P(X_2 = 1|X_1 = 2) = .1$
X_2
X_3
$P(X_3 = 1|X_1 = 1) = .7$
$P(X_3 = 1|X_1 = 2) = .4$
X_4
X_5
$P(X_4 = 1|X_2 = 1) = .6$
$P(X_4 = 1|X_2 = 2) = .1$
$P(X_5 = 1|X_2 = 1, X_3 = 1) = .1$
$P(X_5 = 1|X_2 = 1, X_3 = 2) = .2$
$P(X_5 = 1|X_2 = 2, X_3 = 1) = .3$
$P(X_5 = 1|X_2 = 2, X_3 = 2) = .4$



Case	X_1	X_2	X_3	X_4	X_5
1	1	2	1	2	2
2	1	2	2		
3	1	2	1	2	1
4	2	1	1	1	
5	2	2	1	2	2
6	2	1	2		
7	1	1	1	2	1

$P(X_1 =1| X_3=1, X_4=2) = ¾$
$P(X_2 =1| X_3, X_3) = ¼$
$P(X_5 =1| X_3, X_3) = 2/4$

Learning Bayesian Networks, Neapolitan, 2004

Likelihood Weighting

$Case$	X_1	X_2	X_3	X_4	X_5	$score'$
1	2	2	1	2	1	.36
2	1	1	1	2	2	.28
3	2	1	1	2	2	.16
4	1	1	1	2	1	.28

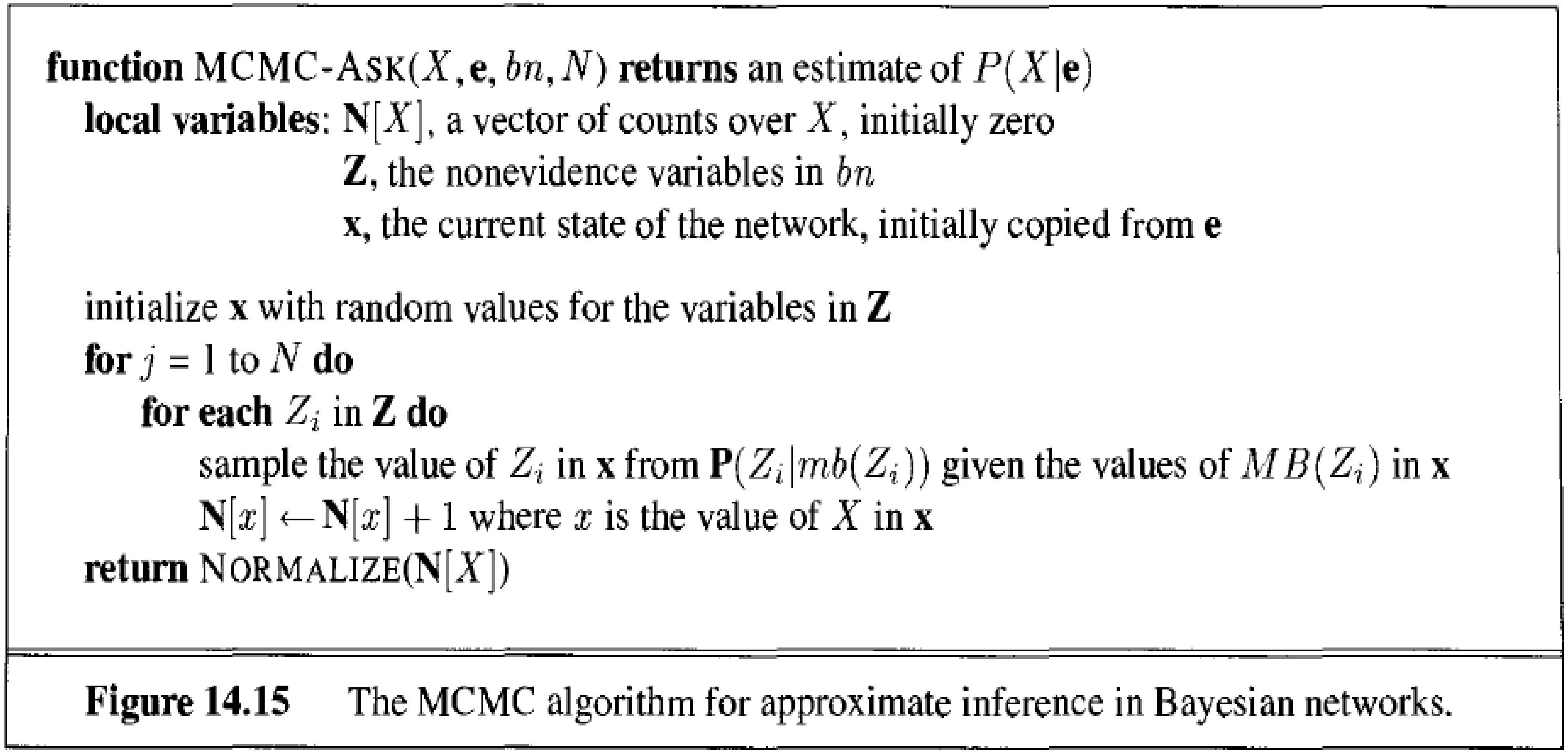
$$\begin{aligned}\hat{P}(X_1 = 1 | X_3 = 1, X_4 = 2) &\propto [score'(Case\ 2) + score'(Case\ 4)] \\ &\propto [.28 + .28] = .56\end{aligned}$$

$$\begin{aligned}\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) &\propto [score'(Case\ 1) + score'(Case\ 3)] \\ &\propto [.36 + .16] = .52.\end{aligned}$$

So

$$\hat{P}(X_1 = 1 | X_3 = 1, X_4 = 2) = \frac{.56}{.56 + .52} = .52.$$

Gibbs sampler



```
function MCMC-ASK($X$, e, $bn$, $N$) returns an estimate of $P(X|\mathbf{e})$
    local variables: N[$X$], a vector of counts over $X$, initially zero
                     Z, the nonevidence variables in $bn$
                     x, the current state of the network, initially copied from e

    initialize x with random values for the variables in Z
    for $j$ = 1 to $N$ do
        for each $Z_i$ in Z do
            sample the value of $Z_i$ in x from $\mathbf{P}(Z_i|mb(Z_i))$ given the values of $MB(Z_i)$ in x
            $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where $x$ is the value of $X$ in x
    return NORMALIZE(N[$X$])
```

Figure 14.15 The MCMC algorithm for approximate inference in Bayesian networks.

Russell & Norvig, AI Modern Approach, 2ed.

Markov blanket

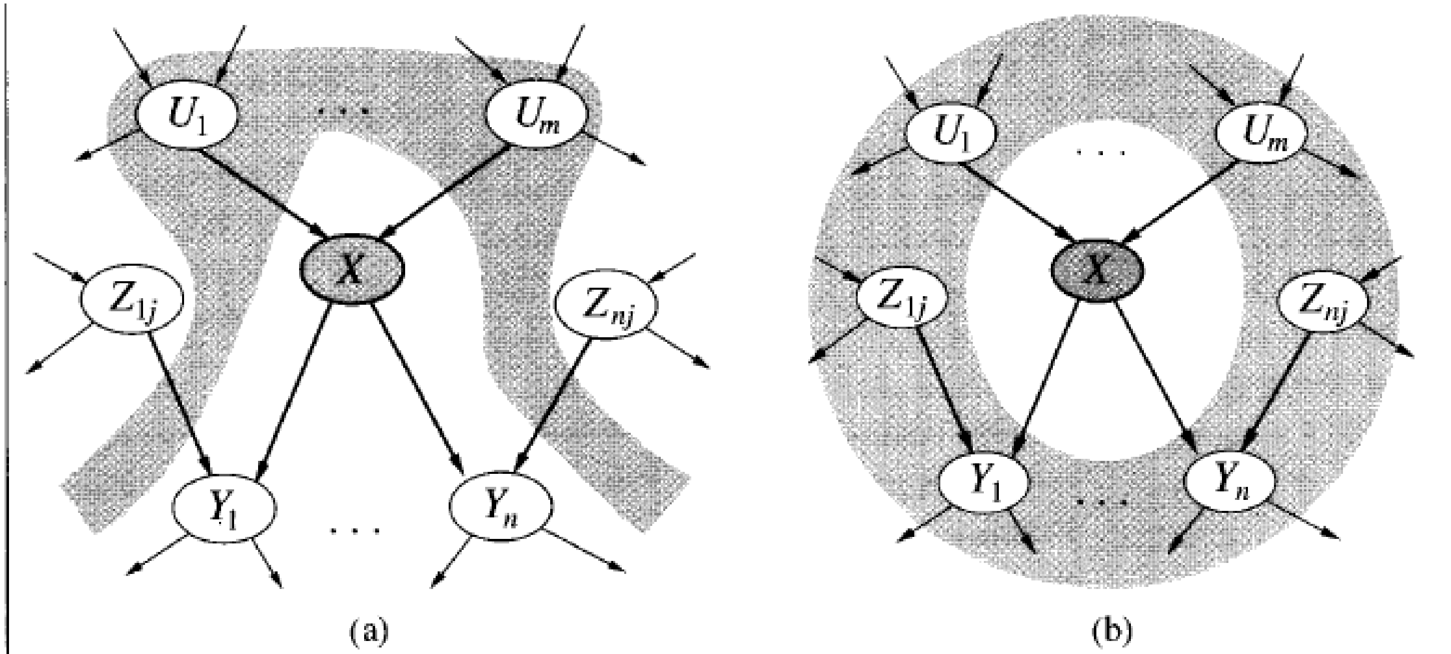


Figure 14.4 (a) A node X is conditionally independent of its non-descendants (e.g., the Z_{ij}s) given its parents (the U_is shown in the gray area). (b) A node X is conditionally independent of all other nodes in the network given its Markov blanket (the gray area).

Russell & Norvig, AI Modern Approach, 2ed.

Model Intervention for causality

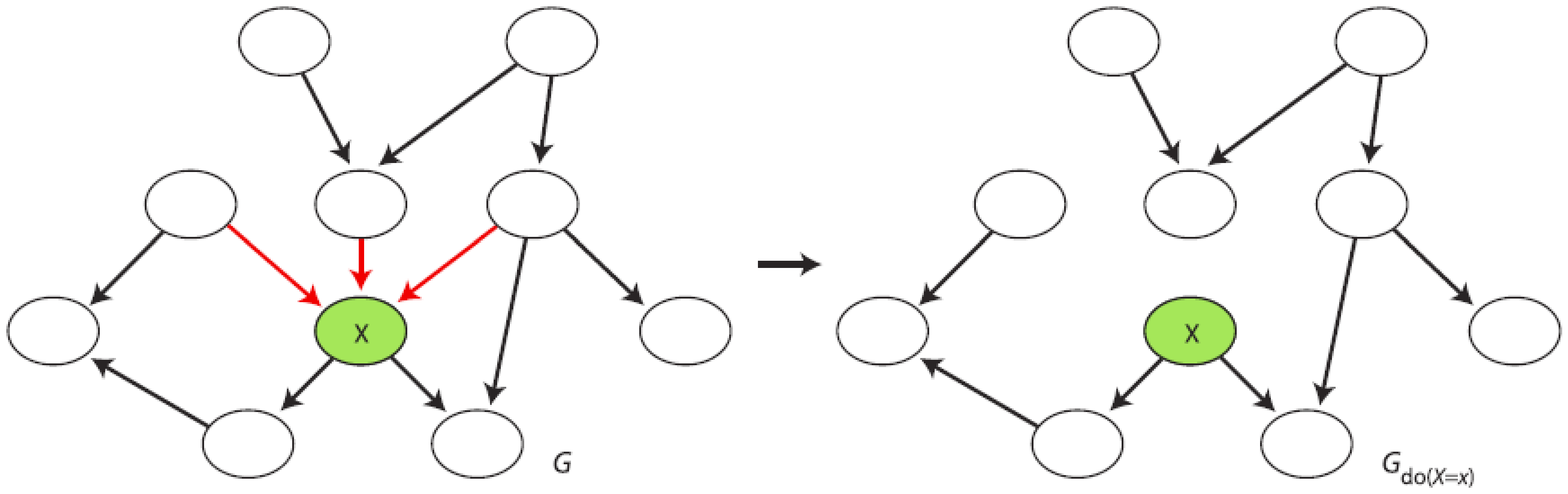
- Bayesian networks <= causal networks

- Bayesian networks + intervention => causality

  External interventions are needed to infer the causal direction for edges in a Bayesian network:
  - Genetic mutations
  - siRNA
  - small chemical interventions as inhibitors or activators

Intervention on X

- do(X=x) for node X to take value x.
- Incoming edges to X are cut off.
- X becomes a root node, and P(X=x) =1.



Example

Two Bayesian networks: $X \rightarrow Y$ and $Y \rightarrow X$

- Equivalent based on observed data
- do(X=x) for node X to take value x.
- If $X \rightarrow Y$ is the causal network, then

  $$P(Y|do(X=x)) = P(Y|X=x)$$

- If $Y \rightarrow X$ is the causal network, then

  $$P(Y|do(X=x)) = P(Y)$$

We will get a different conditional distribution in the observational and inhibited (intervened) samples.
Whereas we cannot distinguish between the two models with the use of observations alone, we can differentiate between them with the use of interventional data.

Dynamics of gene expression regulation

Linear model:

$$\frac{dE_i^t}{dt} = \sum_j w_{ij} E_j^t + b_i$$

Non-linear model:

$$\frac{d^2 E_i(t)}{dt^2} + 2\lambda_i \omega_i \frac{dE_i(t)}{dt} + \omega_i^2 E_i(t) = \sum_j w_{ij} E_j(t)$$

- $w_{ij} > 0$: gene j activates gene i
- $w_{ij} < 0$: gene j inhibits gene i
- $w_{ij} = 0$: gene j does not regulate gene i

Discretization:

$$\frac{\Delta E_i(t)}{\Delta t} = E_i(t+1) - E_i(t).$$

$$X_t = \left(E_1(t), \cdots, E_n(t), \frac{\Delta E_1(t)}{\Delta t}, \cdots, \frac{\Delta E_n(t)}{\Delta t} \right)'$$

$$X_{t+1} = AX_t$$

$$A = \left[\begin{array}{cc} \text{identity} & \text{identity} \\ W - \Omega^2 & \text{identity} - 2\Omega\Lambda \end{array} \right]$$

Stochastic -- adding noise

$$\left\{ \begin{array}{lcl} X_{t+1} & = & AX_t + \mathbf{u} \\ Y_t & = & CX_t + \mu_{obs} + \mathbf{v} \end{array} \right.$$

Dynamic Bayesian networks

A *dynamic Bayesian network* N is a representation of stochastic evolution of a set of random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ over discretized time. Represented temporal process is assumed to be *Markovian*, i.e.

$$P(\mathbf{X}(t)|\mathbf{X}(0), \mathbf{X}(1), \ldots, \mathbf{X}(t-1)) = P(\mathbf{X}(t)|\mathbf{X}(t-1))$$

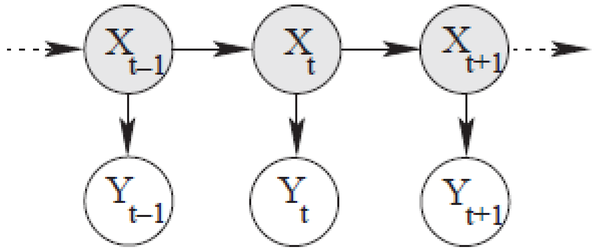
and *time homogenous*, i.e.$P(\mathbf{X}(t)|\mathbf{X}(t-1))$ are independent of t. The representation consists of two components:

- a directed graph $G = (\mathbf{X}, \mathbf{E})$ encoding con ditional (in-)dependencies

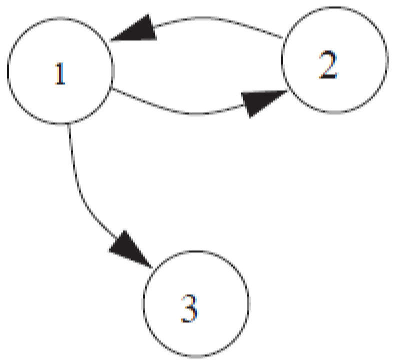
- a family of conditional distributions $P(X_i(t)|\mathbf{Pa}_i(t-1))$, where $\mathbf{Pa}_i = \{X_j \in \mathbf{X}|(X_j, X_i) \in \mathbf{E}\}$

$$P(\mathbf{X}(0), \mathbf{X}(1), \ldots, \mathbf{X}(T)) = P(\mathbf{X}(0)) \prod_{t=1}^{T} P(\mathbf{X}(t) \mid \mathbf{X}(t-1))$$

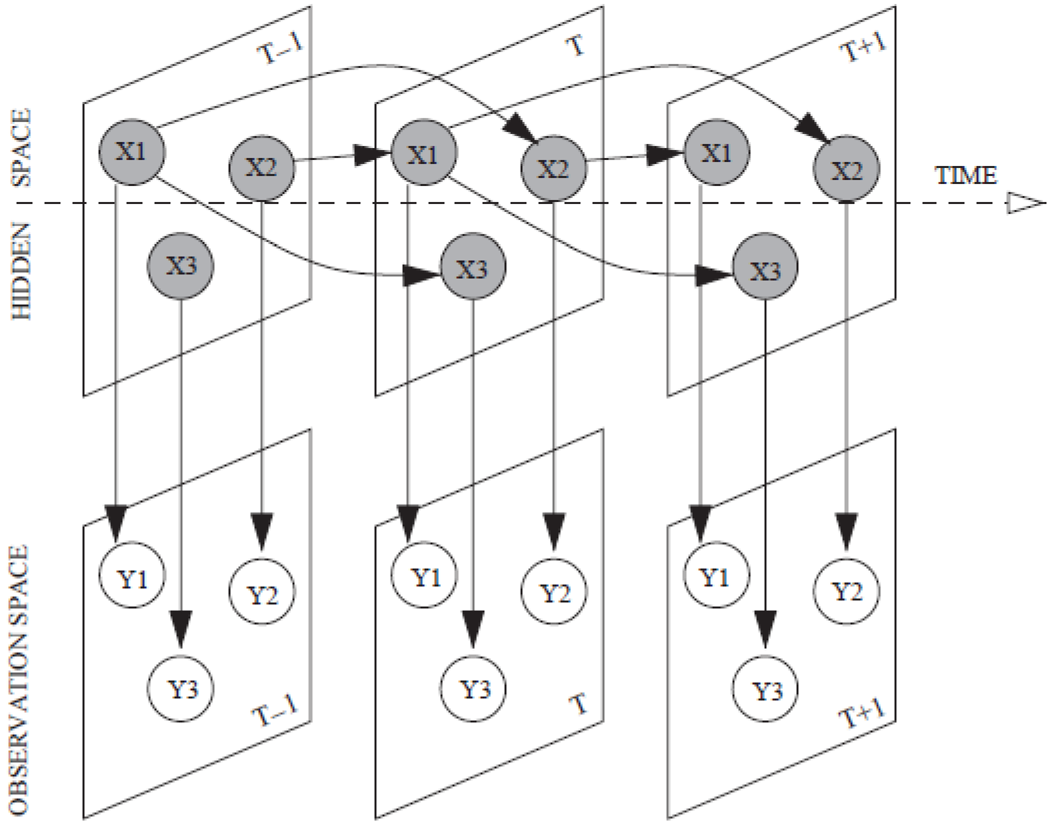
$$\begin{aligned} P(\mathrm{X}(1), \ldots, \mathrm{X}(T) \mid \mathrm{X}(0)) = \prod_{t=1}^{T} P(\mathrm{X}(t) \mid \mathrm{X}(t-1)) &= \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_i(t) \mid \mathrm{Pa}_i(t-1)) = \\ &= \prod_{i=1}^{n} \prod_{t=1}^{T} P(X_i(t) \mid \mathrm{Pa}_i(t-1)) \end{aligned}$$


X_{t-1}
X_t
X_{t+1}
Y_{t-1}
Y_t
Y_{t+1}

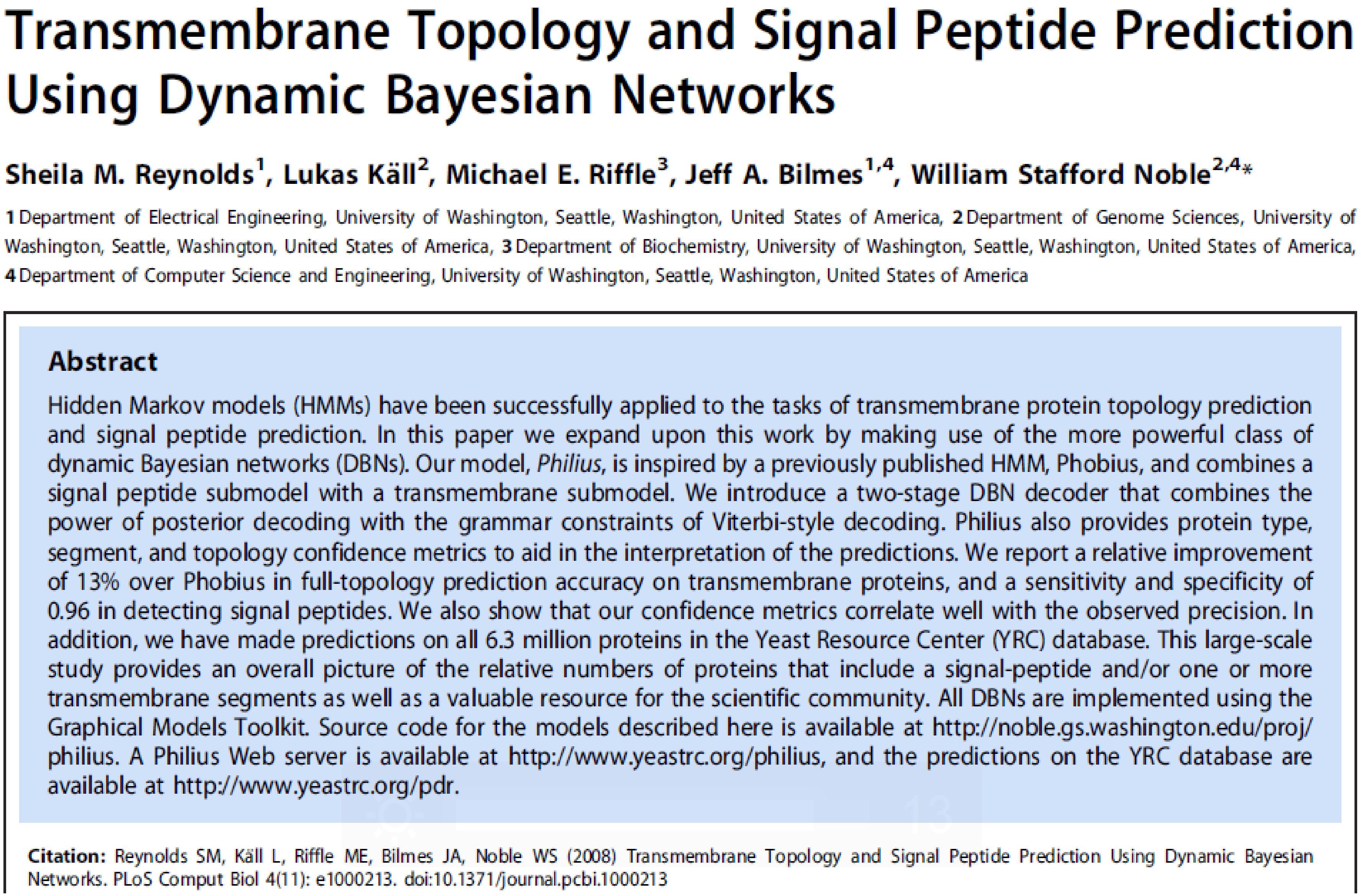
“Loops” in DBN



1
2
3



HIDDEN SPACE
OBSERVATION SPACE
TIME
T–1
T
T+1
X1
X2
X3
Y1
Y2
Y3



Transmembrane Topology and Signal Peptide Prediction Using Dynamic Bayesian Networks

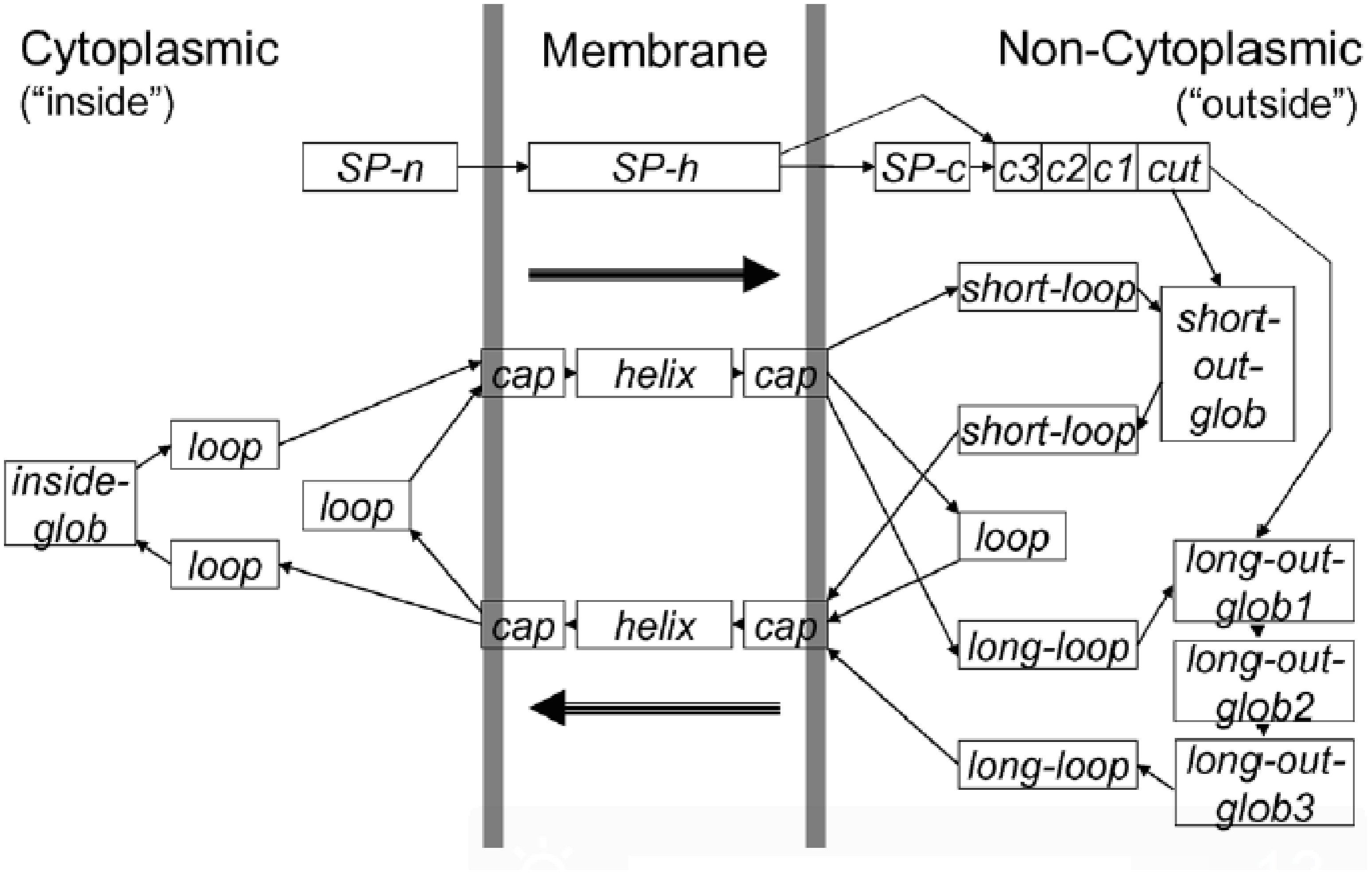
Sheila M. Reynolds[1], **Lukas Käll**[2], **Michael E. Riffle**[3], **Jeff A. Bilmes**[1,4], **William Stafford Noble**[2,4*]

1 Department of Electrical Engineering, University of Washington, Seattle, Washington, United States of America, 2 Department of Genome Sciences, University of Washington, Seattle, Washington, United States of America, 3 Department of Biochemistry, University of Washington, Seattle, Washington, United States of America, 4 Department of Computer Science and Engineering, University of Washington, Seattle, Washington, United States of America

Abstract

Hidden Markov models (HMMs) have been successfully applied to the tasks of transmembrane protein topology prediction and signal peptide prediction. In this paper we expand upon this work by making use of the more powerful class of dynamic Bayesian networks (DBNs). Our model, *Philius*, is inspired by a previously published HMM, Phobius, and combines a signal peptide submodel with a transmembrane submodel. We introduce a two-stage DBN decoder that combines the power of posterior decoding with the grammar constraints of Viterbi-style decoding. Philius also provides protein type, segment, and topology confidence metrics to aid in the interpretation of the predictions. We report a relative improvement of 13% over Phobius in full-topology prediction accuracy on transmembrane proteins, and a sensitivity and specificity of 0.96 in detecting signal peptides. We also show that our confidence metrics correlate well with the observed precision. In addition, we have made predictions on all 6.3 million proteins in the Yeast Resource Center (YRC) database. This large-scale study provides an overall picture of the relative numbers of proteins that include a signal-peptide and/or one or more transmembrane segments as well as a valuable resource for the scientific community. All DBNs are implemented using the Graphical Models Toolkit. Source code for the models described here is available at http://noble.gs.washington.edu/proj/philius. A Philius Web server is available at http://www.yeastrc.org/philius, and the predictions on the YRC database are available at http://www.yeastrc.org/pdr.

Citation: Reynolds SM, Käll L, Riffle ME, Bilmes JA, Noble WS (2008) Transmembrane Topology and Signal Peptide Prediction Using Dynamic Bayesian Networks. PLoS Comput Biol 4(11): e1000213. doi:10.1371/journal.pcbi.1000213


Cytoplasmic
("inside")
Membrane
Non-Cytoplasmic
("outside")
SP-n
SP-h
SP-c
c3
c2
c1
cut
short-loop
short-out-glob
short-loop
cap
helix
cap
loop
inside-glob
loop
loop
loop
long-out-glob1
long-loop
long-out-glob2
cap
helix
cap
long-loop
long-out-glob3

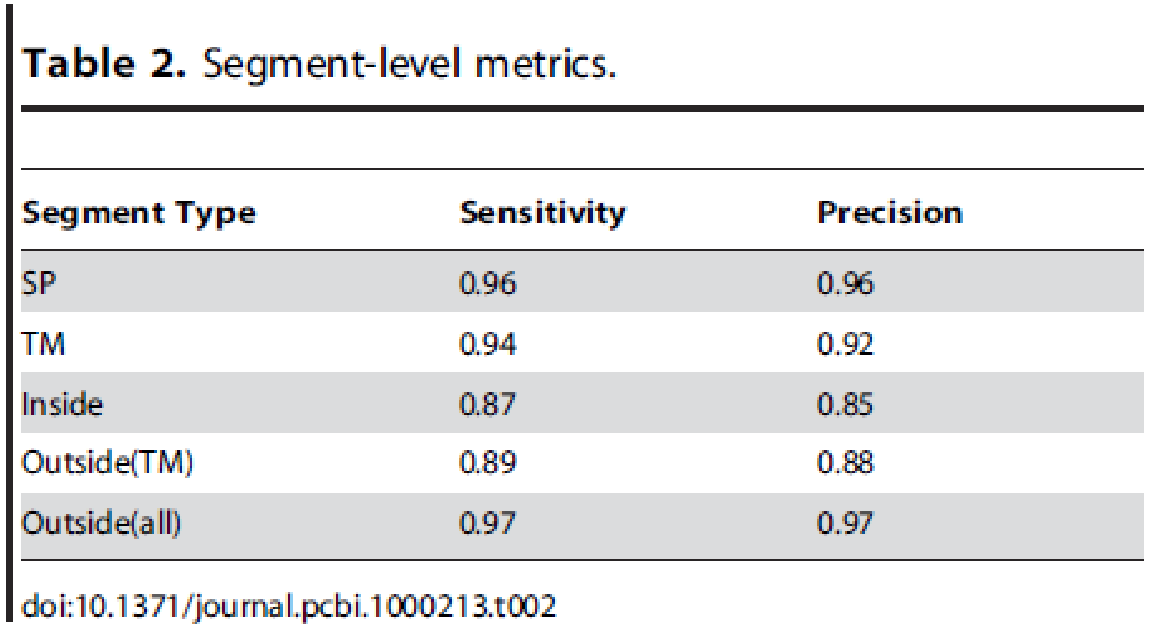
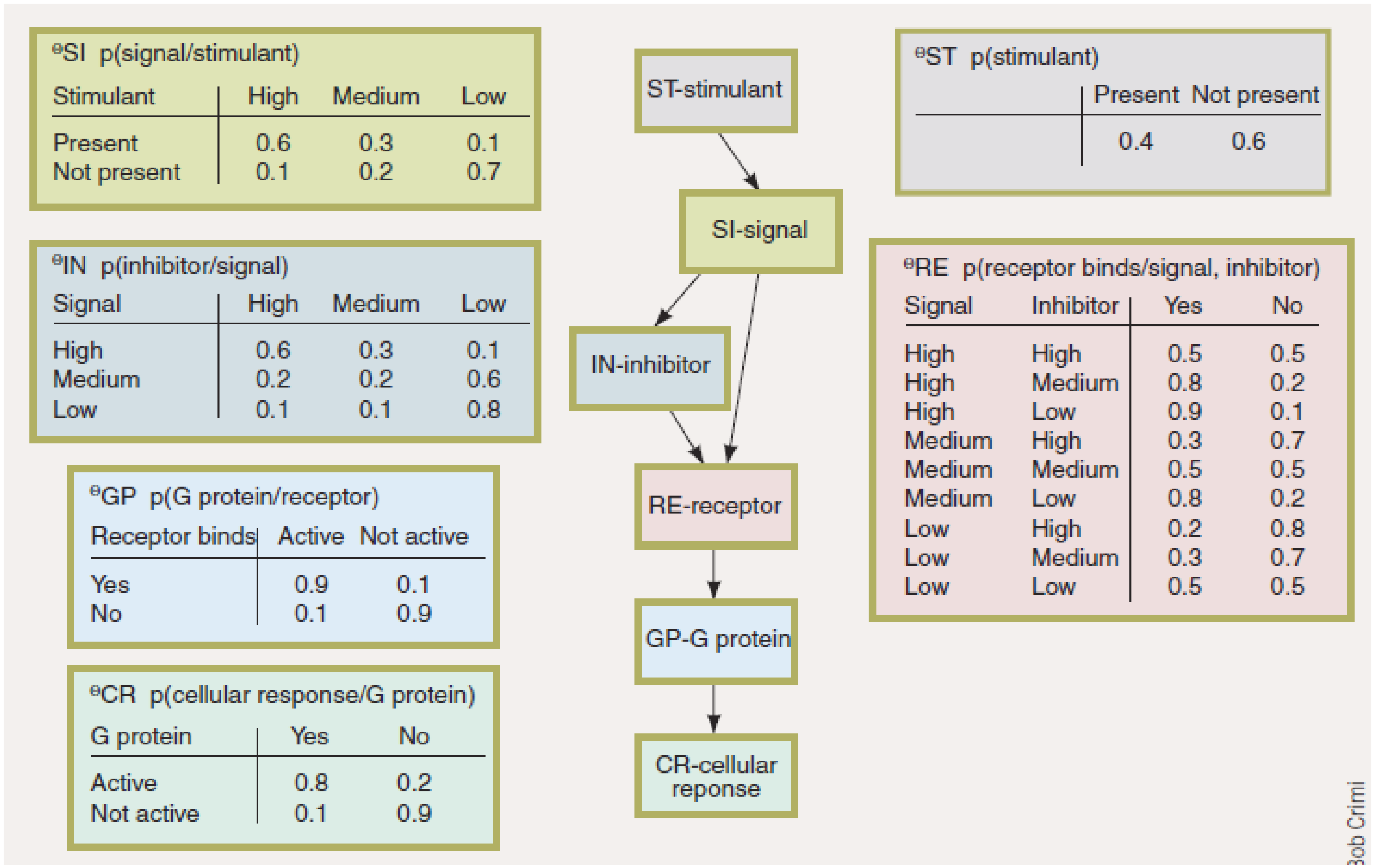


Table 2. Segment-level metrics.

Segment Type	Sensitivity	Precision
SP	0.96	0.96
TM	0.94	0.92
Inside	0.87	0.85
Outside(TM)	0.89	0.88
Outside(all)	0.97	0.97

doi:10.1371/journal.pcbi.1000213.t002

Bayesian networks for cellular signaling



^{e}SI p(signal/stimulant)

Stimulant	High	Medium	Low
Present	0.6	0.3	0.1
Not present	0.1	0.2	0.7

^{e}ST p(stimulant)

Present	Not present
0.4	0.6

^{e}IN p(inhibitor/signal)

Signal	High	Medium	Low
High	0.6	0.3	0.1
Medium	0.2	0.2	0.6
Low	0.1	0.1	0.8

^{e}RE p(receptor binds/signal, inhibitor)

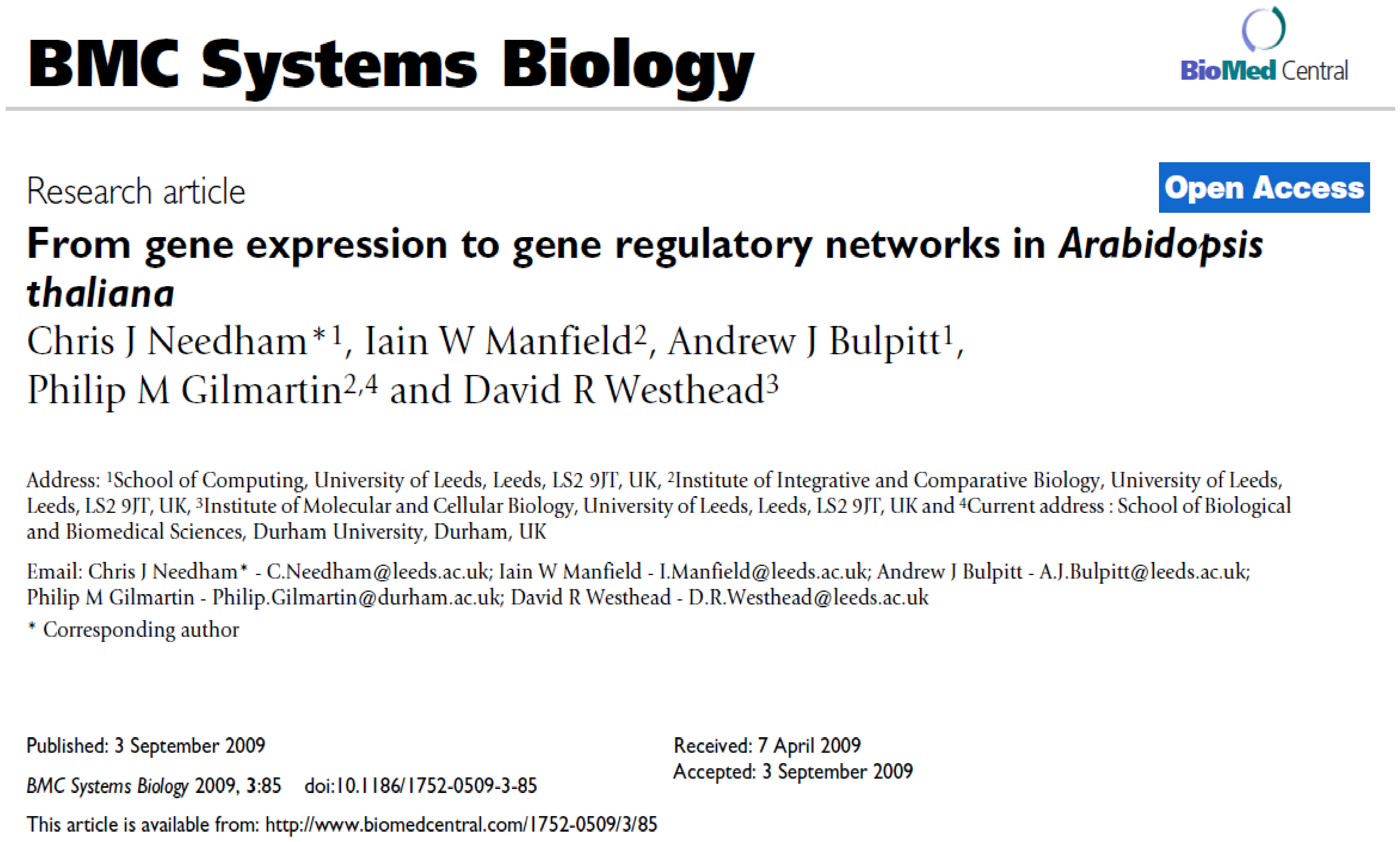
Signal	Inhibitor	Yes	No
High	High	0.5	0.5
High	Medium	0.8	0.2
High	Low	0.9	0.1
Medium	High	0.3	0.7
Medium	Medium	0.5	0.5
Medium	Low	0.8	0.2
Low	High	0.2	0.8
Low	Medium	0.3	0.7
Low	Low	0.5	0.5

^{e}GP p(G protein/receptor)

Receptor binds	Active	Not active
Yes	0.9	0.1
No	0.1	0.9

^{e}CR p(cellular response/G protein)

G protein	Yes	No
Active	0.8	0.2
Not active	0.1	0.9



BMC Systems Biology

BioMed Central

Research article

Open Access

From gene expression to gene regulatory networks in *Arabidopsis thaliana*

Chris J Needham*[1], Iain W Manfield[2], Andrew J Bulpitt[1], Philip M Gilmartin[2,4] and David R Westhead[3]

Address: [1]School of Computing, University of Leeds, Leeds, LS2 9JT, UK, [2]Institute of Integrative and Comparative Biology, University of Leeds, Leeds, LS2 9JT, UK, [3]Institute of Molecular and Cellular Biology, University of Leeds, Leeds, LS2 9JT, UK and [4]Current address : School of Biological and Biomedical Sciences, Durham University, Durham, UK

Email: Chris J Needham* - C.Needham@leeds.ac.uk; Iain W Manfield - I.Manfield@leeds.ac.uk; Andrew J Bulpitt - A.J.Bulpitt@leeds.ac.uk; Philip M Gilmartin - Philip.Gilmartin@durham.ac.uk; David R Westhead - D.R.Westhead@leeds.ac.uk

* Corresponding author

Published: 3 September 2009

BMC Systems Biology 2009, **3**:85 doi:10.1186/1752-0509-3-85

This article is available from: http://www.biomedcentral.com/1752-0509/3/85

Received: 7 April 2009
Accepted: 3 September 2009

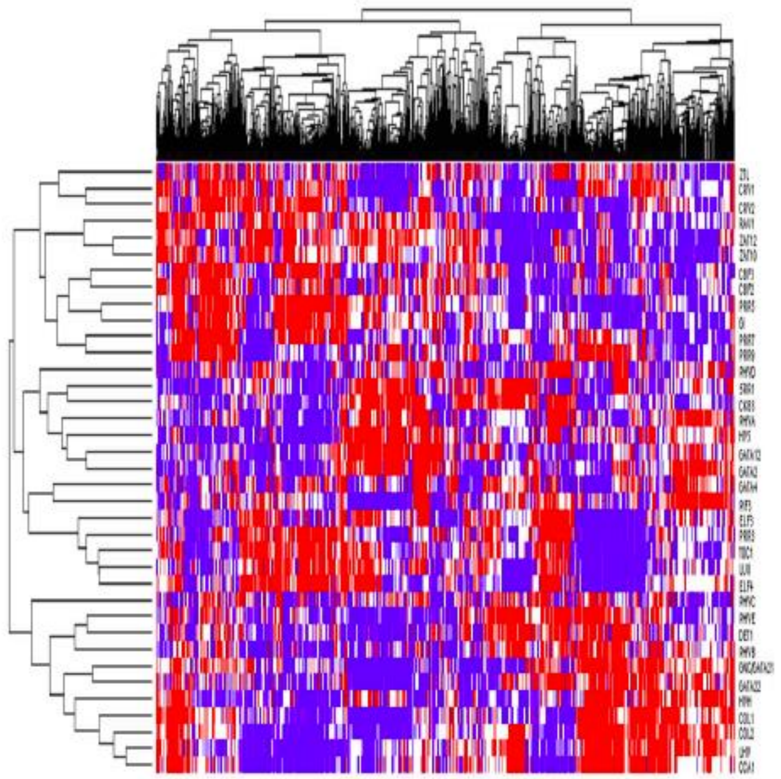


Figure 4
Clustergram of quantized gene expression profiles for 37 genes of interest, over 2904 microarrays. Both genes and experiments have been clustered. The three classes representing the low, medium and high classes are coloured blue, white and red respectively.

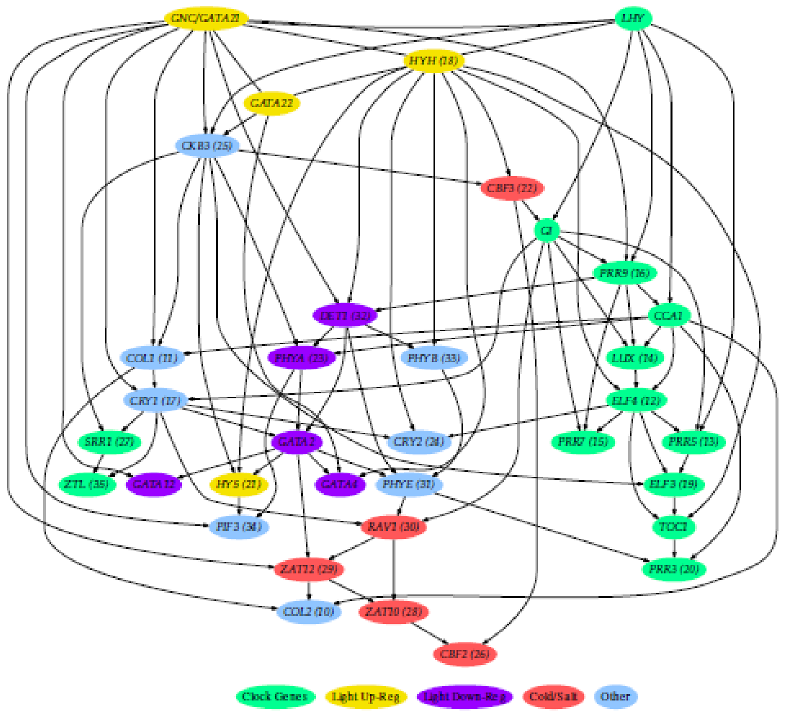


Figure 5
Learned regulatory network for other networks and poorly-characterized genes. The learned network structure starting from a set of nine genes (four clock and five *GATA* genes of interest), with additional genes added to the network from a selection of 37 genes. The number in parentheses next to the gene name denotes the order it was added to the network. Most of these genes were added to the network in early iterations, however, genes such as *SRR1* and *ZTL* with *bona fide* roles in the clock were added late and only indirectly linked to other clock components. All these interactions are very similar throughout the later iterations, once most of these components have been added to the network.