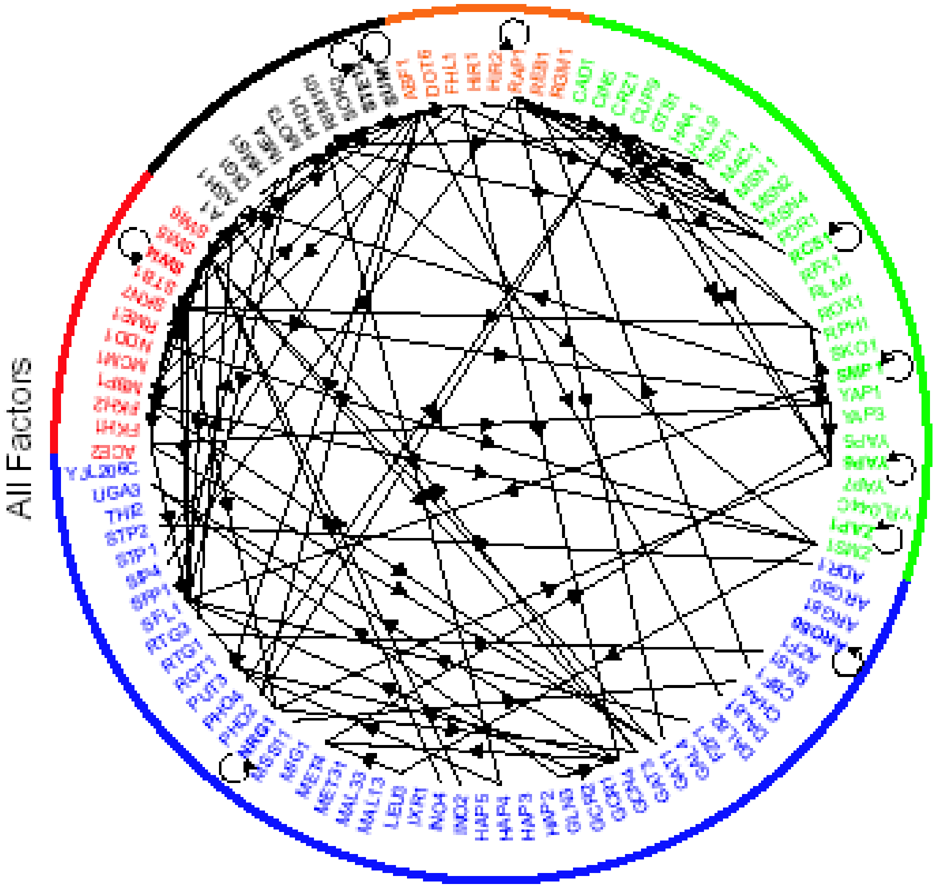
第7章: Regulatory Network

- Regulatory network
- Reverse engineering
- Bayesian network
- Gaussian Graphical Model

Part I: Regulatory Network



Lee et al. Science 2002.

Transcription Regulatory Code

- Each gene is regulated by a set of TFs.
- Each TF can regulate many genes.
- Which **genes** are regulated by which **TFs** on which **conditions**?
- How does regulator control the expression of its target gene?

How to Clarify Transcription Regulatory code ?

In silico.

- From sequence to gene regulatory network.
- Find all the potential TFBS upstream a gene.
- *Predict gene expression from gene sequence. Cell,2004.*

↓

Too much noise!

Experimental methods

- Gel shift
- DNA footprinting.
- Reporter genes
- ….

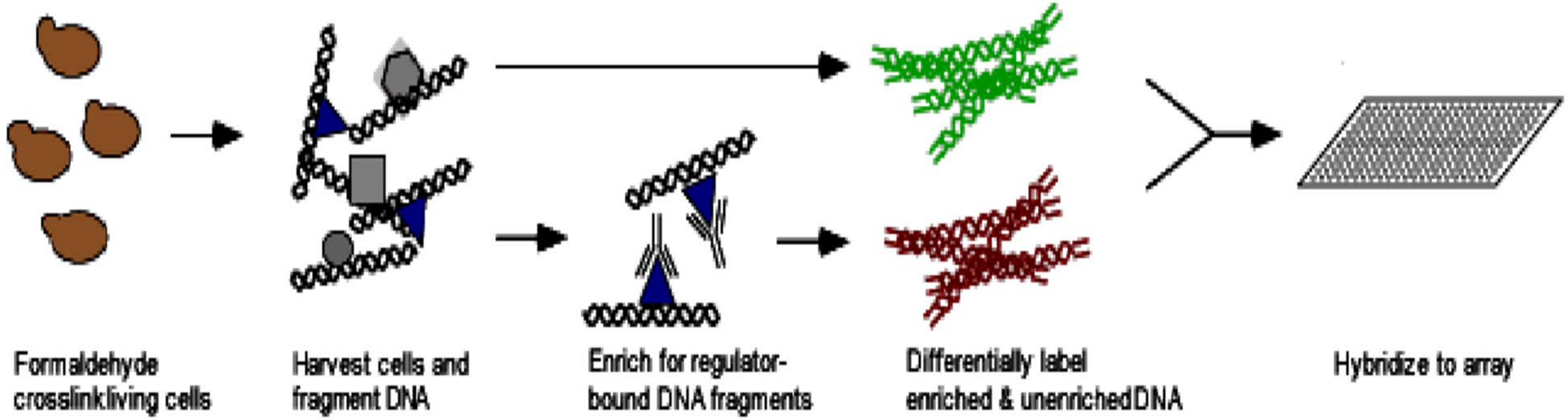
↓

Not large scale

Not systematic

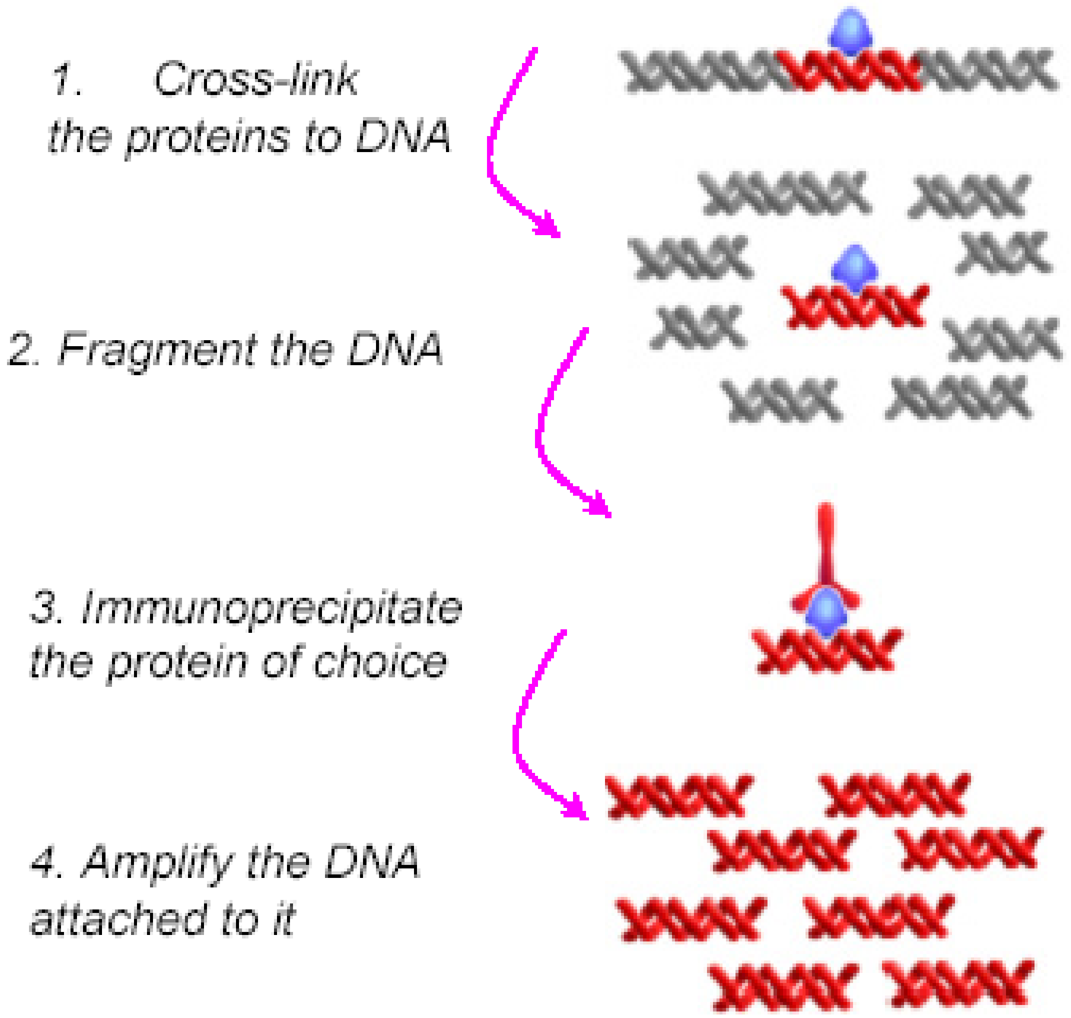
ChIP-chip Experiments

Identify all the target genes that can be directly or indirectly bind by a TF.



Formaldehyde crosslinkliving cells
Harvest cells and fragment DNA
Enrich for regulator-bound DNA fragments
Differentially label enriched & unenrichedDNA
Hybridize to array

ChIP-chip Experiments



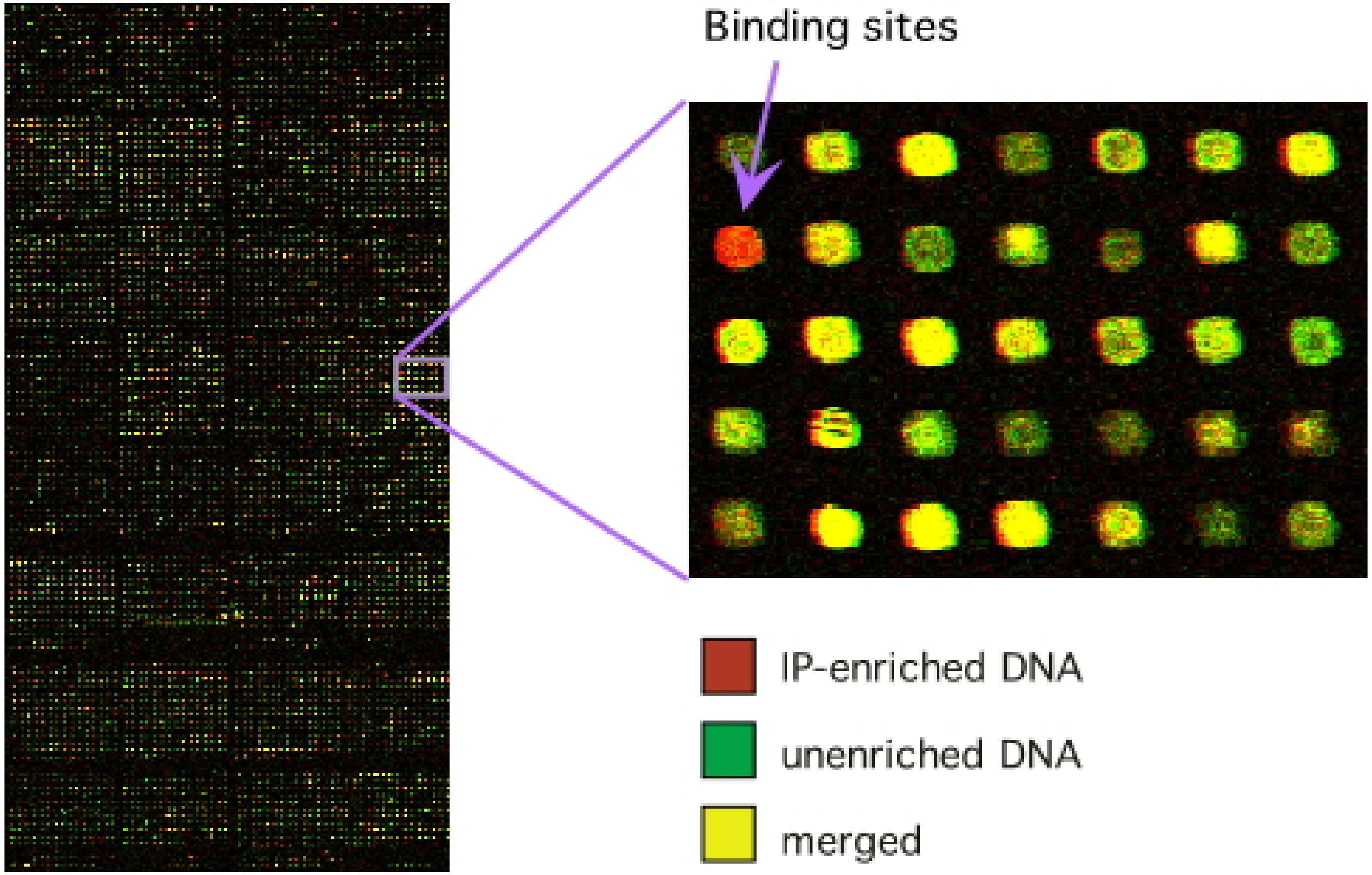
1. Cross-link the proteins to DNA

2. Fragment the DNA

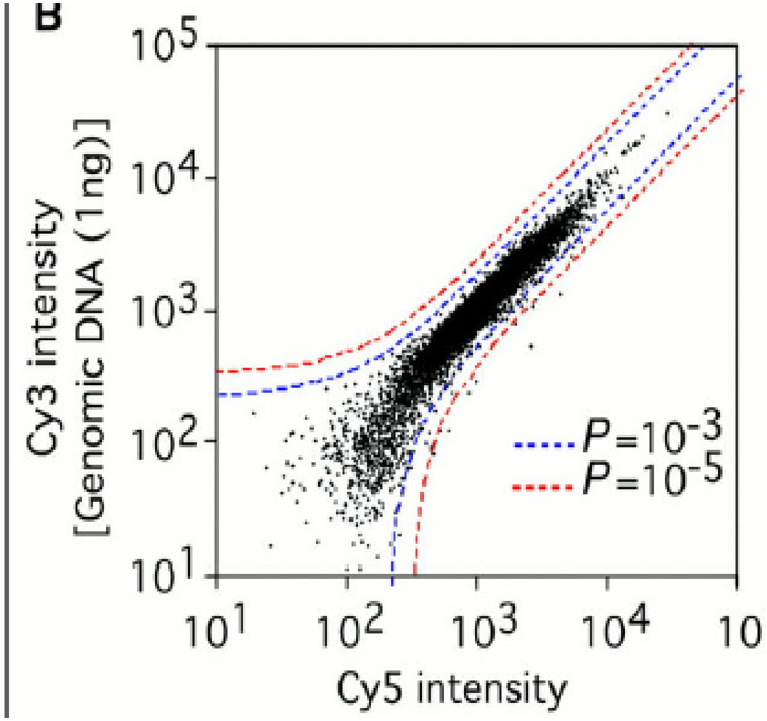
3. Immunoprecipitate the protein of choice

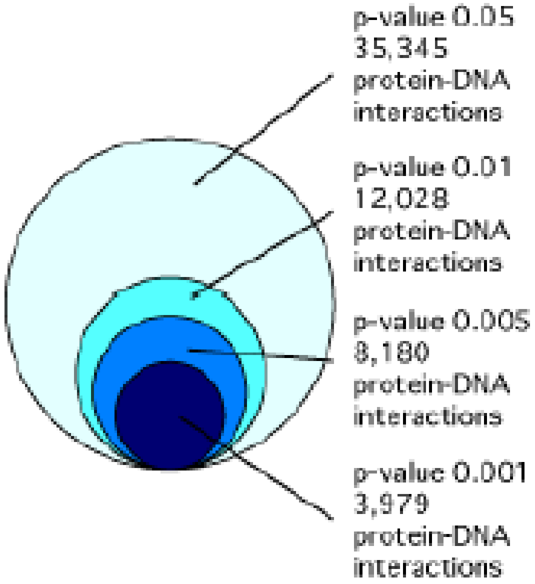
4. Amplify the DNA attached to it

ChIP-chip Experiments



Protein-DNA Interactions





Lee, et al. Science, 2002.

ChIP-chip Experiments

- 1 condition , 1 TF

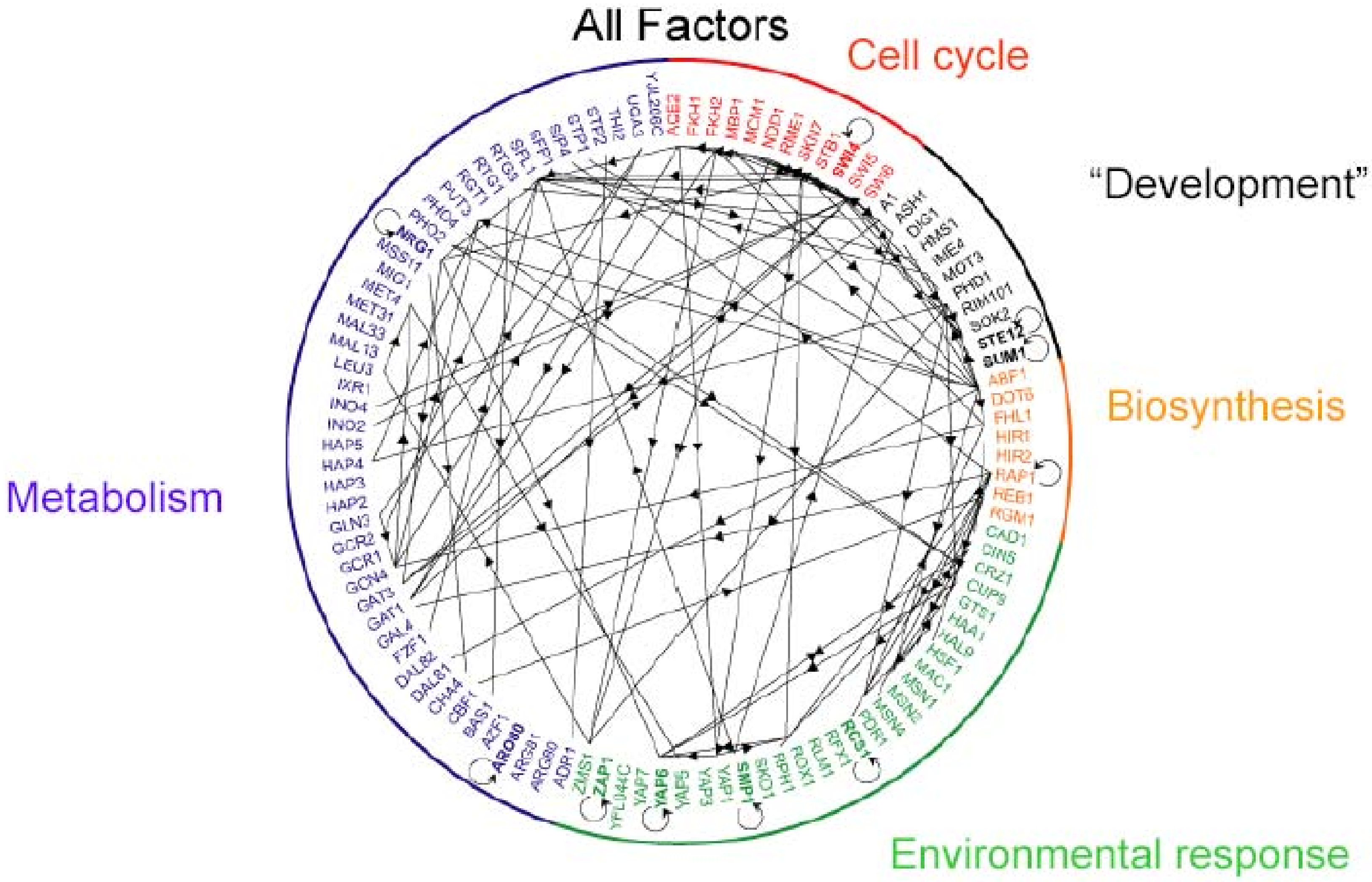
  Jason et.al. Nature (2001). Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association.

- 1 condition, 106 TFs

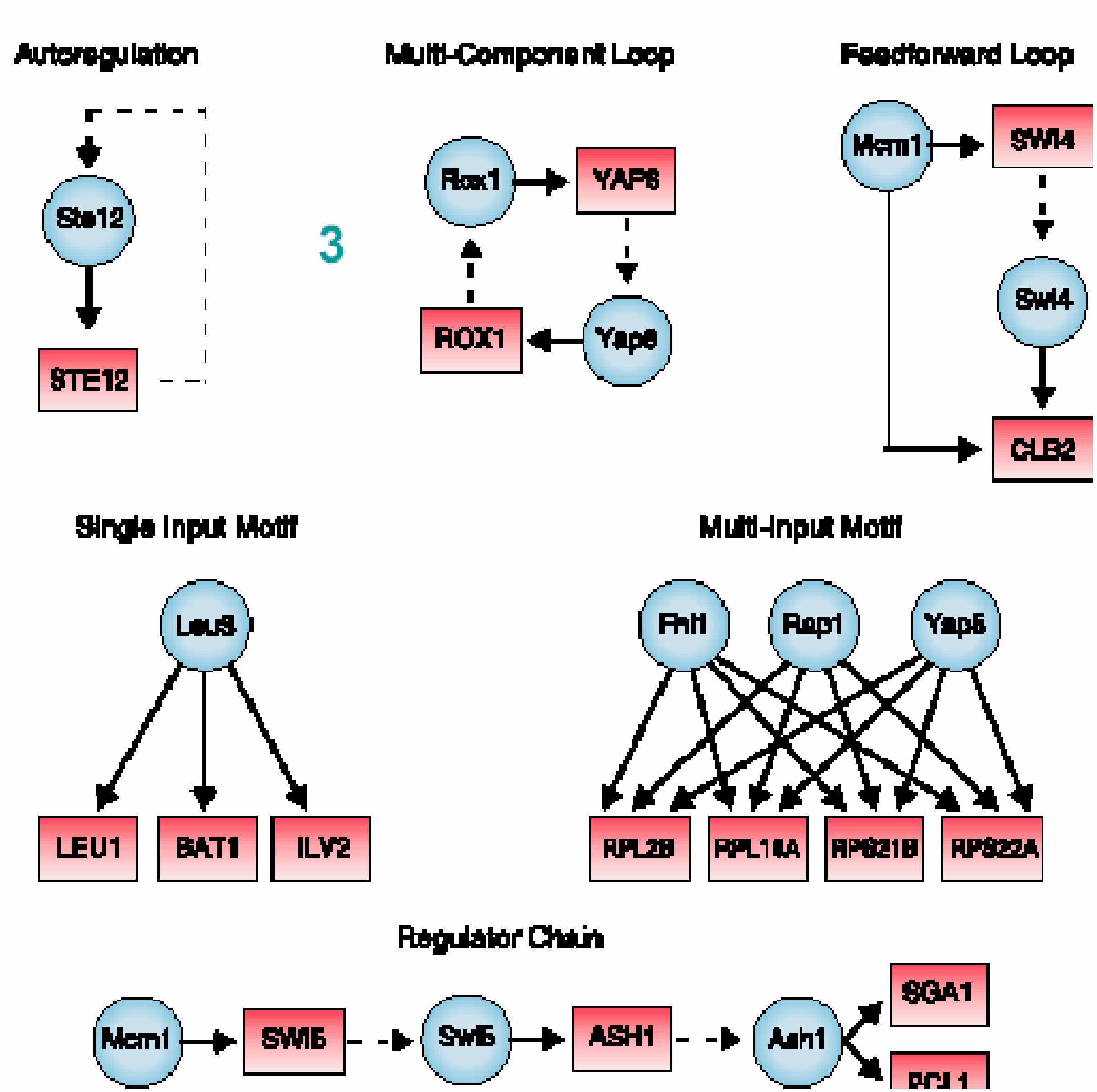
  Lee et.al. Science(2002). Transcription regulatory networks in *Saccharomyces cerevisiae*

- Multiple conditions , 203 TFs.

  Harbison, et.al. (2004). Transcription regulatory code of a eukaryotic genome.



Lee, et al. Science, 2002.



10

3

49
(39 TF;
240 genes)

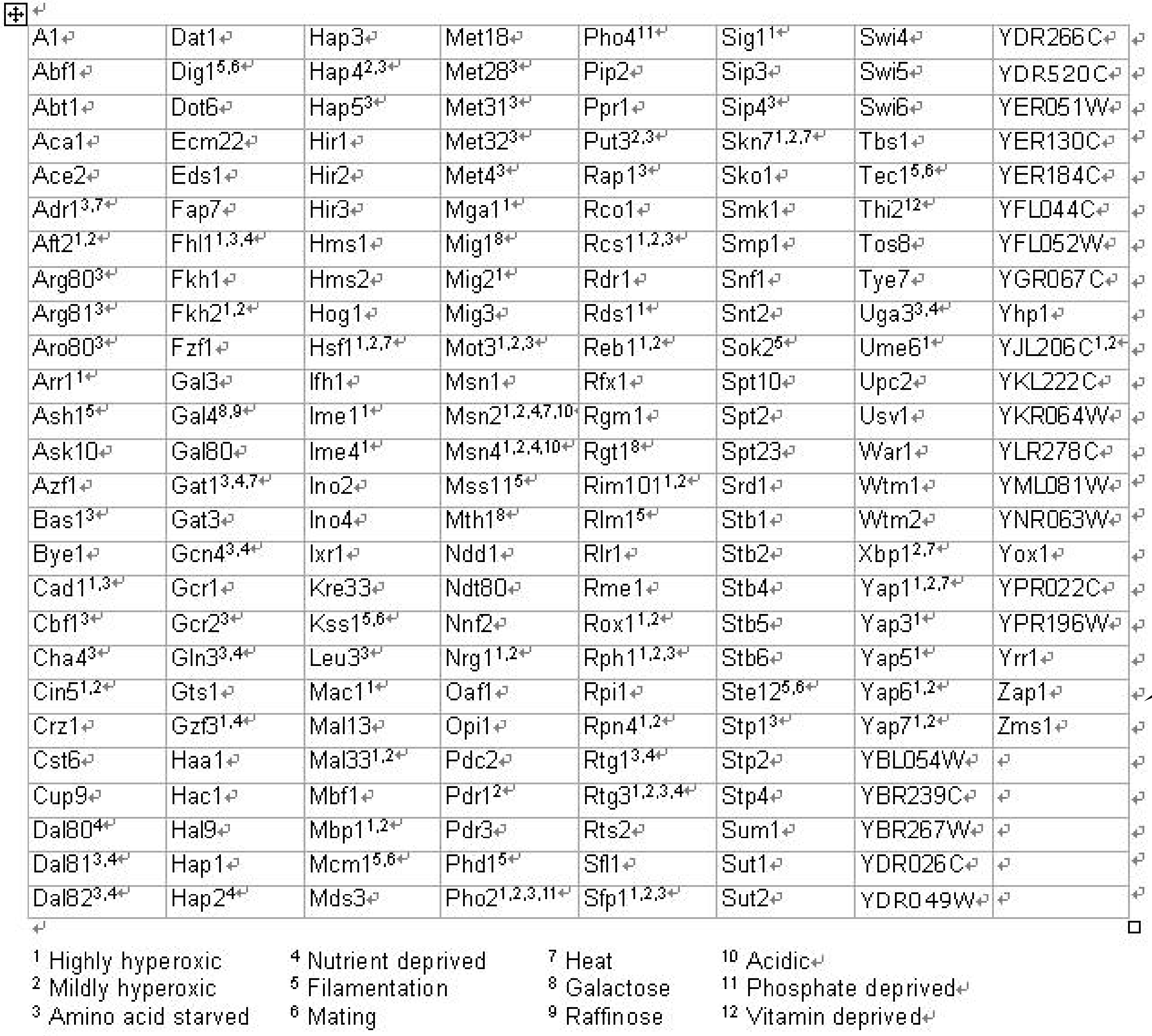
90

81

188

Lee, et al. Science, 2002.

Multiple Conditions



A1	Dat1	Hap3	Met18	Pho4[11]	Sig1[1]	Swi4	YDR266C
Abf1	Dig1[5,6]	Hap4[2,3]	Met28[3]	Pip2	Sip3	Swi5	YDR520C
Abt1	Dot6	Hap5[3]	Met31[3]	Ppr1	Sip4[3]	Swi6	YER051W
Aca1	Ecm22	Hir1	Met32[3]	Put3[2,3]	Skn7[1,2,7]	Tbs1	YER130C
Ace2	Eds1	Hir2	Met4[3]	Rap1[3]	Sko1	Tec1[5,6]	YER184C
Adr1[3,7]	Fap7	Hir3	Mga1[1]	Rco1	Smk1	Thi2[12]	YFL044C
Aft2[1,2]	Fhl1[1,3,4]	Hms1	Mig1[8]	Rcs1[1,2,3]	Smp1	Tos8	YFL052W
Arg80[3]	Fkh1	Hms2	Mig2[1]	Rdr1	Snf1	Tye7	YGR067C
Arg81[3]	Fkh2[1,2]	Hog1	Mig3	Rds1[1]	Snt2	Uga3[3,4]	Yhp1
Aro80[3]	Fzf1	Hsf1[1,2,7]	Mot3[1,2,3]	Reb1[1,2]	Sok2[5]	Ume6[1]	YJL206C[1,2]
Arr1[1]	Gal3	Ifh1	Msn1	Rfx1	Spt10	Upc2	YKL222C
Ash1[5]	Gal4[8,9]	Ime1[1]	Msn2[1,2,4,7,10]	Rgm1	Spt2	Usv1	YKR064W
Ask10	Gal80	Ime4[1]	Msn4[1,2,4,10]	Rgt1[8]	Spt23	War1	YLR278C
Azf1	Gat1[3,4,7]	Ino2	Mss11[5]	Rim101[1,2]	Srd1	Wtm1	YML081W
Bas1[3]	Gat3	Ino4	Mth1[8]	Rlm1[5]	Stb1	Wtm2	YNR063W
Bye1	Gcn4[3,4]	Ixr1	Ndd1	Rlr1	Stb2	Xbp1[2,7]	Yox1
Cad1[1,3]	Gcr1	Kre33	Ndt80	Rme1	Stb4	Yap1[1,2,7]	YPR022C
Cbf1[3]	Gcr2[3]	Kss1[5,6]	Nnf2	Rox1[1,2]	Stb5	Yap3[1]	YPR196W
Cha4[3]	Gln3[3,4]	Leu3[3]	Nrg1[1,2]	Rph1[1,2,3]	Stb6	Yap5[1]	Yrr1
Cin5[1,2]	Gts1	Mac1[1]	Oaf1	Rpi1	Ste12[5,6]	Yap6[1,2]	Zap1
Crz1	Gzf3[1,4]	Mal13	Opi1	Rpn4[1,2]	Stp1[3]	Yap7[1,2]	Zms1
Cst6	Haa1	Mal33[1,2]	Pdc2	Rtg1[3,4]	Stp2	YBL054W	
Cup9	Hac1	Mbf1	Pdr1[2]	Rtg3[1,2,3,4]	Stp4	YBR239C	
Dal80[4]	Hal9	Mbp1[1,2]	Pdr3	Rts2	Sum1	YBR267W	
Dal81[3,4]	Hap1	Mcm1[5,6]	Phd1[5]	Sfl1	Sut1	YDR026C	
Dal82[3,4]	Hap2[4]	Mds3	Pho2[1,2,3,11]	Sfp1[1,2,3]	Sut2	YDR049W	

[1] Highly hyperoxic
[2] Mildly hyperoxic
[3] Amino acid starved
[4] Nutrient deprived
[5] Filamentation
[6] Mating
[7] Heat
[8] Galactose
[9] Raffinose
[10] Acidic
[11] Phosphate deprived
[12] Vitamin deprived

- *All regulators were profiled in rich medium
- *A subset of these were profiled in at least one of 12 other environmental condition

Harbison et al.
Nature 2004.

Part II: Reverse Engineering

- Given: a (large) set of gene expression observations
- Goal: find the network fits that observation data.

- References:
  - Gardner, di Bernardo, Lorenz, and Collins. Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science* **301**, pp.102-105 (2003)
  - Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene van Someren, Reinhard Guthke. Gene regulatory network inference: Data integration in dynamic models—A review. BioSystems 96 (2009) 86–103.

DREAM Project

- DREAM: Dialogue for Reverse Engineering Assessments and Methods.
- Objective: To catalyze the interaction between experiment and theory in the area of cellular network inference and quantitative model building in systems biology.
- http://www.the-dream-project.org/
- http://sagebase.org/challenges-overview/2013-dream-challenges/ (DREAM8)

Modeling Expression with Differential Equations

Assumes network behavior can be modeled as a system of linear differential equations of the form:

$$\frac{dx}{dt} = Ax + u$$

x is a vector representing the continuous-valued levels (concentrations) of each network component

A is the network model: an *N* x *N* matrix of coefficients describing how each x_i is controlled by upstream genes x_j, x_k, *etc.*

u is a vector representing an external additive perturbation to the system

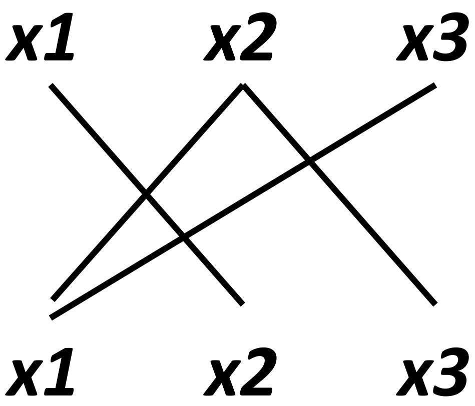
An example:
From discrete- to continuous-valued networks

Three genes: x_1, x_2, x_3

x1 activates *x2*

x2 activates *x1* and *x3*

x3 inhibits *x1*



$$d\mathbf{x}/dt = \mathbf{A}\mathbf{x} + \mathbf{u}$$

$$dx_1/dt = a_{12}x_2 - a_{13}x_3$$

$$dx_2/dt = a_{21}x_1$$

$$dx_3/dt = a_{32}x_2$$

$$\boxed{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & a_{12} & -a_{13} \\ a_{21} & 0 & 0 \\ 0 & a_{32} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}$$

The steady state assumption

- Near a steady-state point, expression levels do not change over time.
- Under the steady-state assumption, the model reduces to $0 = \mathbf{Ax} + \mathbf{u} \rightarrow \mathbf{Ax} = -\mathbf{u}$
- A straightforward method to infer **A** would be to apply *N* perturbations, **u**, to the network, in each case measuring steady-state expression levels for the **x**.
- However, in larger networks it may be impractical to apply so many perturbations
- As a simplifying assumption, ***consider that each gene has a maximum of k non-zero regulatory inputs***.

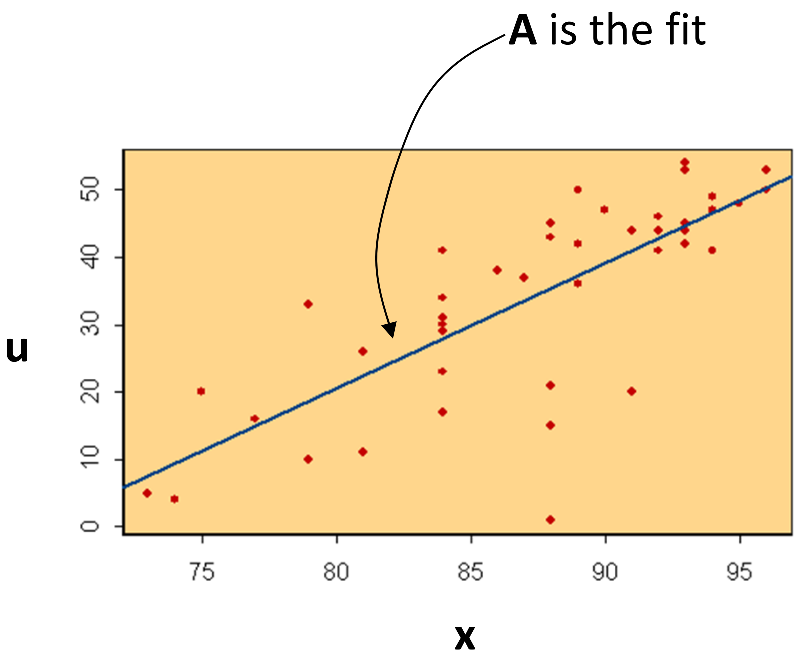
The Inference Procedure

$$\mathbf{Ax} = -\mathbf{u}$$

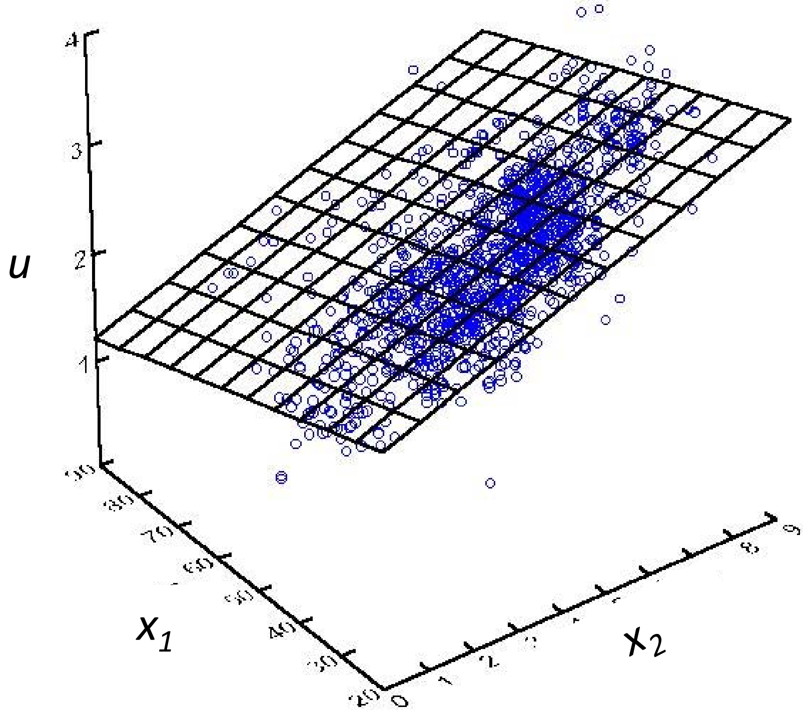
- Infer inputs to each gene separately
- For the given gene, consider all possible combinations of the *k* regulatory inputs
- For each combination, use multiple linear regression to determine optimal values of the *k* coefficients
- Choose the combination that fits the observed data with the least error

Multiple regression

$$\mathbf{u} = -\mathbf{A}\mathbf{x}$$

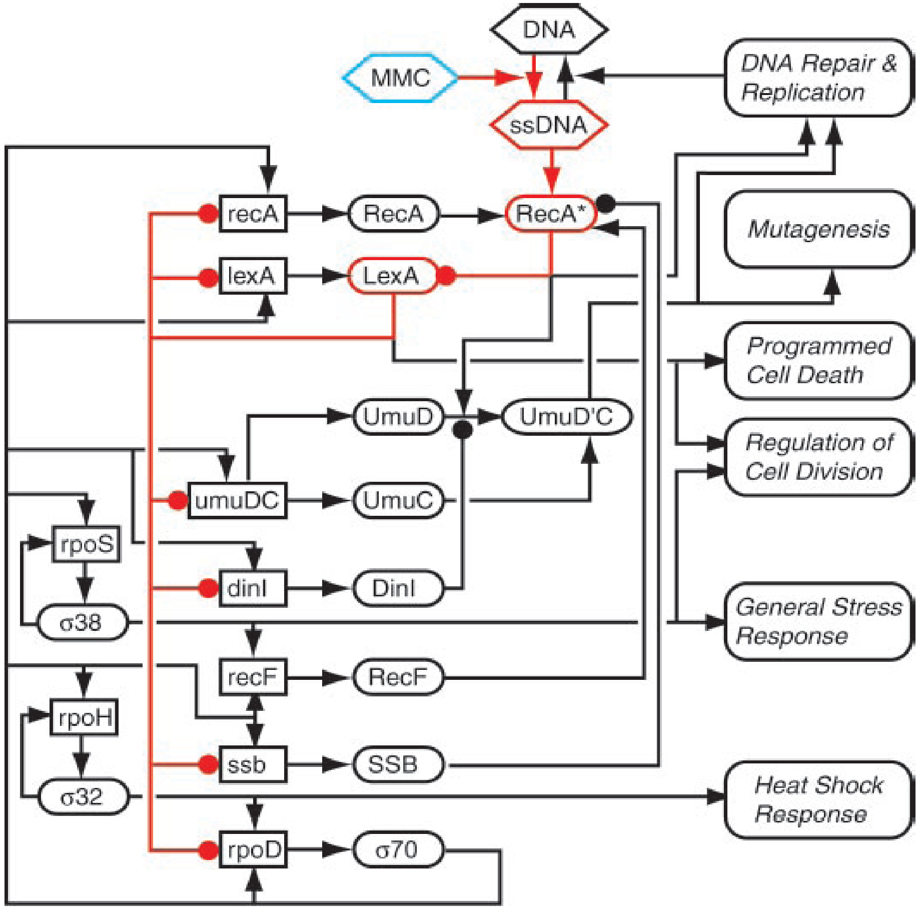


A is the fit



Application to SOS System

Fig. 1. Diagram of interactions in the SOS network. DNA lesions caused by mitomycin C (MMC) (blue hexagon) are converted to single-stranded DNA during chromosomal replication. Upon binding to ssDNA, the RecA protein is activated (RecA*) and serves as a coprotease for the LexA protein. The LexA protein is cleaved, thereby diminishing the repression of genes that mediate multiple protective responses. Boxes denote genes, ellipses denote proteins, hexagons indicate metabolites, arrows denote positive regulation, filled circles denote negative regulation. Red emphasis denotes the primary pathway by which the network is activated after DNA damage.



Gardner, di Bernardo, Lorenz, and Collins. Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science* **301**, pp.102-105 (2003)

Part III: Bayesian Network

- 本部分Slides主要来自于N.Friedman and D.Heckman’s slides.

- References:
- N.Friedman et al. Using Bayesian Networks to analyze expression data. *J. Comput. Biol.,* **7**:601-620, 2000.

Motivation

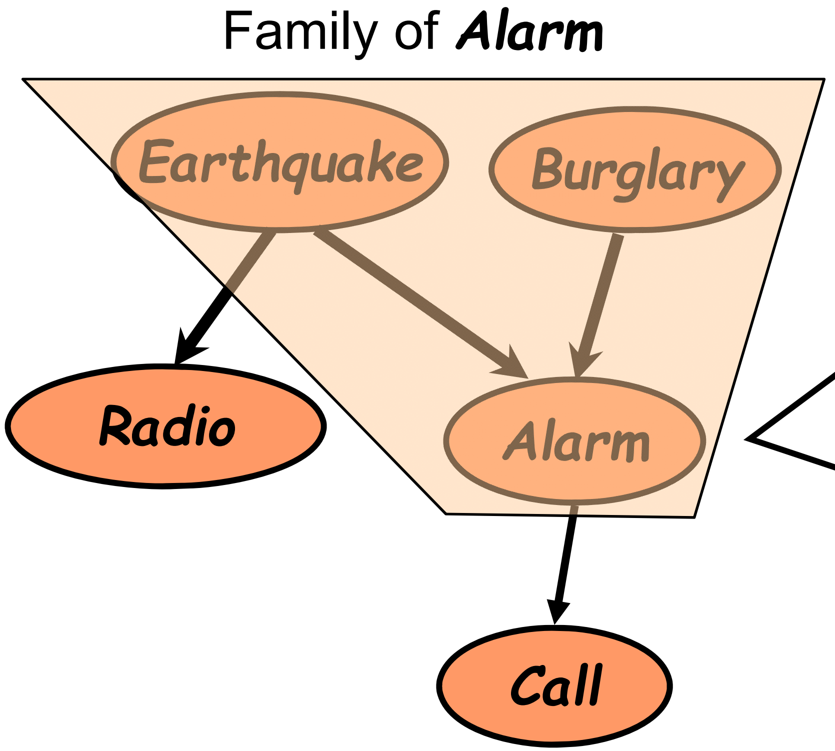
- Given gene expression data, what's the relationship between genes?
  - Who regulates who?
  - How does does one gene regulate other gene?
- Exploring the relationship among features to construct a better classifier instead of treating them independently.

Bayesian Network

- Directed acyclic graph (DAG).
  - Nodes: random variables.
  - Edges: direct influence.
- Set of conditional probability distributions.
- Joint distribution.

$$P(X) = \prod_{i=1}^{n} p(X_i|\text{parents}(X_i))$$

Bayesian Networks: Example



E	B	P(A | E,B)	
e	b	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	b	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

$$P(B,E,A,C,R) = P(B)P(E)P(A|B,E)P(R|E)P(C|A)$$

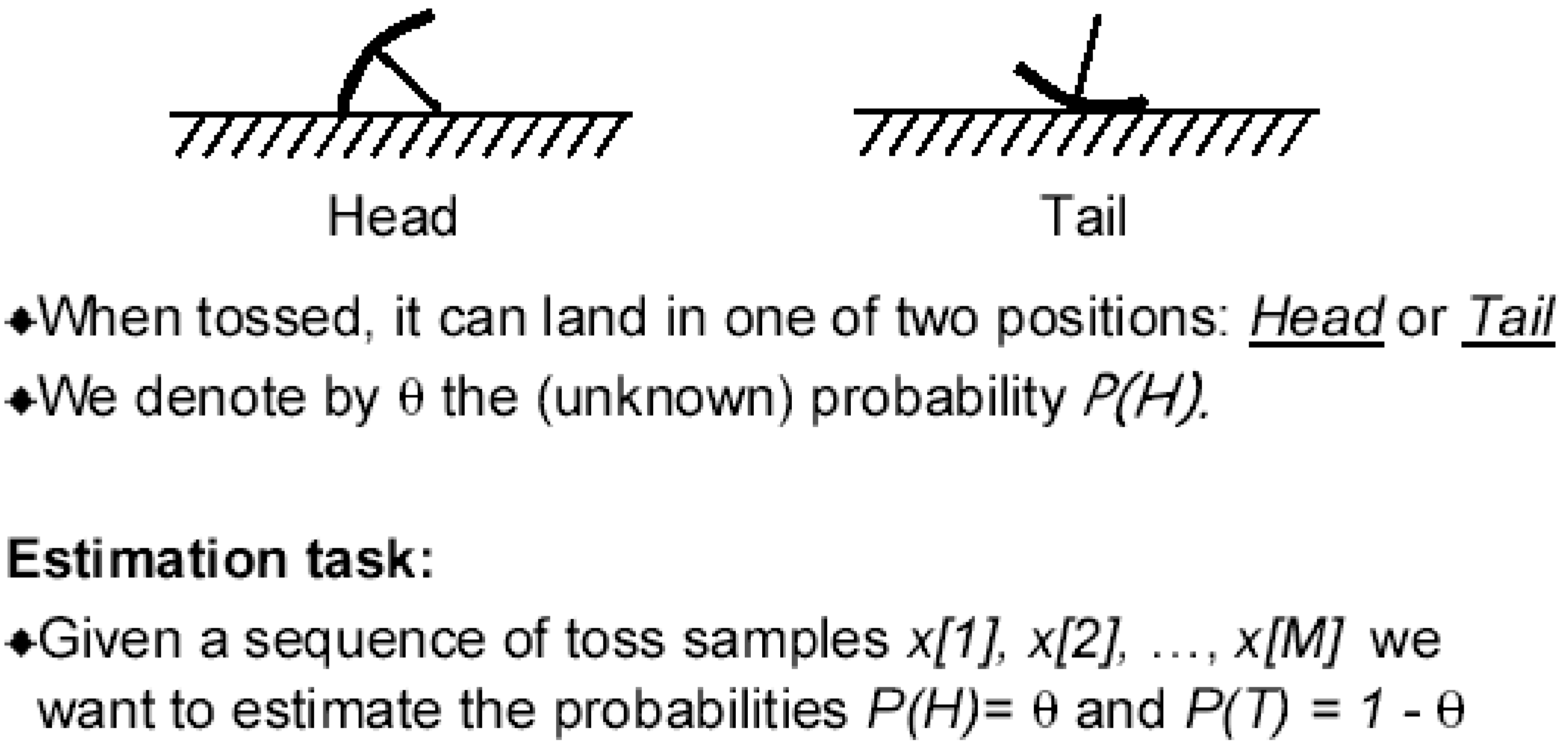
Learning Problems

- Estimation of the parameters.
- Construct the structure.

Let’s start from the basic parameter estimation problem.

A: Learning Parameters

Simple Case: Binomial Experiment



- When tossed, it can land in one of two positions: *Head* or *Tail*
- We denote by θ the (unknown) probability *P(H)*.

Estimation task:

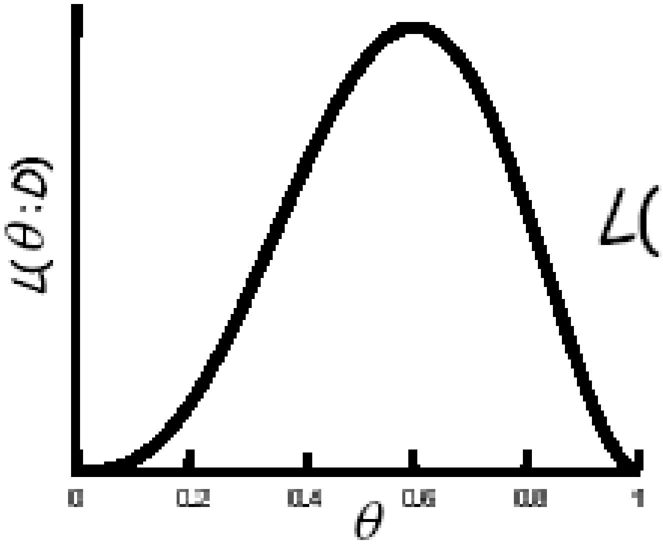
- Given a sequence of toss samples *x[1], x[2], …, x[M]* we want to estimate the probabilities *P(H)*= θ and *P(T)* = *1* - θ

Likelihood Function

- How good is a particular θ?
  It depends on how likely it is to generate the observed data

$$\mathcal{L}(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

- Thus, the likelihood for the sequence H,T, T, H, H is



$$\mathcal{L}(\theta : D) = \theta \cdot (1 - \theta) \cdot (1 - \theta) \cdot \theta \cdot \theta$$

Sufficient Statistics

- To compute the likelihood in the thumbtack example we only require N_H and N_T (the number of heads and the number of tails)

  $$L(\theta : D) = \theta^{N_H} \cdot (1-\theta)^{N_T}$$

  N_H and N_T are **sufficient statistics** for the binomial distribution

- A **sufficient statistic** is a function that summarizes, from the data, the relevant information for the likelihood
  - If $s(D) = s(D')$, then $L(\theta \mid D) = L(\theta \mid D')$

Maximum Likelihood Estimation (MLE)

- MLE principle: **Learn parameters that maximize the likelihood function**.

- This is one of the most commonly used estimation in statistics (Classical approach) and intuitively appealing.

MLE In Binomial Case

- Applying the MLE principle we get

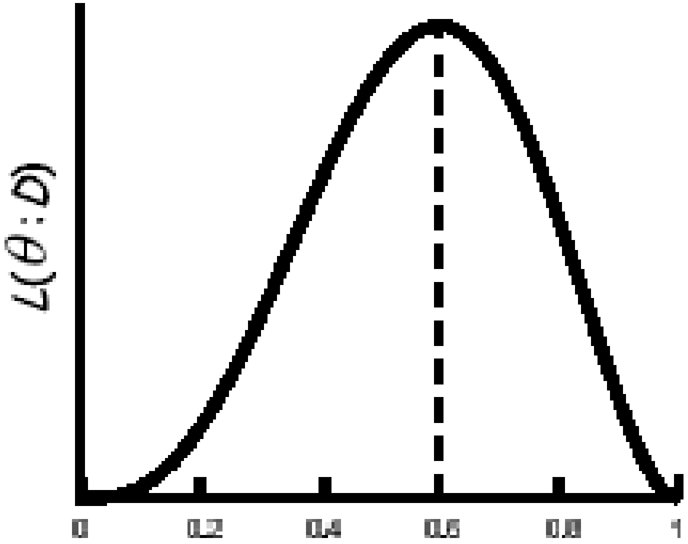
$$\hat{\theta} = \frac{N_H}{N_H + N_T}$$

(Which coincides with what one would expect)

Example:

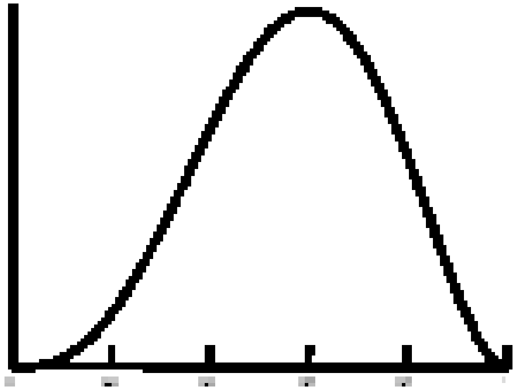
$(N_H, N_T) = (3,2)$

MLE estimate is 3/5 = 0.6



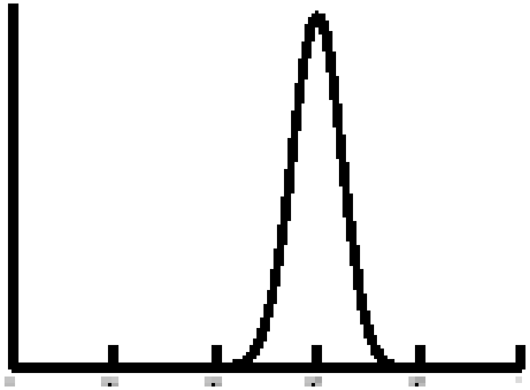
MLE is Not Enough

- MLE commits to a specific value of the unknown parameter(s)



Coin

vs.



Thumbtack

- MLE is the same in both cases
- Confidence in prediction is clearly different

Bayesian Inference

- Representing uncertainty about parameters using a probability distribution over parameters, data.
- Using Bayes' rule to learn.
  - Data (D) and their probability distribution $p(x|\xi)$
  - Prior distribution $p(\theta|\xi)$

$$\begin{cases} p(\theta|D,\xi) = \dfrac{p(\theta|\xi)p(D|\theta,\xi)}{p(D|\xi)} \\ p(D|\xi) = \displaystyle\int p(D|\theta,\xi)p(\theta|\xi)d\theta \end{cases}$$

Binomial Experiment Revised

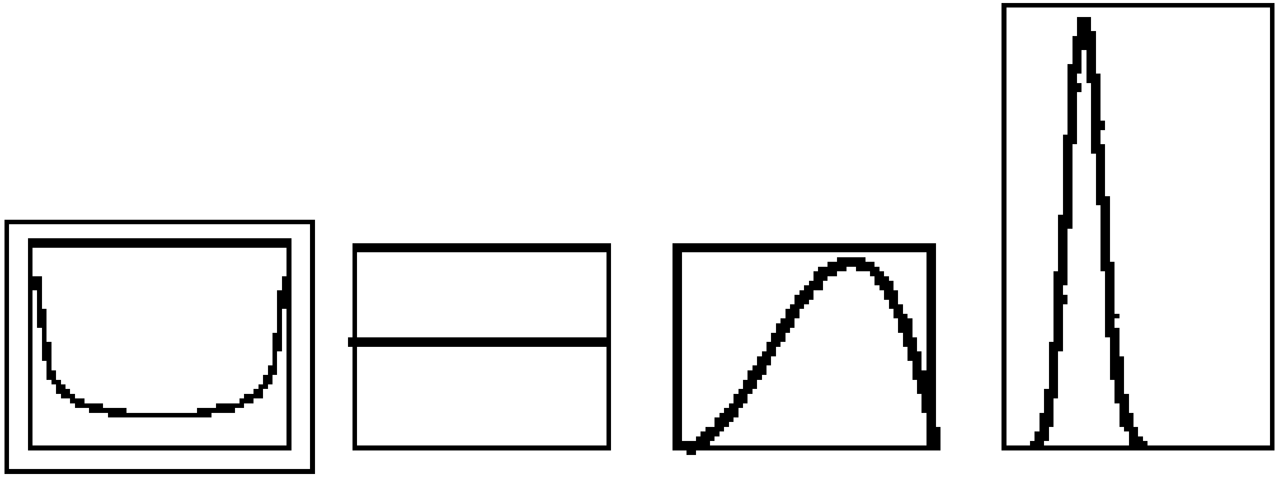
- Prior: Beta distribution

$$\begin{aligned} p(\theta) &= Beta(\alpha_H, \alpha_T) \\ &= \frac{\Gamma(\alpha_H + \alpha_T)}{\Gamma(\alpha_H)\Gamma(\alpha_T)} \theta^{\alpha_H - 1}(1-\theta)^{\alpha_T - 1} \end{aligned}$$

- Posterior

$$\begin{aligned} p(\theta|D) &= Beta(\alpha_H + N_H, \alpha_T + N_T) \\ &= \frac{\Gamma(\alpha_H + N_H + \alpha_T + N_T)}{\Gamma(\alpha_H + N_H)\Gamma(\alpha_T + N_T)} \theta^{N_H + \alpha_H - 1}(1-\theta)^{N_T + \alpha_T - 1} \end{aligned}$$

Beta Distribution



Beta(0.5, 0.5)   Beta(1, 1)   Beta(3, 2)   Beta(19, 39)

MAP (Maximum A-Posterior Probability)

- Using MAP, we can obtain an estimation of the parameter

$$\tilde{\theta} = \frac{\alpha_H + N_H}{\alpha_H + \alpha_T + N_H + N_T}$$

- Recall that the MLE is

$$\hat{\theta} = \frac{N_H}{N_H + N_T}$$

Intuition

- The hyperparameters α_H and α_T can be thought of imaginary counts (psudo-counts) from our experience.
- Equivalent sample size= $\alpha_H + \alpha_T$.
- The larger the equivalent sample size, the more confident we are about the true probability.

Bayesian Inference vs. MLE

- Frequentist approach
  - Assumes there is an unknown but fixed parameter
  - Estimates with some confidence
  - Prediction by using the estimated parameter value
- Bayesian approach
  - Represents uncertainty about the unknown parameter
  - Uses probability to quantify this uncertainty: unknown parameters as random variables
  - Prediction follows from the rules of probability: take expectation over the unknown parameters

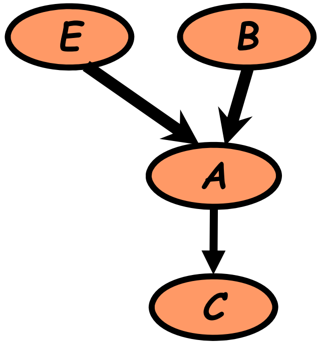
Bayesian Inference vs. MLE (Cont.)

- In our example, MLE and Bayesian prediction differ.

- However, If prior is well-behaved (does not assign 0 density to any feasible parameter value), then <u>both MLE and Bayesian prediction converge to the same value, the ”true” distribution</u>.

Learning Parameters

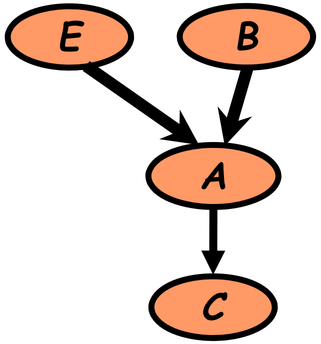
- Training data has the form:



$$D = \begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

Likelihood Function

- Assume i.i.d. samples
- Likelihood function is



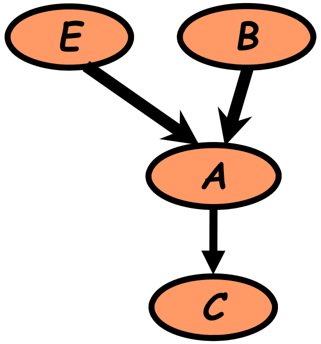
$$L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

Likelihood Function

- By definition of network, we get

$$L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

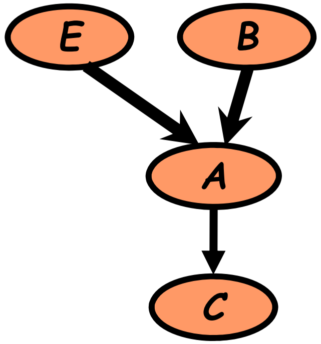
$$= \prod_m \begin{pmatrix} P(E[m] : \Theta) \\ P(B[m] : \Theta) \\ P(A[m] \mid B[m], E[m] : \Theta) \\ P(C[m] \mid A[m] : \Theta) \end{pmatrix}$$



$$\begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

Likelihood Function

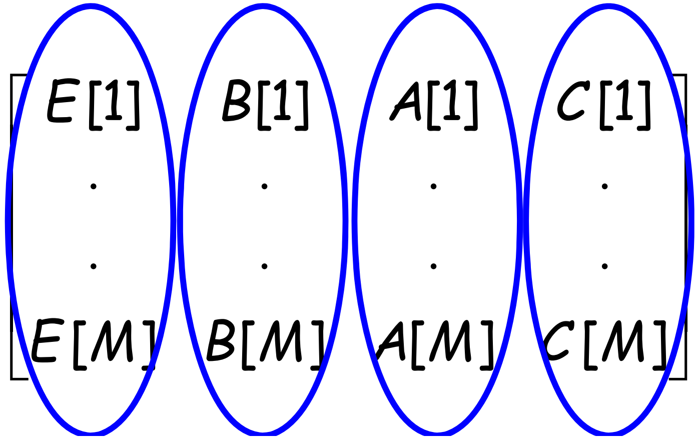
- Rewriting terms, we get



$$L(\Theta:D)=\prod_m P(E[m],B[m],A[m],C[m]:\Theta)$$

$$=\begin{array}{l}\prod_m P(E[m]:\Theta)\\ \prod_m P(B[m]:\Theta)\\ \prod_m P(A[m]\mid B[m],E[m]:\Theta)\\ \prod_m P(C[m]\mid A[m]:\Theta)\end{array}$$

4 Subnetworks



$$\begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

General Bayesian Networks

Generalizing for any Bayesian network:

$$
\begin{aligned}
L(\Theta : D) &= \prod_{m} P(x_1[m], \ldots, x_n[m] : \Theta) \\
&= \prod_{i} \prod_{m} P(x_i[m] \mid Pa_i[m] : \Theta_i) \\
&= \prod_{i} L_i(\Theta_i : D)
\end{aligned}
$$

The likelihood decomposes to small ones according to the structure of the network.

General Bayesian Networks (Cont.)

- **Decomposition $\Rightarrow$ Independent estimation problems**

- If the parameters for each family are not related, they can be estimated independently of each other.

From Binomial to Multinomial

- For example, suppose X can have the values $1,2,\ldots,K$
- We want to learn the parameters $\theta_1, \theta_2, \ldots, \theta_K$

Sufficient statistics:

- $N_1, N_2, \ldots, N_K$ - the number of times each outcome is observed

Likelihood function:

$$L(\theta : D) = \prod_{k=1}^{K} \theta_k^{N_k}$$

MLE:

$$\hat{\theta}_k = \frac{N_k}{\sum_{\ell} N_\ell}$$

From Beta to Dirichlet Distribution

- Prior: Dirichlet distribution

$$p(\theta) = Dir(\theta|\alpha_1, \cdots, \alpha_K)$$

$$\frac{\Gamma(\alpha_1 + \cdots + \alpha_K)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k}$$

- Posterior

$$p(\theta|D) = Dir(\theta|N_1 + \alpha_1, \cdots, N_K + \alpha_K)$$

$$\frac{\Gamma\left((N_1 + \alpha_1) + \cdots + (N_K + \alpha_K)\right)}{\prod_{k=1}^{K} \Gamma(N_k + \alpha_k)} \prod_{k=1}^{K} \theta_k^{N_k + \alpha_k}$$

From Beta to Dirichlet Distribution

- The MAP is

$$\theta_k = \frac{\alpha_k + N_k}{\sum_{i=1}^{K}(\alpha_i + N_i)}$$

- The marginal likelihood is

$$\begin{aligned}
P(D|G) &= \int P(D|\theta, G)P(\theta|G)d\theta \\
&= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\int_0^1 \prod_{k=1}^{K}\theta_k^{N_k+\alpha_k-1}d\theta_k \\
&= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\Gamma(\sum_{k=1}^{K}\alpha_k + \sum_{k=1}^{K}N_k)}\prod_{k=1}^{K}\frac{\Gamma(\alpha_k + N_k)}{\Gamma(\alpha_k)}
\end{aligned}$$

Likelihood for Multinomial Network

- When we assume that $P(X_i / Pa_i)$ is multinomial, we get further decomposition:

$$L_i(\Theta_i : D) = \prod_{m} P(x_i[m] \mid Pa_i[m] : \Theta_i)$$

$$= \prod_{pa_i} \prod_{m, Pa_i[m] = pa_i} P(x_i[m] \mid pa_i : \Theta_i)$$

$$= \prod_{pa_i} \prod_{x_i} P(x_i \mid pa_i : \Theta_i)^{N(x_i, pa_i)} = \prod_{pa_i} \prod_{x_i} \theta_{x_i \mid pa_i}{}^{N(x_i, pa_i)}$$

- For each value pa_i of the parents of X_i we get an independent multinomial problem
- The MLE is $\hat{\theta}_{x_i \mid pa_i} = \dfrac{N(x_i, pa_i)}{N(pa_i)}$

Bayesian Inference for Multinomial Network

- Given data, we can compute the posterior for each multinomial independently. The posteriors are also Dirichlet with parameters

$$\alpha(X_i=1|pa_i)+N\,(X_i=1|pa_i),\ldots,\ \alpha(X_i=k|pa_i)+N\,(X_i=k|pa_i)$$

- The predictive distribution is then represent by parameters

$$\tilde{\theta}_{x_j|pa_j} = \frac{\alpha(x_j,pa_j)+N(x_j,pa_j)}{\alpha(pa_j)+N(pa_j)}$$

More Generalizations

- Likelihood from exponential family.
  - Binomial distribution
  - Multinomial distribution
  - Poisson distribution
  - Gamma distribution
  - Normal distribution
- Conjugated distributions.

Learning Parameters: Summary

- Estimation relies on **sufficient statistics**
  - For multinomials: counts $N(x_i, pa_i)$
  - Parameter estimation

$$\hat{\theta}_{x_i|pa_i} = \frac{N(x_i, pa_i)}{N(pa_i)}$$

MLE

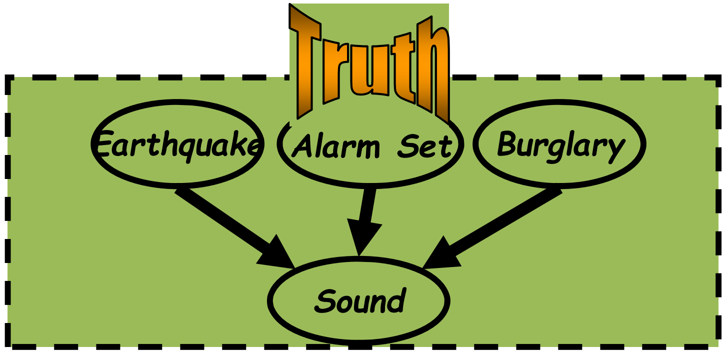
$$\tilde{\theta}_{x_i|pa_i} = \frac{\alpha(x_i, pa_i) + N(x_i, pa_i)}{\alpha(pa_i) + N(pa_i)}$$

Bayesian (Dirichlet)

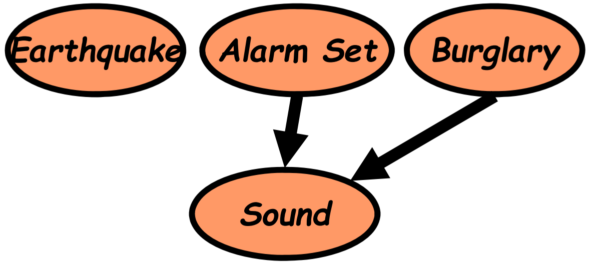
- Both are asymptotically equivalent.

B. Learning Structure From Data

Why Struggle for Accurate Structure?

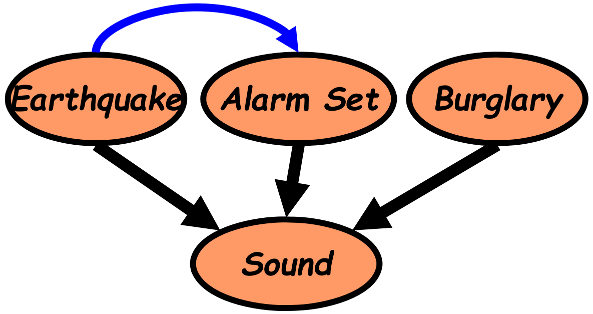


Missing an arc



- Cannot be compensated for by fitting parameters
- Wrong assumptions about domain structure

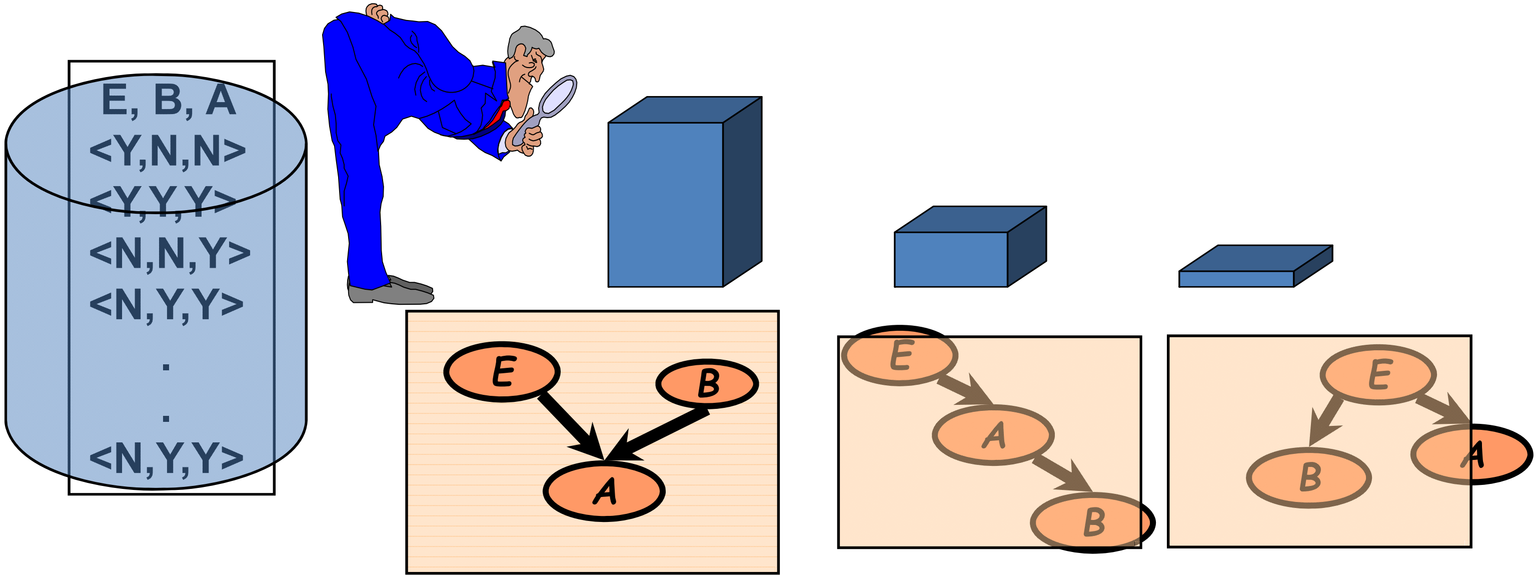
Adding an arc



- Increases the number of parameters to be estimated
- Wrong assumptions about domain structure

Scorebased Learning

Define scoring function that evaluates how well a structure matches the data



Search for a structure that maximizes the score

Score Function I

Which structure is good?

- BDe scores (Heckman)

$$BDe(G : D) = \underbrace{\log \int P(D \mid G, \Theta) P(\Theta \mid G) d\Theta}_{\text{Marginal likelihood}} + \underbrace{\log P(G)}_{\text{Structure Prior}}$$

Marginal Likelihood (Multinomial Case)

- If data are complete, we can obtain the close form.

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk} + \sum_{k=1}^{r_i} N_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

N_{ijk} : Number of cases where$X_i = k, Pa_{X_i} = j$

r_i : number of states ofX_i

q_i : number of instance of parents ofX_i.

Practical Consideration

Super exponential number (in the number of variables) of possible structures.

- How do we find the best graphs?
- How do we assign structure and parameter priors to all possible graphs?

Structure Prior Choice

- All possible structures are equally likely.
- Fix (or forbid) some arcs.
- Choosing a prior proportions to the similarity to a prior network structure.

Model Selection

- Theorem: finding the best BN structure among those structures with at most k parents in NP-hard problem (k>1).
- Heuristic searching.
  - Greedy.
  - MCMC.

Score Function II

Which structure is good?

- BIC/MDL scores
  - BIC: Bayesian Information Criterion.
  - MDL: Minimum Description Length.

$$\mathrm{BIC}(G, \Theta : D) = \log P(D \mid G, \Theta) - \frac{\log N}{2} \# \text{ param in } G$$

Fitness to data

Complexity regularization

Minimum Description Length Principle

- Universal coding.
  - Description length of the compressed form (model) of data.
  - Description length of the model itself used in the compression.

Minimum Description Length Principle (Cont.)

- Bayesian network case.
  - Modeling of data (Probability distribution).
  - Network coding (number of parameters).

See: N.Friedman. Learning Bayesian networks with local structure.

Decomposability

- Key property of the Bayesian network with complete data.

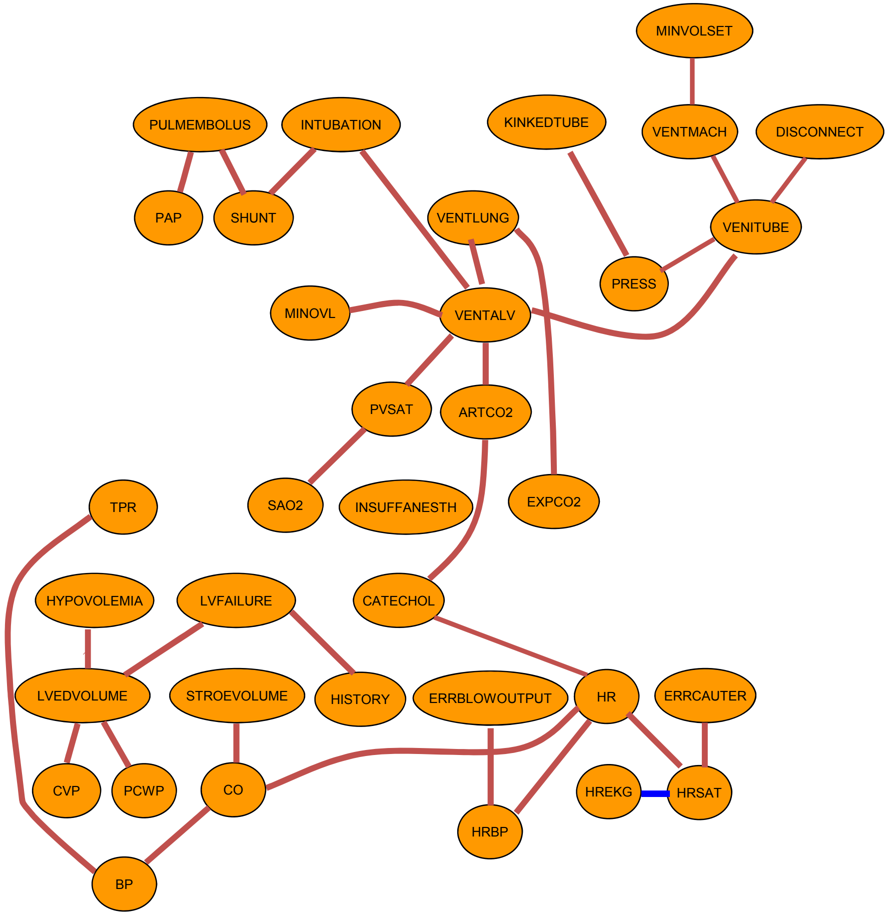
$$\text{score}(G) = \sum \text{score (family of X in G)}$$

Tree-structured Networks

Trees: At most one parent per variable.

Why trees?

- Elegant math=>we can solve the optimization problem
- Sparse parameterization to avoid over-fitting



Learning Trees

- Let *p(i)* denote parent of X_i
- The Bayesian score can be written as sum of edge scores.

$$
\begin{aligned}
Score(G:D) &= \sum_i Score(X_i : Pa_i) \\
&= \sum_i \left(Score(X_i : X_{p(i)}) - Score(X_i) \right) + \sum_i Score(X_i)
\end{aligned}
$$

Improvement over "empty" network

Score of "empty" network

Learning Tree

- Set edge weight as: $Score(X_j \rightarrow X_i) - Score(X_i)$.
- Well studied Problem in graph theory: Find the tree with maximum weight. It can be solved by maximum spanning tree algorithm (MST) in an efficient way.

Kruskal’s Algorithm on MST

```
begin Kruskal;
    sort the arcs in $A$ in decreasing order of their weights;
    LIST = $\varnothing$;
    while $|\text{LIST}| < n - 1$ do
    begin
        if the next arc does not create a cycle then add it to LIST
        else discard it
    end;
end;
```

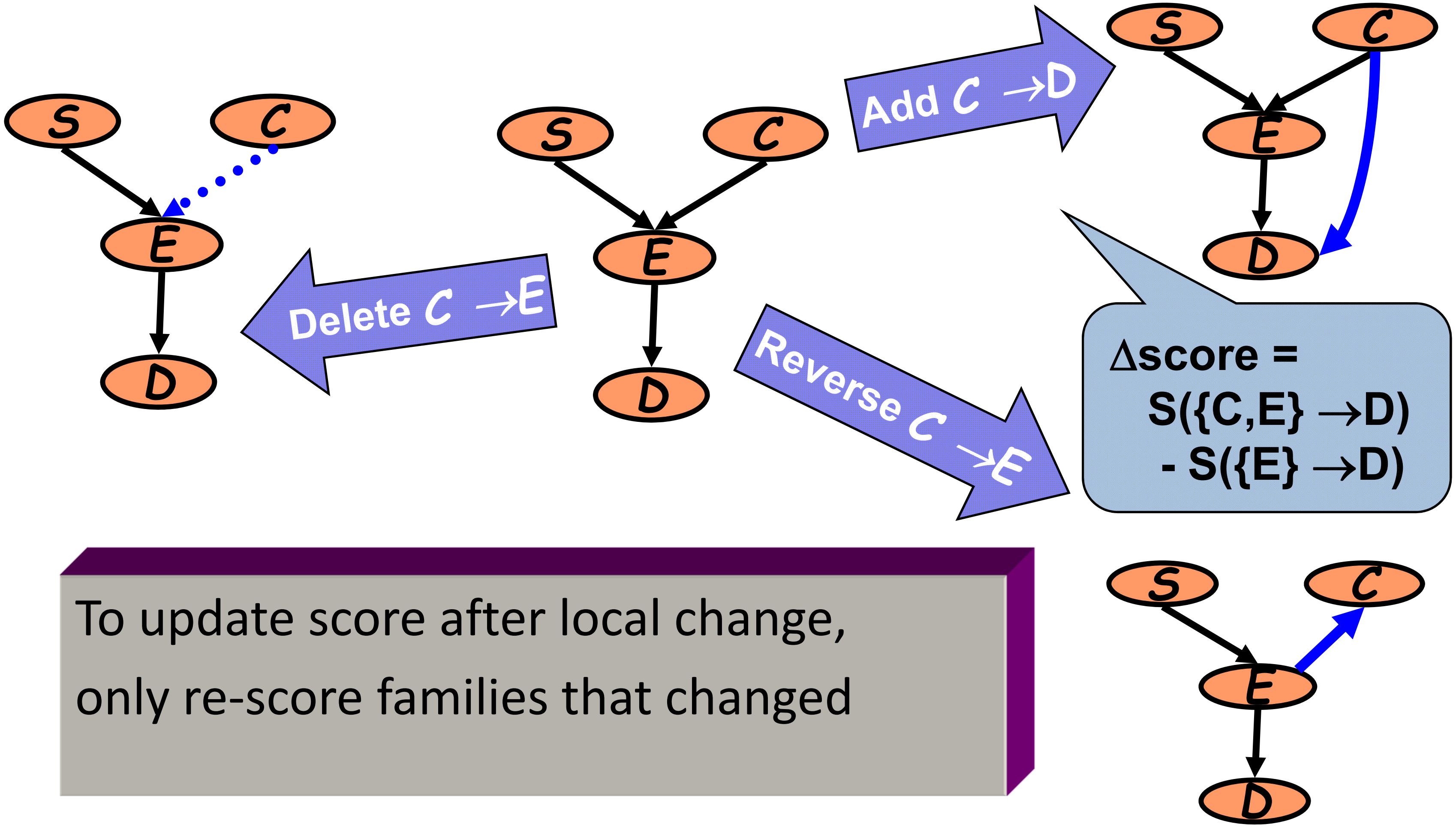
Heuristic Search: Beyond Trees

- Define a search space:
  - search states are possible structures
  - operators make small changes to structure
- Search techniques:
  - Greedy hill-climbing
  - Best first search
  - Simulated Annealing
  - ...

Local Search

- Start with a given network
  - empty network
  - best tree
  - a random network
- At each iteration
  - Evaluate all possible changes
  - Apply change based on score
- Stop when no modification improves score

Typical Operations In Heuristic Search



Δscore =
S({C,E} →D)
- S({E} →D)

To update score after local change,
only re-score families that changed

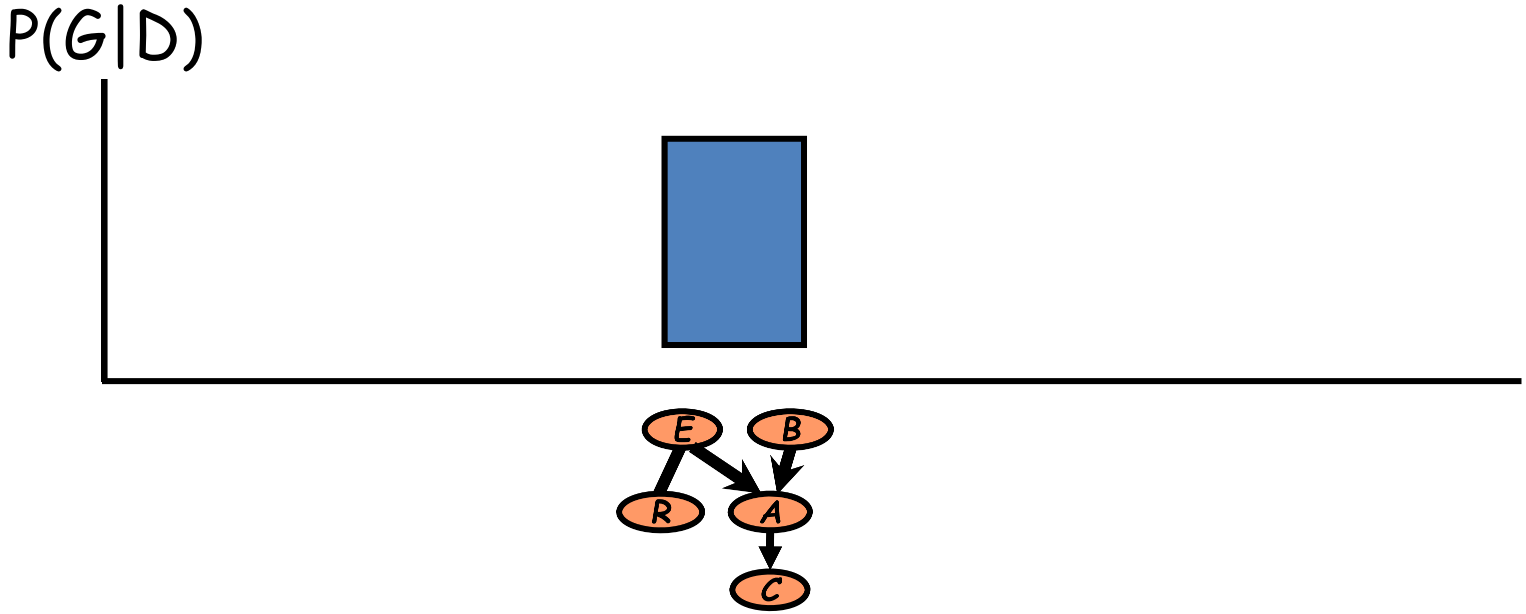
Local Search: Possible Pitfalls

- Local search can get stuck in:
  - **Local Maxima:**
    - All one-edge changes reduce the score
  - **Plateaus:**
    - Some one-edge changes leave the score unchanged

Escape From Traps

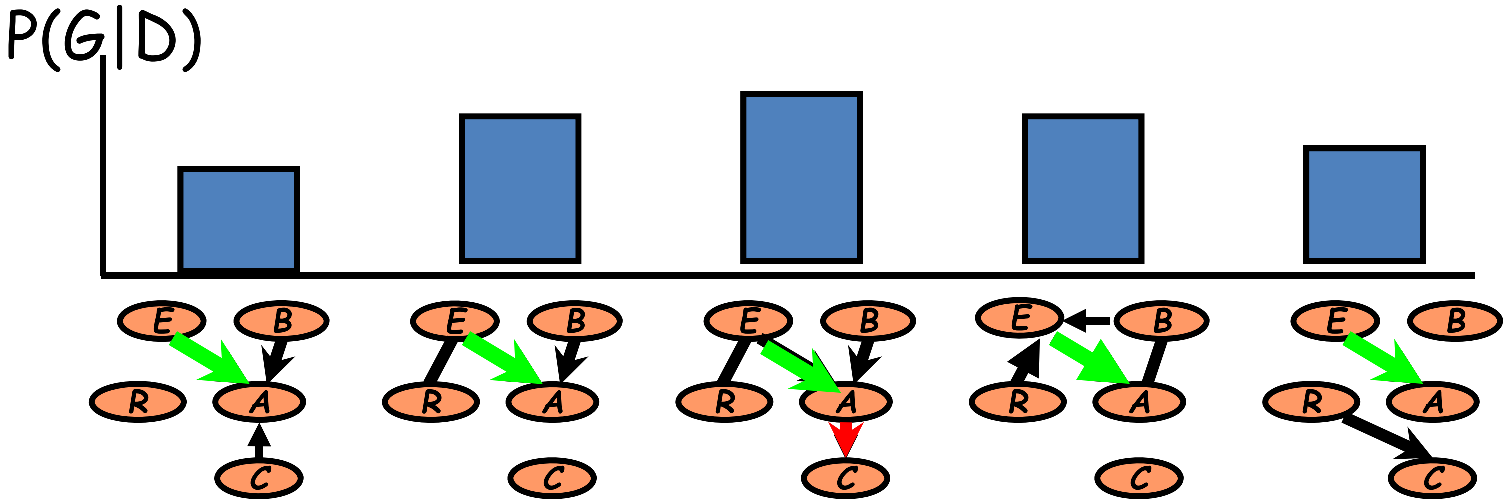
- Random restarts.
- Simulated annealing
  - Take the bad score with probability proportion to exp(Δscore/t).
  - Cool down slowly.

Discovering Structure



- Current practice: model selection
  - Pick a single high-scoring model
  - Use that model to infer domain structure

Discovering Structure



Problem

- Small sample size ⇒ many high scoring models
- Answer based on one model often useless.
- We want features common to many models.

Bayesian Approach

- Posterior distribution over structures
- Estimate probability of **features**
  - Edge X→Y
  - Path X→… → Y
  - …

Bayesian score for G

$$P(f \mid D) = \sum_{G} f(G) P(G \mid D)$$

Feature of *G*, e.g., *X*→*Y*

Indicator function for feature *f*

Practical Implementation

- Bootstrap method.
  - Randomly generate m “perturbed” sample sets.
  - For each sample set, choose a best model G_i.
  - Average the feature among these m structures.

$$P(f(G)\,|\,D) \approx \frac{1}{n}\sum_{i=1}^{n} f(G_i)$$

C: Dealing With Missing Data

- Structure known, how to learn the parameters?

- Structure unknown, how to learn the structure and parameters?

Incomplete Data

Data is often **incomplete**

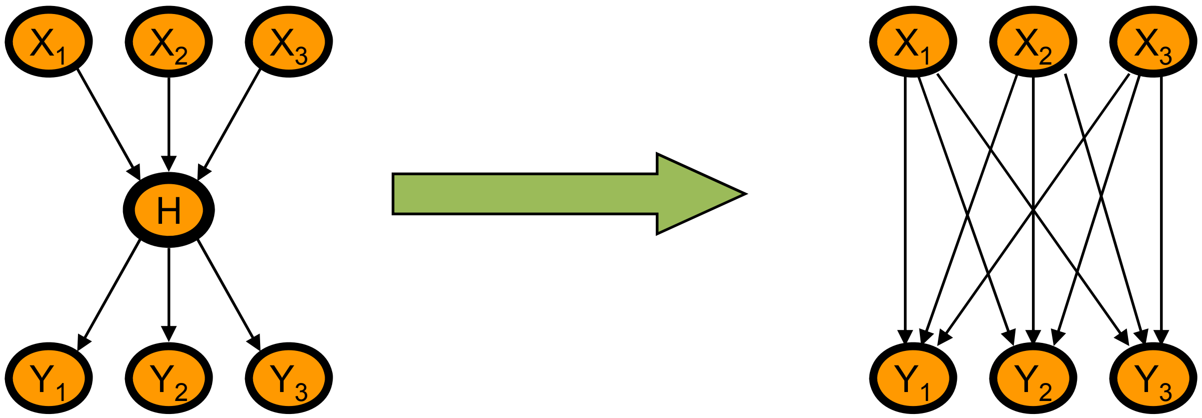
- Some variables of interest are not assigned values.

This phenomenon happens when we have

- **Missing values:**
  - Some variables unobserved in some instances
- **Hidden variables:**
  - Some variables are never observed
  - We might not even know they exist

Hidden (Latent) Variables

- Why should we care about unobserved variables?



17 parameters
17=1+1+1+8+2+2+2

27 parameters
27=1+1+1+8+8+8

More Computation

- The likelihood of the data does **not** decompose.
- Complete data.

$$\log L(\Theta : D = (x_1, \ldots, x_n)) = \sum_i \log P(x_i \mid \mathrm{Pa}(x_i))$$

- Incomplete data.

$$\log L(\Theta : D = (x_1, \ldots, x_k)) = \log \sum_{x_{k+1}, \ldots, x_n} \prod_i P(x_i \mid \mathrm{Pa}(x_i))$$

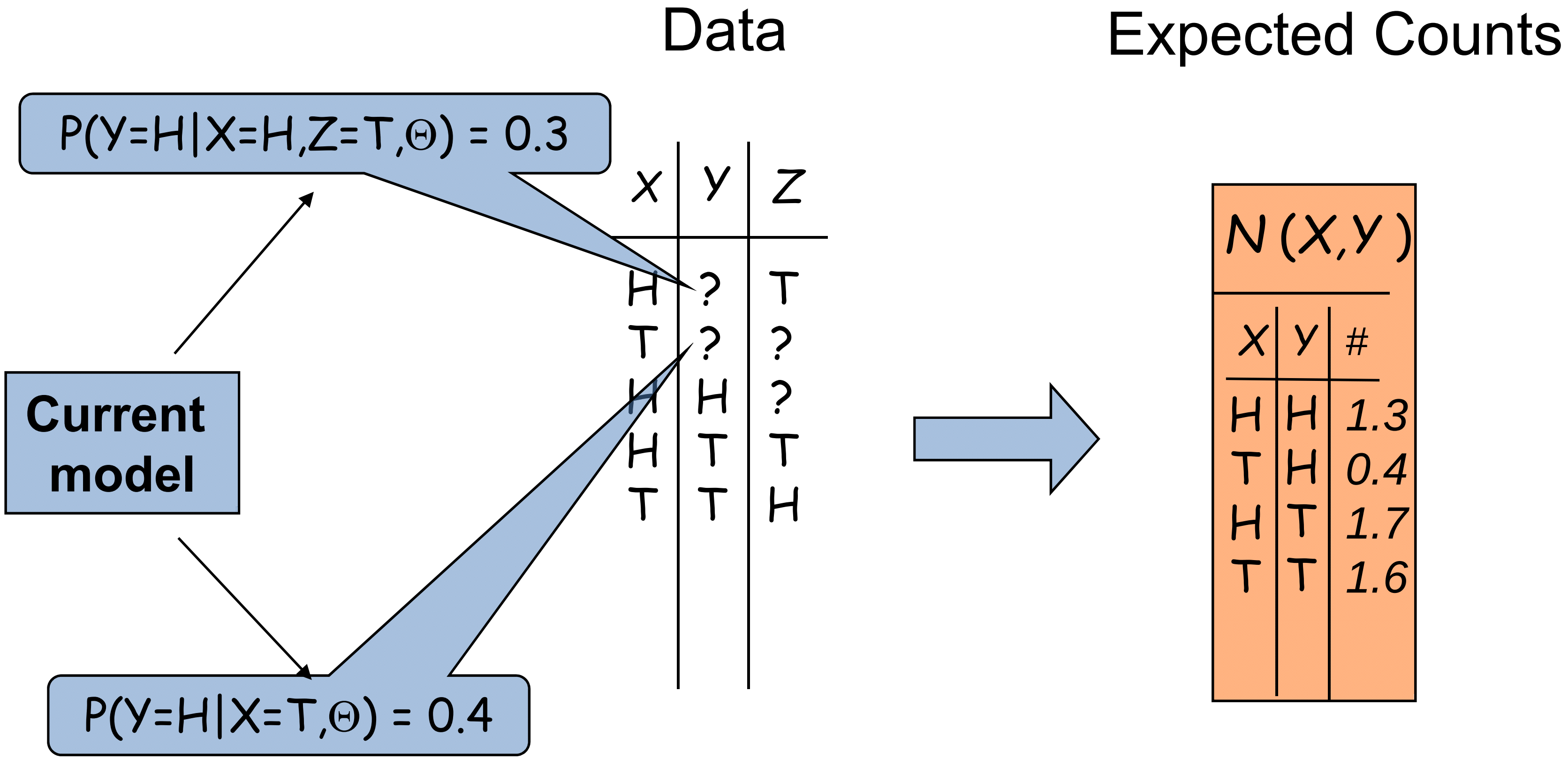
Learning Parameters With Incomplete Data

- Expectation maximization (EM) iteration algorithm is the general purpose method for learning from incomplete data.
  - E-Step.
  - M-Step.

EM Intuition

- If we had true counts, we could estimate parameters.
- But with missing values, counts are unknown.
- We “complete” counts using probabilistic inference based on current parameter assignment.
- We use completed counts as if real to re-estimate parameters.

EM Algorithm



Current model

$P(Y=H|X=H,Z=T,\Theta) = 0.3$

$P(Y=H|X=T,\Theta) = 0.4$

Data

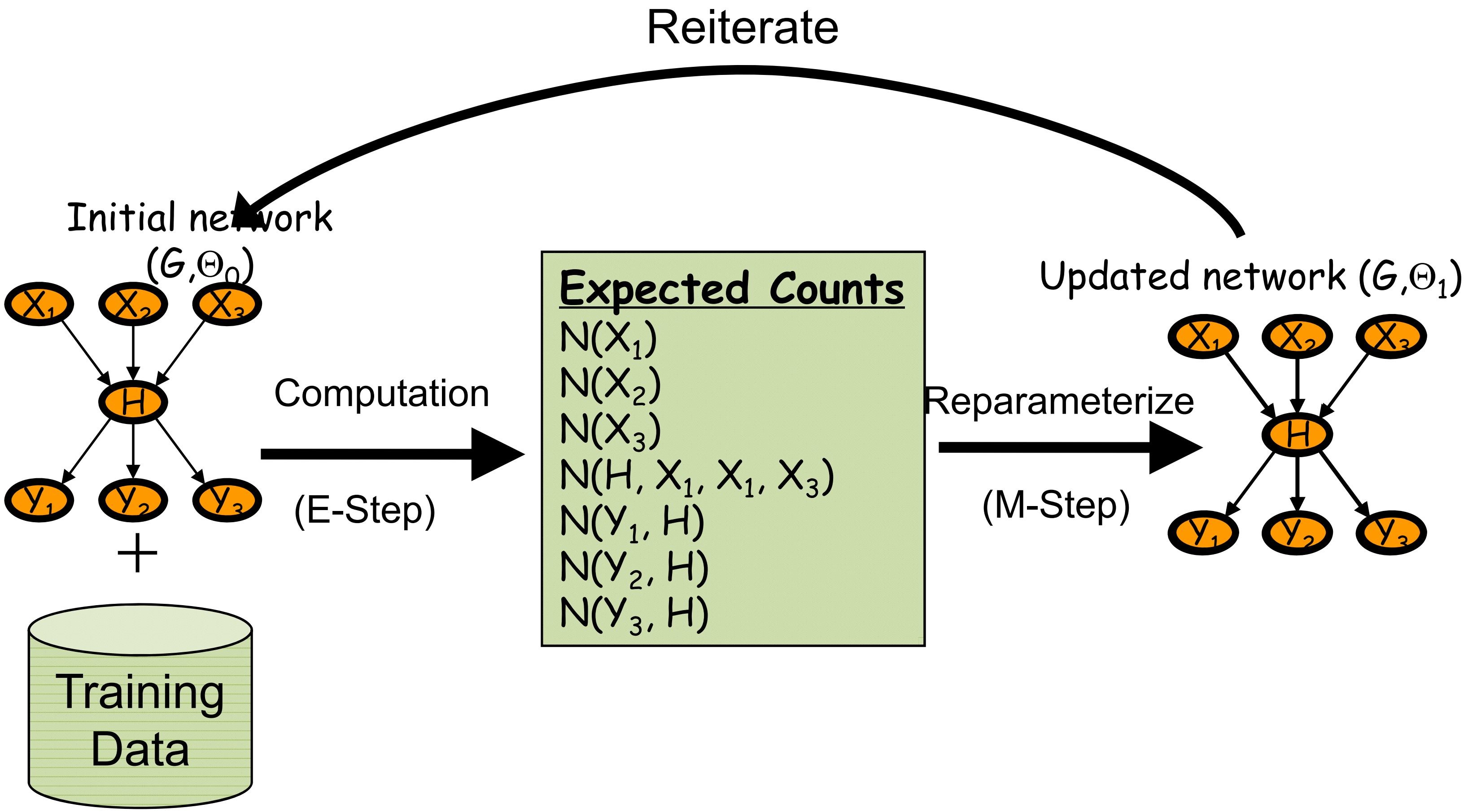
X	Y	Z
H	?	T
T	?	?
H	H	?
H	T	T
T	T	H

Expected Counts

$N(X,Y)$

X	Y	#
H	H	*1.3*
T	H	*0.4*
H	T	*1.7*
T	T	*1.6*

EM Algorithm (Cont.)



Reiterate

Initial network (G, Θ_0)

Training Data

Computation

(E-Step)

Expected Counts

$N(X_1)$

$N(X_2)$

$N(X_3)$

$N(H, X_1, X_1, X_3)$

$N(Y_1, H)$

$N(Y_2, H)$

$N(Y_3, H)$

Reparameterize

(M-Step)

Updated network (G, Θ_1)

EM Algorithm (Cont.)

Formal Guarantees:

- $L(\Theta_1{:}D) \geq L(\Theta_0{:}D)$
  - Each iteration improves the likelihood
- If $\Theta_1 = \Theta_0$, then Θ_0 is a **stationary point** of $L(\Theta{:}D)$
  - Usually, this means a local maximum

Computational Bottleneck

Computation of expected counts in E-Step

- Need to compute posterior for each unobserved variable in each instance of training set.
- All posteriors for an instance can be derived from one pass of standard BN inference.

Summary: Parameter Learning With Incomplete Data

- Incomplete data makes parameter estimation hard
- Likelihood function
  - Does not have closed form
  - Is multimodal
- Finding maximum likelihood parameters:
  - EM
  - Gradient ascent
- Both exploit inference procedures for Bayesian networks to compute expected sufficient statistics

Incomplete Data: Structure Scores

Recall, Bayesian score:

$$
\begin{aligned}
P(G \mid D) &\propto P(G)P(D \mid G) \\
&= P(G)\boxed{\int P(D \mid G, \Theta)P(\Theta \mid G)d\theta}
\end{aligned}
$$

With incompleté data:

- Cannot evaluate marginal likelihood in closed form.
- We have to resort to **approximations**:
  - Evaluate score around MAP parameters
  - Need to find MAP parameters (e.g., EM)

Naïve Approach

- Perform EM for each candidate graph.
- Computationally expensive:
  - Parameter optimization via EM — non-trivial
  - Need to perform EM for all candidate structures
  - Spend time even on poor candidates
- In practice, considers only a few candidates.

Structural EM

Recall, in complete data we had

–Decomposition $\Rightarrow$ efficient search.

Idea:

- Instead of optimizing the real score…
- Find **decomposable** alternative score.
- Such that maximizing new score $\Rightarrow$ improvement in real score.

Structural EM (Cont.)

Idea:

- Use current model to help evaluate new structures

Outline:

- Perform search in (Structure, Parameters) space.
- At each iteration, use current model for finding either:
  - Better scoring parameters: “parametric” EM step.
  - Better scoring structure: “structural” EM step.

Structural EM Steps

Assume $B_0 = (G_0, \Theta_0)$ is "current" hypothesis.

Goal: Maximize **expected score**, given B_0

$$E[Score(B : D^+) \mid D, B_0] = \sum_{D^+} Score(B : D^+) P(D^+ \mid D, B_0)$$

where D^+ denotes **completed** data sets.

Theorem:(progress)

If $E[Score(B : D^+) \mid D, B_0] > E[Score(B_0 : D^+) \mid D, B_0]$

$\Rightarrow Score(B : D) > Score(B_0 : D)$.

- This implies that by improving the expected score, we find networks that have higher objective score.

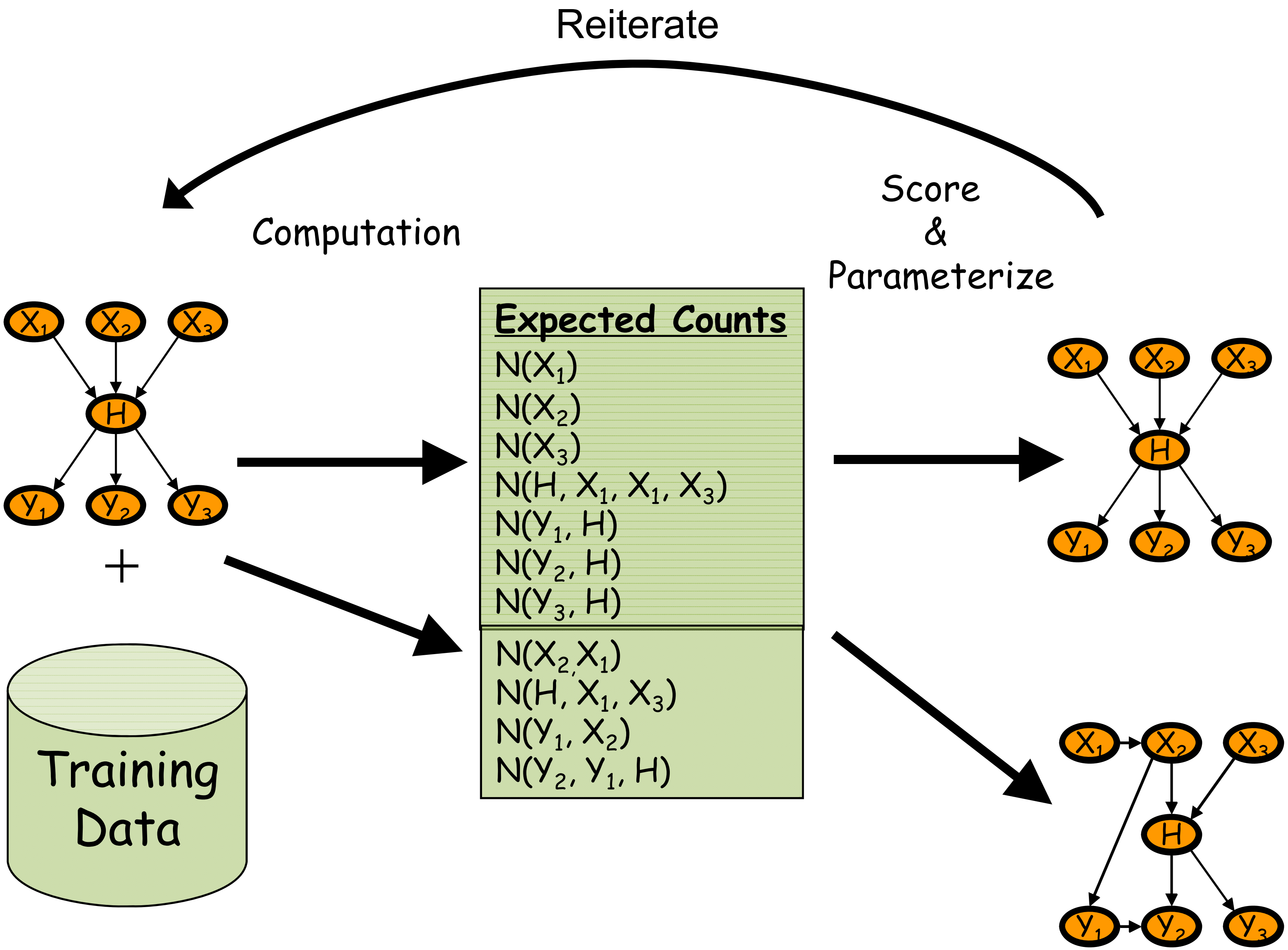
Structural EM for BIC/MDL

For the BIC/MDL score, we get that

$$
\begin{aligned}
&\mathrm{E}[\mathrm{BIC}(B : D^{+}) \mid D, B_0\,] \\
&\qquad = \mathrm{E}[\log \mathrm{P}(D^{+} \mid B) \mid D, B_0\,] \text{ - Penalty(B)} \\
&\qquad = E[\sum_i N(X_i, Pa_i) \log P(X_i \mid Pa_i) \mid D, B_0] - \mathrm{Penalty}(B) \\
&\qquad = \sum_i E[N(X_i, Pa_i) \mid D, B_0] \log P(X_i \mid Pa_i) - \mathrm{Penalty}(B)
\end{aligned}
$$

Consequence:

- We can use complete-data methods, where we use expected counts, instead of actual counts.



Reiterate

Computation

Score
&
Parameterize

Expected Counts

$N(X_1)$
$N(X_2)$
$N(X_3)$
$N(H, X_1, X_1, X_3)$
$N(Y_1, H)$
$N(Y_2, H)$
$N(Y_3, H)$

$N(X_{2,} X_1)$
$N(H, X_1, X_3)$
$N(Y_1, X_2)$
$N(Y_2, Y_1, H)$

Training Data

The Structural EM Procedure

```
Input: $B_0 = (G_0,\Theta_0)$
    loop for $n = 0,1,\ldots$ until convergence
    Improve parameters:
        $\Theta`_n =$ Parametric-EM $(G_n,\Theta_n)$
        let $B`_n = (G_{n`},\Theta`_n)$
    Improve structure:
        Search for a network $B_{n+1} = (G_{n+1},\Theta_{n+1})$ s.t.
        $E[Score(B_{n+1}:D) \mid B`_n] > E[Score(B`_n:D) \mid B`_n]$
```

- Parametric-EM() can be replaced by Gradient Ascent, Newton-Raphson methods, or accelerated EM.
- Early stopping parameter optimization stage avoids “entrenchment” in current structure.

App1: Expression Data Analysis

Reference:

- N.Friedman et al. Using Bayesian Networks to analyze expression data. *J. Comput. Biol.,* **7**:601-620, 2000.
- A.Hartemink et al. Combining location and expression data for principled discovery of genetic regulatory network models. PSB 2002.

Motivation

- Extract meaningful information from gene expression data.
  - Infer regulatory mechanism.
  - Reveal function of proteins.
  - ……

Case 1: Cell-cycle Data

- Yeast cell-cycle data (P.Spellman, *Mol. Biol. of the cell*, 1998).
- 7 time series under different cell cycle synchronization methods (alpha, beta factor, CDC15, CDC24, CDC28,cln2,3).
- 6177 ORFs, 77 time points.
- 800 genes are identified related to cell cycle process (big variation).

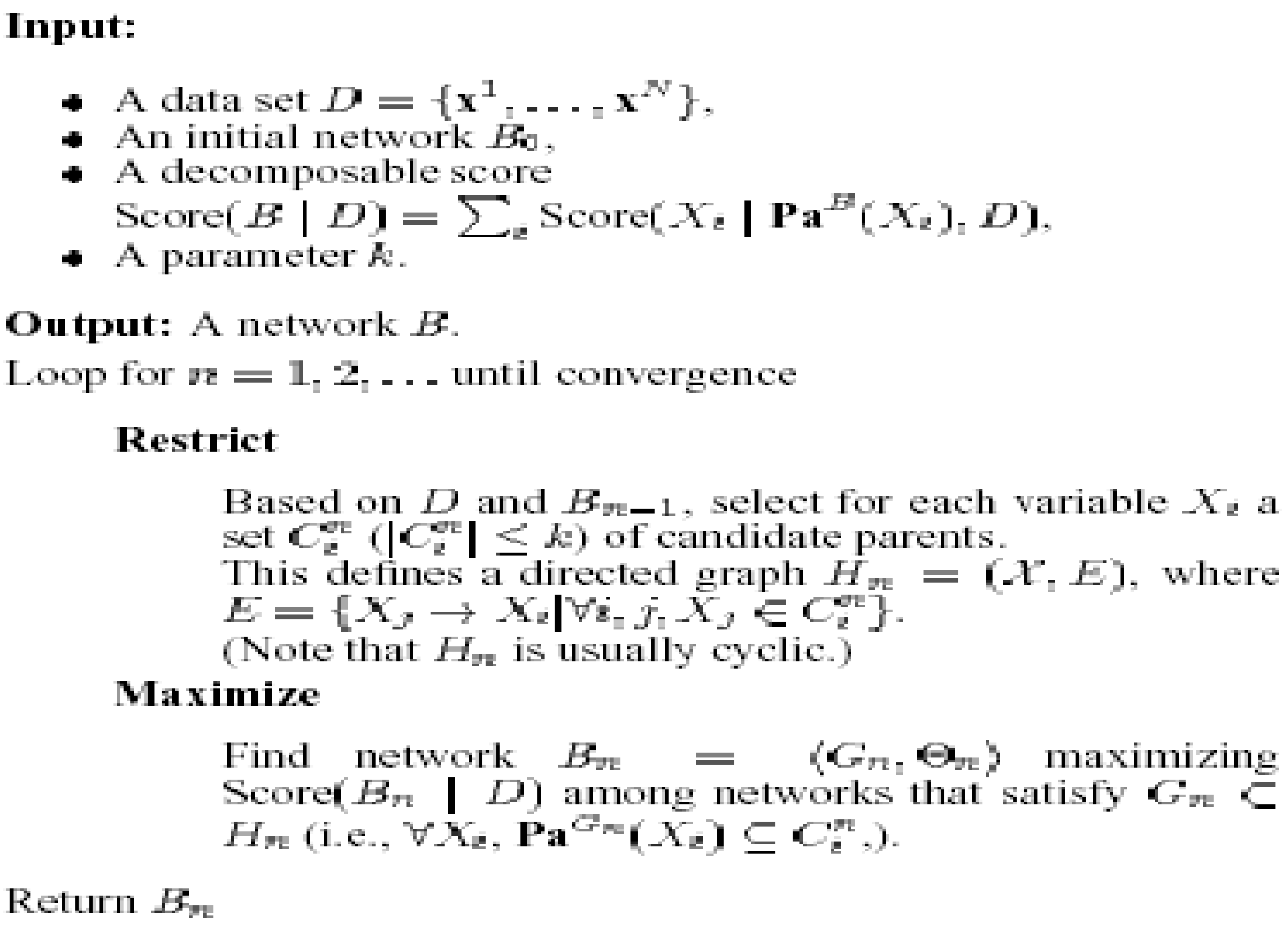
Bayesian Network Model

- Random Variables
  - Individual genes
  - Experimental condition
  - Cell phase.
- Discretization: 3 levels, -1,0,1, depending on whether the expression level is significantly lower than, similar to, great than the respective control. However, this may not be necessary (For continuous variable, a linear Gaussian conditional model can be used).

Learning Bayesian Network (Cont.)

- Sparse candidate algorithm: identify small number of candidate parents for each gene based on simple local statistics (such as mutual information).
- Bootstrap confidence estimation:
  - Use re-sampling to generate perturbations of training data.
  - Use the number of times of feature is repeated among networks from these datasets to estimate confidence of Bayesian network features.

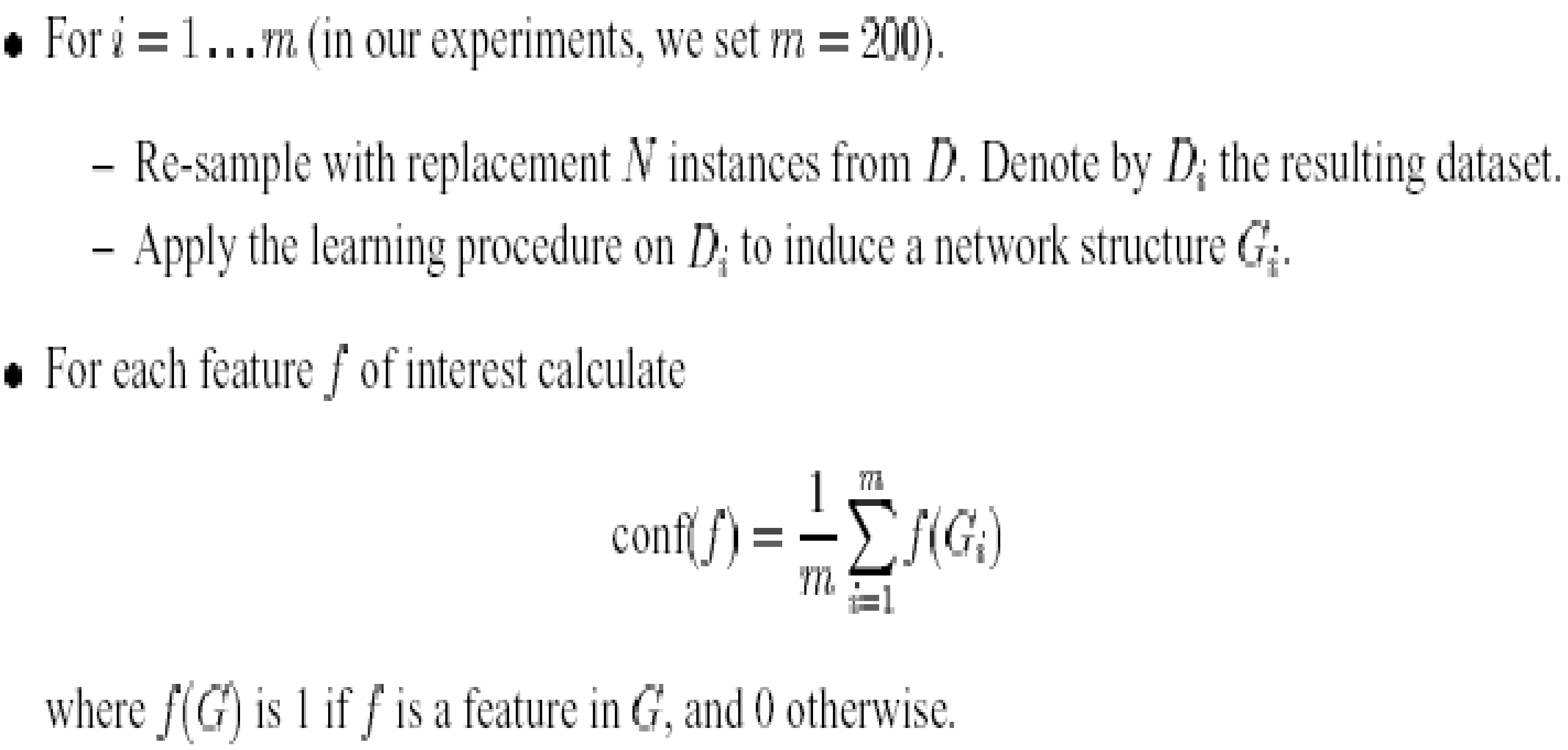
Sparse Candidate Algorithm



```
Input:
    • A data set $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$,
    • An initial network $B_0$,
    • A decomposable score
      $\text{Score}(B \mid D) = \sum_i \text{Score}(X_i \mid \mathbf{Pa}^B(X_i), D)$,
    • A parameter $k$.
Output: A network $B$.
Loop for $n = 1, 2, \ldots$ until convergence
    Restrict
        Based on $D$ and $B_{n-1}$, select for each variable $X_i$ a
        set $C_i^n$ ($|C_i^n| \le k$) of candidate parents.
        This defines a directed graph $H_n = (\mathcal{X}, E)$, where
        $E = \{X_j \rightarrow X_i | \forall i, j, X_j \in C_i^n\}$.
        (Note that $H_n$ is usually cyclic.)
    Maximize
        Find network $B_n = \langle G_n, \Theta_n \rangle$ maximizing
        $\text{Score}(B_n \mid D)$ among networks that satisfy $G_n \subset$
        $H_n$ (i.e., $\forall X_i$, $\mathbf{Pa}^{G_n}(X_i) \subseteq C_i^n$.).
Return $B_n$
```

Figure 1: Outline of the *Sparse Candidate* algorithm

Estimate Feature Significance Bootstrap Method



- For $i = 1 \ldots m$ (in our experiments, we set $m = 200$).
  - Re-sample with replacement N instances from D. Denote by D_i the resulting dataset.
  - Apply the learning procedure on D_i to induce a network structure G_i.
- For each feature f of interest calculate

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

where $f(G)$ is 1 if f is a feature in G, and 0 otherwise.

Markov Relation

- Pairs with 80% confidence were evaluated against original clustering.
  - 70% of these were intra-cluster.
  - The rest show interesting inter-cluster relations.
- Most pairs are functionally related.

Markov Relation (Cont.)

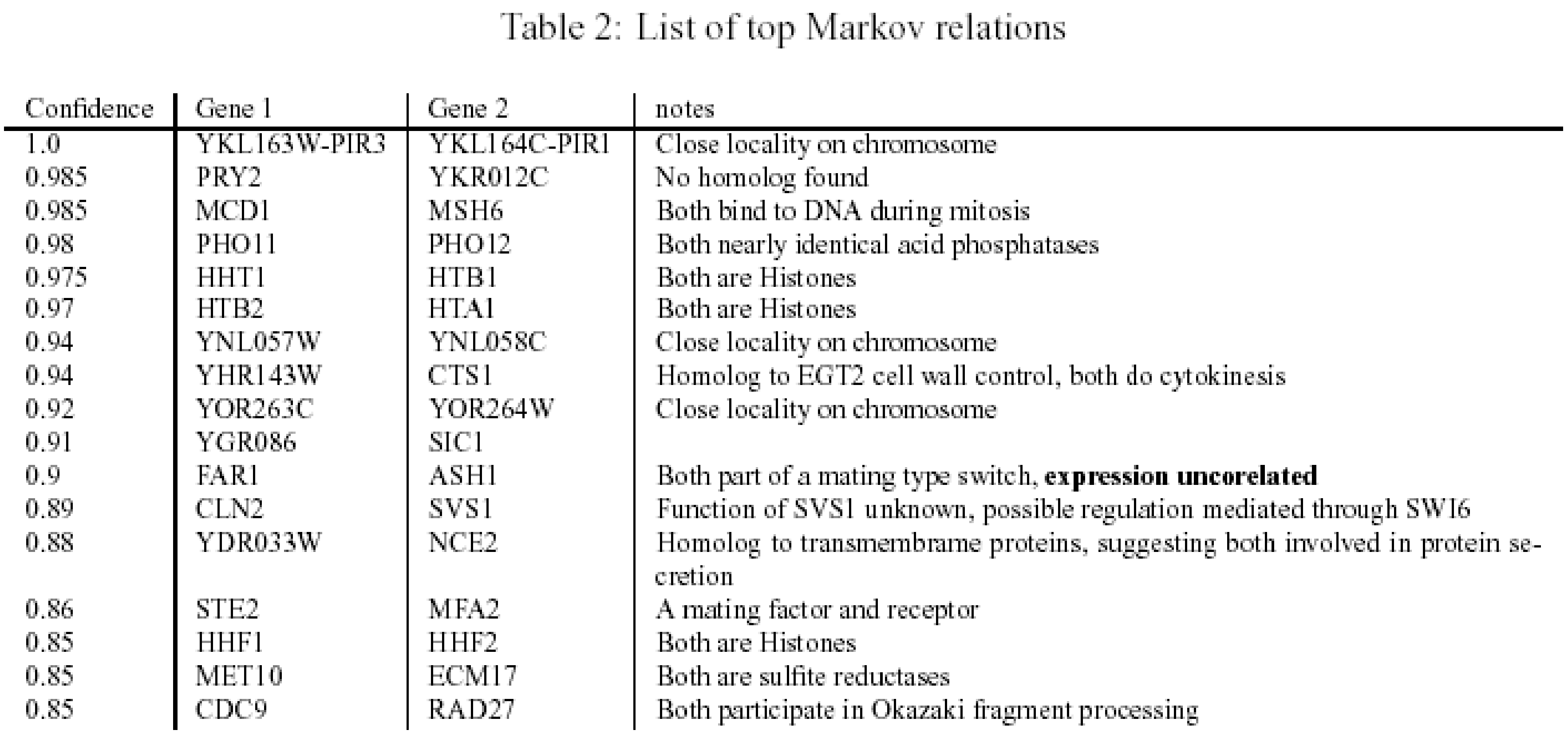


Table 2: List of top Markov relations

Confidence	Gene 1	Gene 2	notes
1.0	YKL163W-PIR3	YKL164C-PIR1	Close locality on chromosome
0.985	PRY2	YKR012C	No homolog found
0.985	MCD1	MSH6	Both bind to DNA during mitosis
0.98	PHO11	PHO12	Both nearly identical acid phosphatases
0.975	HHT1	HTB1	Both are Histones
0.97	HTB2	HTA1	Both are Histones
0.94	YNL057W	YNL058C	Close locality on chromosome
0.94	YHR143W	CTS1	Homolog to EGT2 cell wall control, both do cytokinesis
0.92	YOR263C	YOR264W	Close locality on chromosome
0.91	YGR086	SIC1	
0.9	FAR1	ASH1	Both part of a mating type switch, **expression uncorelated**
0.89	CLN2	SVS1	Function of SVS1 unknown, possible regulation mediated through SWI6
0.88	YDR033W	NCE2	Homolog to transmembrane proteins, suggesting both involved in protein secretion
0.86	STE2	MFA2	A mating factor and receptor
0.85	HHF1	HHF2	Both are Histones
0.85	MET10	ECM17	Both are sulfite reductases
0.85	CDC9	RAD27	Both participate in Okazaki fragment processing

Order Relation

- Dominant gene: genes are indicative or potential source of the cell-cycle process.
- Dominance score: describing how strong that one gene can be the ancestor of other genes in the network.

Dominant Genes

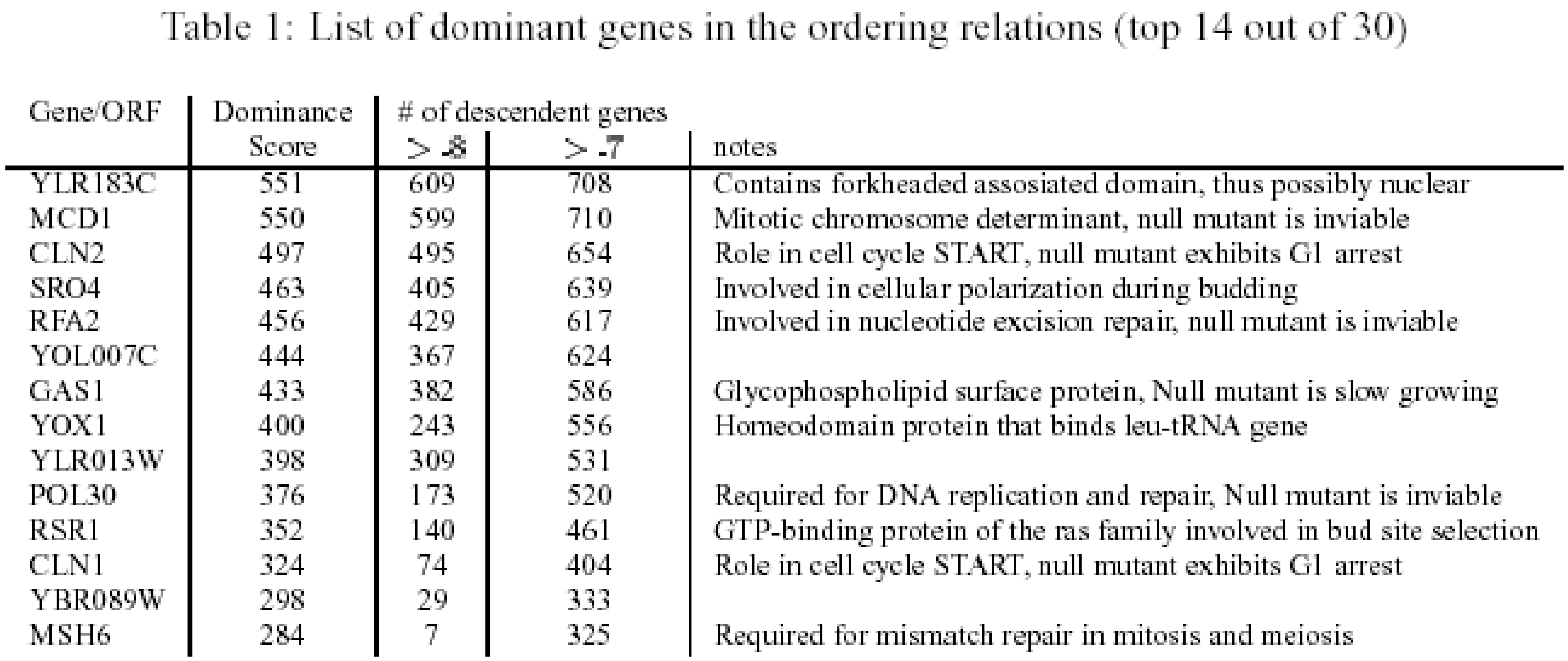


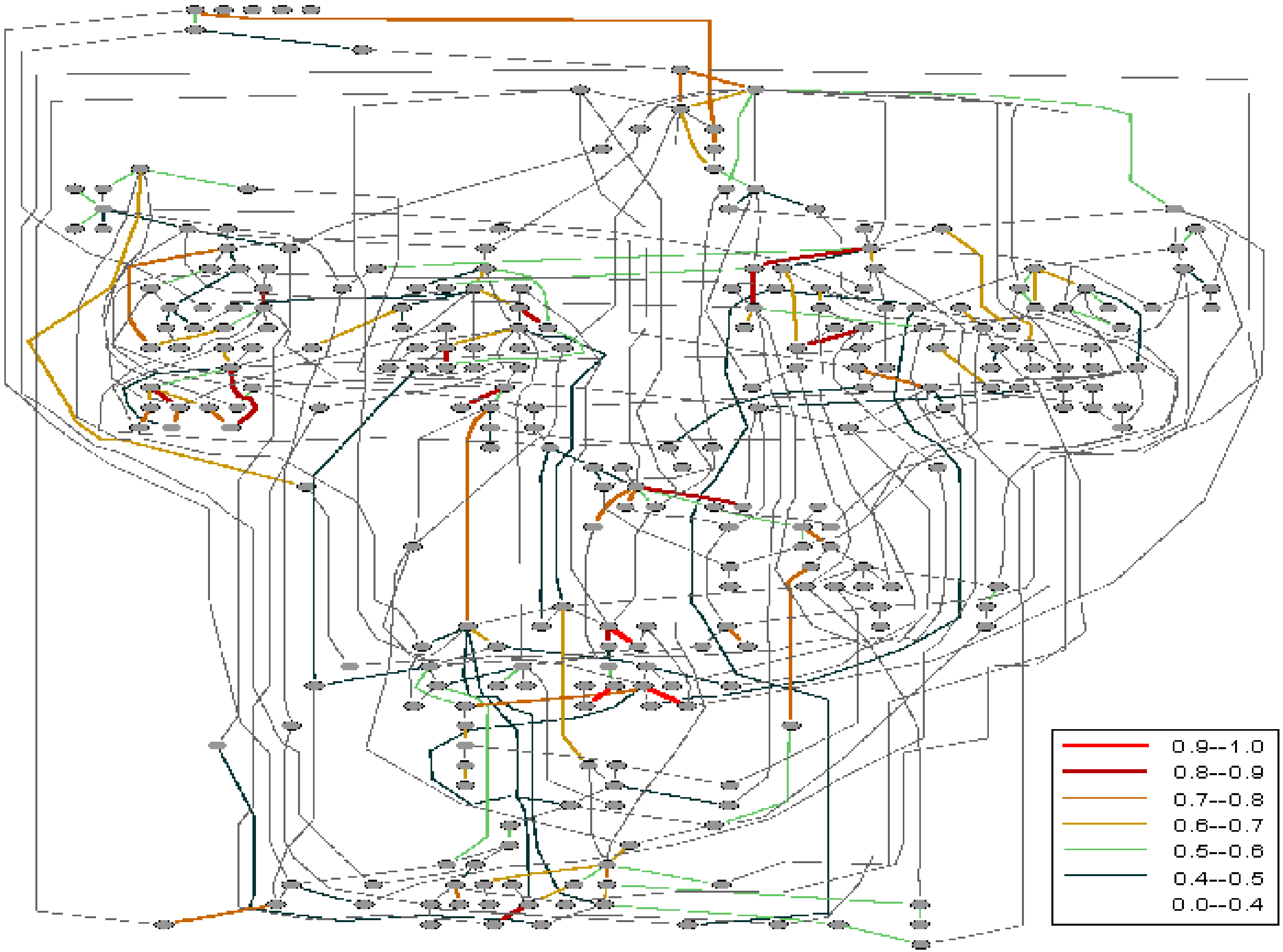
Table 1: List of dominant genes in the ordering relations (top 14 out of 30)

Gene/ORF	Dominance	# of descendent genes		
	Score	> .8	> .7	notes
YLR183C	551	609	708	Contains forkheaded assosiated domain, thus possibly nuclear
MCD1	550	599	710	Mitotic chromosome determinant, null mutant is inviable
CLN2	497	495	654	Role in cell cycle START, null mutant exhibits G1 arrest
SRO4	463	405	639	Involved in cellular polarization during budding
RFA2	456	429	617	Involved in nucleotide excision repair, null mutant is inviable
YOL007C	444	367	624	
GAS1	433	382	586	Glycophospholipid surface protein, Null mutant is slow growing
YOX1	400	243	556	Homeodomain protein that binds leu-tRNA gene
YLR013W	398	309	531	
POL30	376	173	520	Required for DNA replication and repair, Null mutant is inviable
RSR1	352	140	461	GTP-binding protein of the ras family involved in bud site selection
CLN1	324	74	404	Role in cell cycle START, null mutant exhibits G1 arrest
YBR089W	298	29	333	
MSH6	284	7	325	Required for mismatch repair in mitosis and meiosis

Cell cycle control and initiation: CLN1, CLN2, CDC5.

……

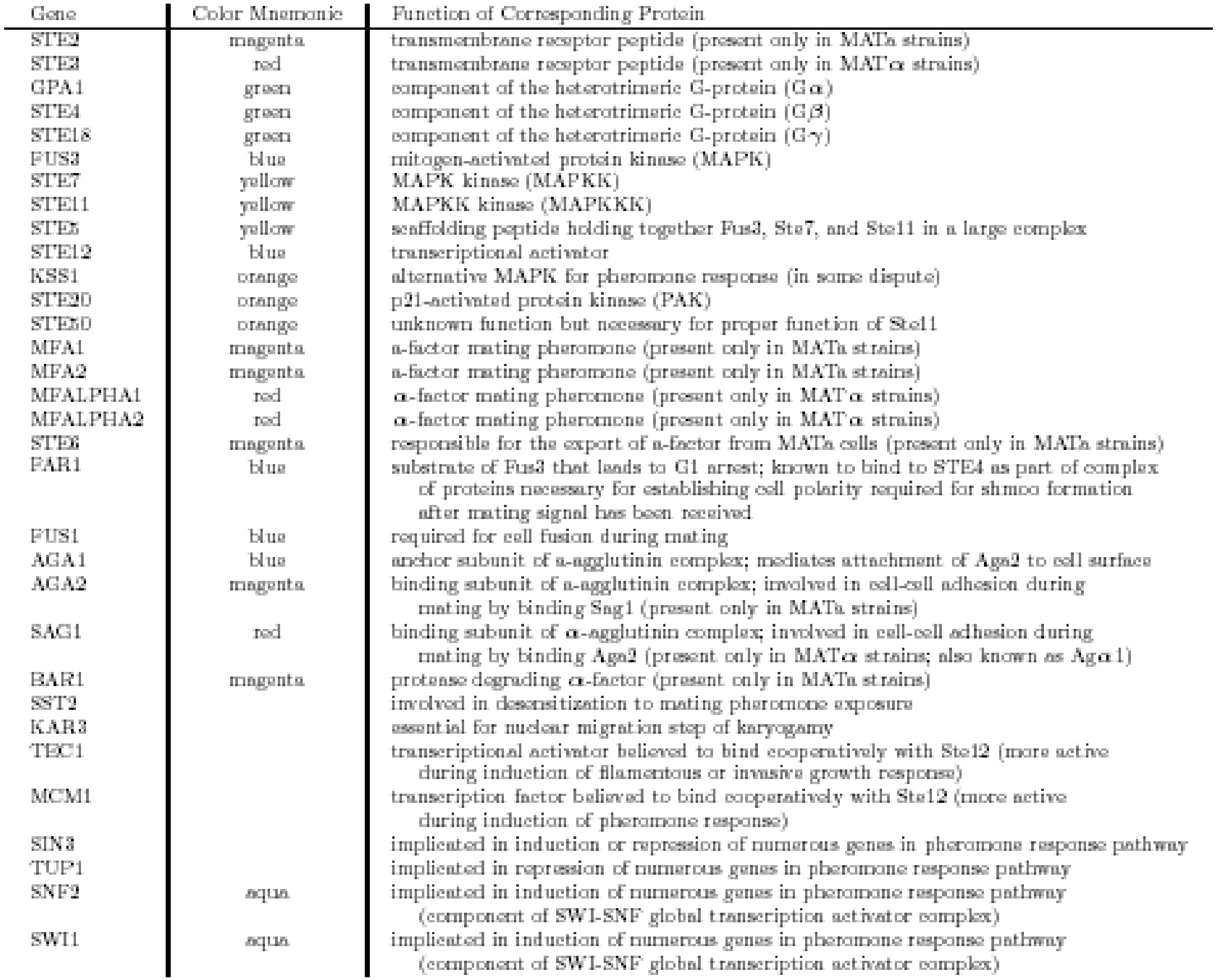
Network Learned



Case 2: Pheromone and Mating Response

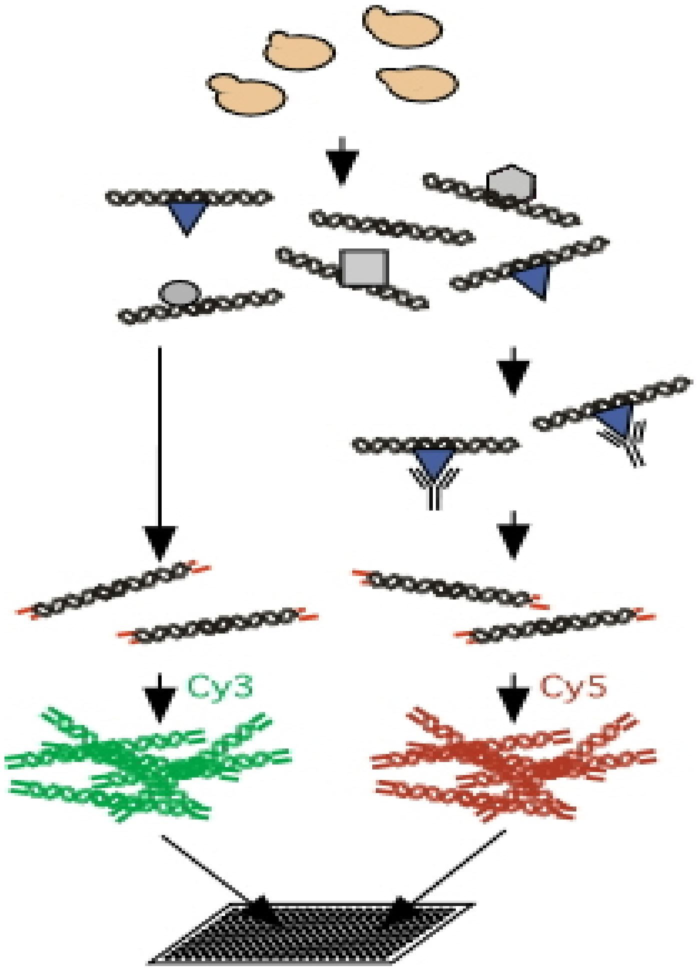
- 6135 genes, 320 samples under different conditions.
- 32 genes are selected.
  - Pheromone response signaling pathway.
  - Mating response.
- Location data (transcription factor and DNA binding experiment, chip-chip data) are included as prior constraints.

Genes Selected



Gene	Color Mnemonic	Function of Corresponding Protein
STE2	magenta	transmembrane receptor peptide (present only in MATa strains)
STE3	red	transmembrane receptor peptide (present only in MATα strains)
GPA1	green	component of the heterotrimeric G-protein (Gα)
STE4	green	component of the heterotrimeric G-protein (Gβ)
STE18	green	component of the heterotrimeric G-protein (Gγ)
FUS3	blue	mitogen-activated protein kinase (MAPK)
STE7	yellow	MAPK kinase (MAPKK)
STE11	yellow	MAPKK kinase (MAPKKK)
STE5	yellow	scaffolding peptide holding together Fus3, Ste7, and Ste11 in a large complex
STE12	blue	transcriptional activator
KSS1	orange	alternative MAPK for pheromone response (in some dispute)
STE20	orange	p21-activated protein kinase (PAK)
STE50	orange	unknown function but necessary for proper function of Ste11
MFA1	magenta	a-factor mating pheromone (present only in MATa strains)
MFA2	magenta	a-factor mating pheromone (present only in MATa strains)
MFALPHA1	red	α-factor mating pheromone (present only in MATα strains)
MFALPHA2	red	α-factor mating pheromone (present only in MATα strains)
STE6	magenta	responsible for the export of a-factor from MATa cells (present only in MATa strains)
FAR1	blue	substrate of Fus3 that leads to G1 arrest; known to bind to STE4 as part of complex of proteins necessary for establishing cell polarity required for shmoo formation after mating signal has been received
FUS1	blue	required for cell fusion during mating
AGA1	blue	anchor subunit of a-agglutinin complex; mediates attachment of Aga2 to cell surface
AGA2	magenta	binding subunit of a-agglutinin complex; involved in cell-cell adhesion during mating by binding Sag1 (present only in MATa strains)
SAG1	red	binding subunit of α-agglutinin complex; involved in cell-cell adhesion during mating by binding Aga2 (present only in MATα strains; also known as Agα1)
BAR1	magenta	protease degrading α-factor (present only in MATa strains)
SST2		involved in desensitization to mating pheromone exposure
KAR3		essential for nuclear migration step of karyogamy
TEC1		transcriptional activator believed to bind cooperatively with Ste12 (more active during induction of filamentous or invasive growth response)
MCM1		transcription factor believed to bind cooperatively with Ste12 (more active during induction of pheromone response)
SIN3		implicated in induction or repression of numerous genes in pheromone response pathway
TUP1		implicated in repression of numerous genes in pheromone response pathway
SNF2	aqua	implicated in induction of numerous genes in pheromone response pathway (component of SWI-SNF global transcription activator complex)
SWI1	aqua	implicated in induction of numerous genes in pheromone response pathway (component of SWI-SNF global transcription activator complex)

Location Analysis (Chip-chip)



- Crosslink protein to DNA in vivo with formaldehyde
- Break open cells and shear DNA
- Immunoprecipitate
- Reverse-crosslinks, blunt DNA and ligate to unidirectional linkers
- LM-PCR
- Hybridize to array

http://inside.wi.mit.edu/young/pub/locationanalysis.html

Bayesian Network Model

- Random variables
  - 32 genes.
  - Mating type (Mata, Matα).
- Discrimination: to 4 levels while preserving over 98% of the original total mutual information between pairs of genes.
- Location data: set the constraints specifying which edges are required to be present and which are required to be absent.

Learning Bayesian Network

- Score: Bayesian score metric (BSM).
- Local heuristic searching algorithm: simulated annealing.
- Caching: keeping the top 500 structures recorded.
- Feature induction: Average features within top 500 structures.

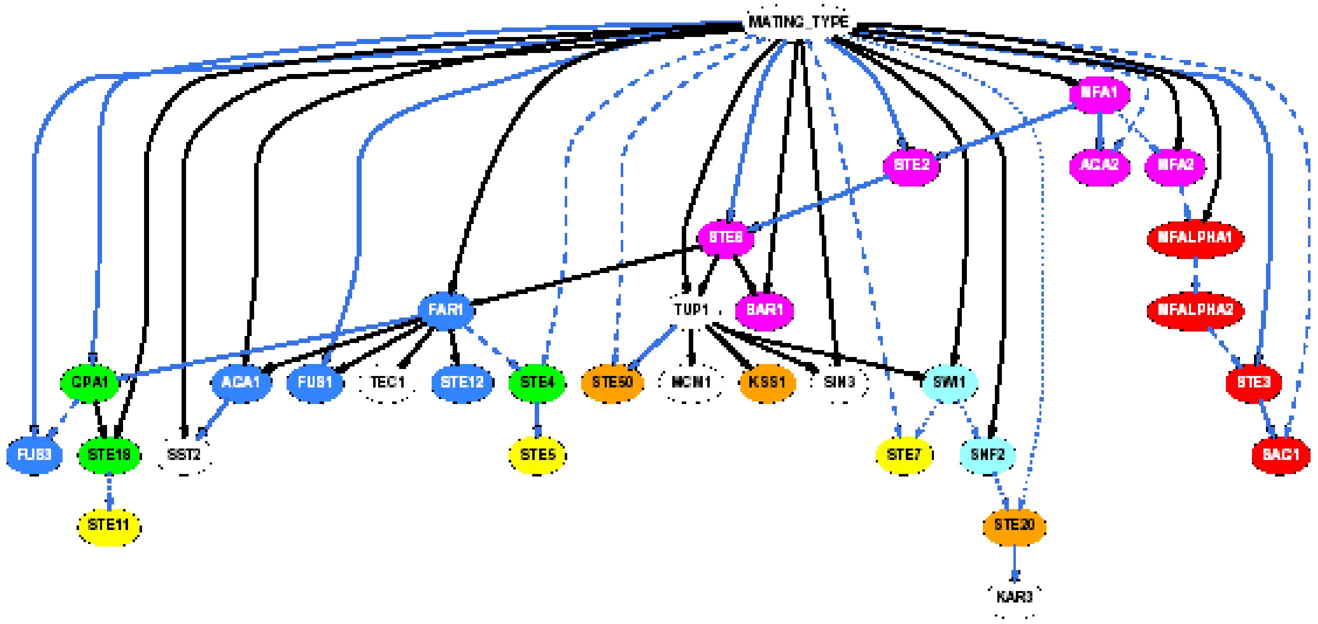
Learning Bayesian Network (Cont.)

$$\begin{aligned}\mathrm{p}(E_{XY}|D) &= \sum_{S} \mathrm{p}(E_{XY}|D,S)\cdot \mathrm{p}(S|D)\\ &= \sum_{S} 1_{XY}(S)\cdot \mathrm{e}^{\mathsf{BSM}(S)}\end{aligned}$$

Approximation:

$$\mathrm{p}(E_{XY}|D) \approx \frac{\sum_{i=1}^{N} 1_{XY}(S_i)\cdot \mathrm{e}^{\mathsf{BSM}(S_i)}}{\sum_{i=1}^{N} \mathrm{e}^{\mathsf{BSM}(S_i)}}$$

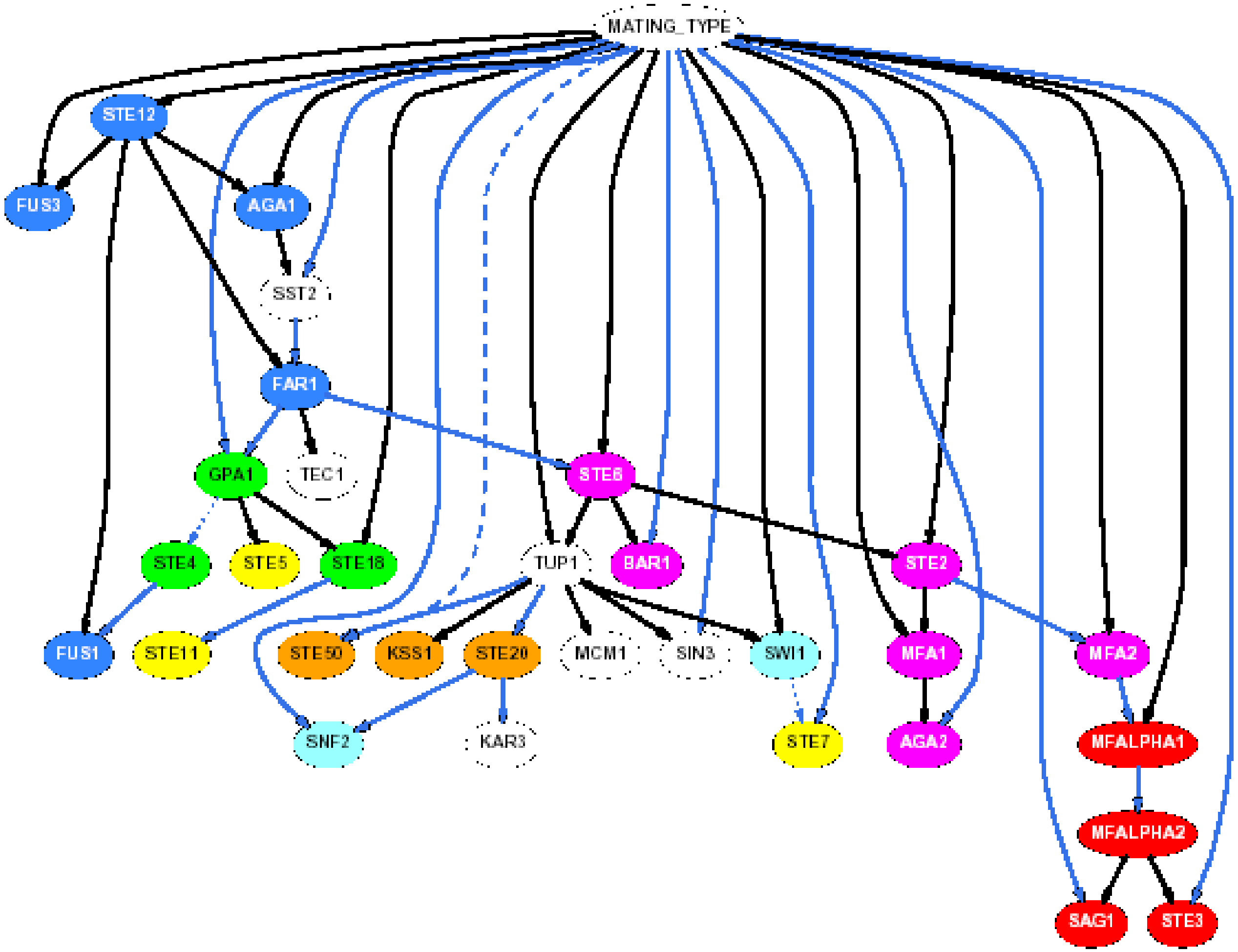
Learned Network Without Constraint



Node color: Different function.

Edge color:Solid black (0.99-1.0), dash blue (0.75-0.99), dot blue (0.5-0.75).

Learned Network With Constraints



Constraints included:

STE12

FUS1

FUS3

AGA1

FAR1

App2. Bayesian Classifier

- Reference:
  - N.Friedman. Building classifier using Bayesian networks. Proc. NCAI 1277-1284, 1996.
  - O.D.King et al. Predicting Gene Function From Patterns of Annotation. *Genome Research* **13**: 896-904, 2003.

Basic Problem

- Given a dataset

$$\{(X_{1,}c), (x_{2,}c), \dots ,(X_{N-1,}c), (X_{N,}c)\}$$

  - Here X_i stands for the training data,c stands for the class label,assuming we have *m* classes,
  - We estimate the probability.

$$P(C_i \mid X),\ i=1,2,\dots,m$$

  - The classifier is then denoted by:

$$\arg\max_i P(C_i \mid X)$$

How can we estimate the posterior probability?

Naïve Bayesian Network

- Assumption: all the variables are independent, given the class label.
- Joint distribution. $P((v_1, v_2 \ldots v_{m-1}, v_m) \mid C) = {}_{i}\; P(v_i \mid C)$

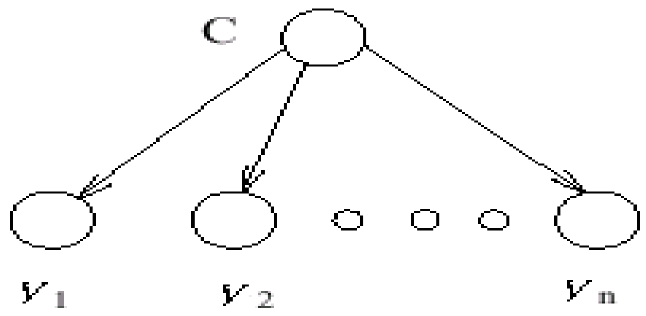


Figure 1. The structure of the naive Bayes network.

Tree Argumented Naive Bayes (TAN) Model

- Bayesian network with the class as the root, will each attribute’s parent set contain class and at most one other attribute.

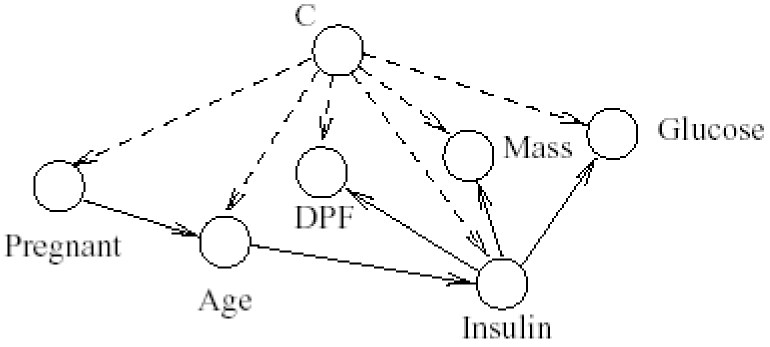


Figure 3. A TAN model learned for the data set “pima.” The dashed lines are those edges required by the naive Bayesian classifier. The solid lines are correlation edges between attributes.

最大权值相关树

- 用一个相关树来近似变量的联合分布

$$q(x) = \prod_{i=1}^{p} p(X_i | X_{P(i)})$$

其中 $X_{P(i)}$ 表示i的父节点所对应的随机变量。

- 采用相对熵描述分布的差异

$$D(p, q) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

- 可以证明，最小化相对熵等价于极大化下面的互信息和

$$W = \sum_{i=1}^{p} I(X_i, X_{P(i)})$$

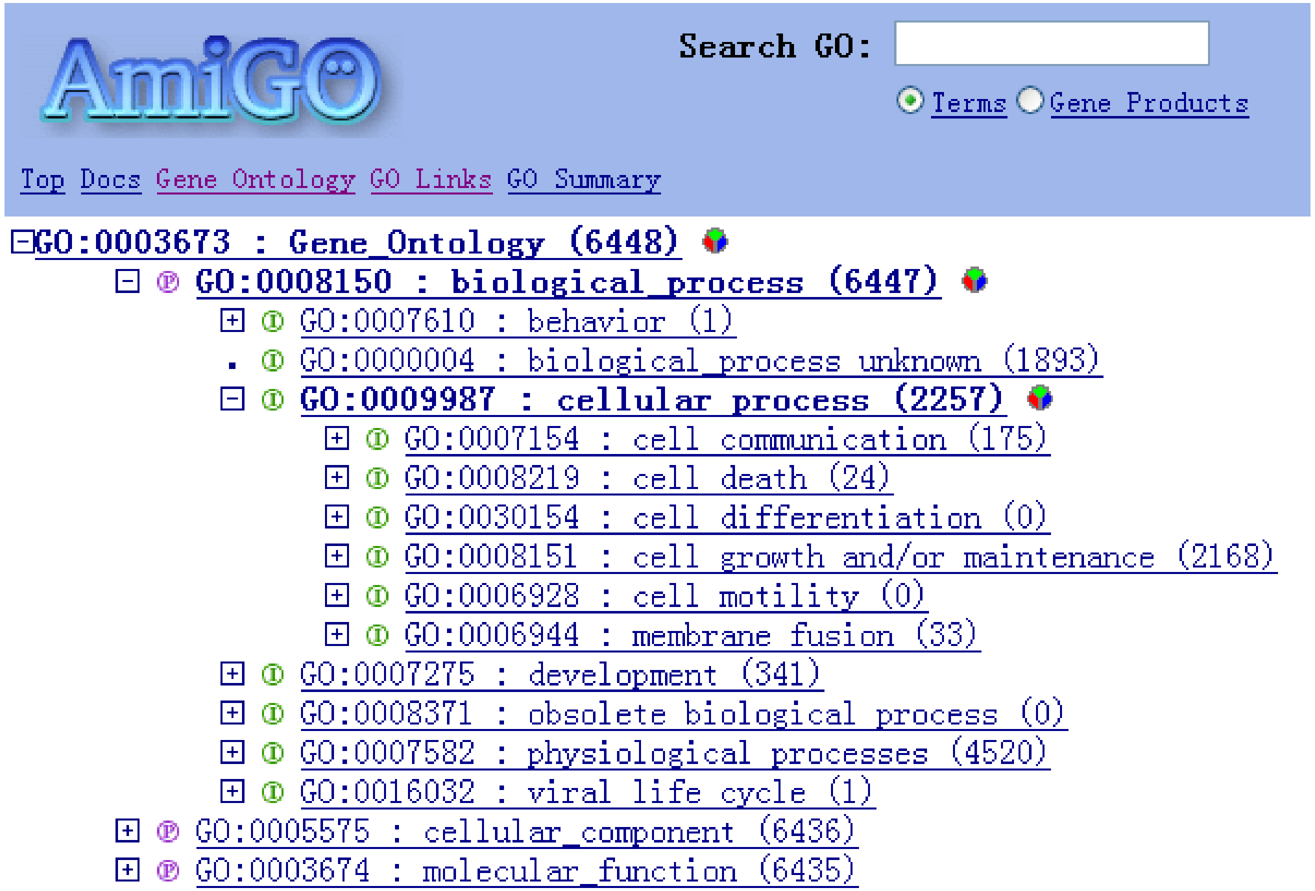
最大权值相关树

- 算法(MWDT, Maximum Weight Dependence Tree), 即为生成树的Kruskal算法
  - 1. 对所有p(p-1)/2个变量对，计算其分支权值，并按降序排列
  - 2. 将权值最大的两个分支放入树中
  - 3. 如果没有形成闭环，则将下一个权值最大的分支加入树中，否则将其抛弃，重复此过程直到p-1个分支已经被选择
  - 4. 选择任意的根节点，计算概率分布

GO Function Prediction

- Motivation: GO is the controlled vocabulary of gene functions. Predict gene function by the pattern of annotation.
- Idea: If the annotation of two attribute tend to occur together in the database, then a gene holding one attribute is likely to hold the other as well.

Gene Ontology Structure



Search GO:

● Terms ○ Gene Products

Top Docs Gene Ontology GO Links GO Summary

- ⊟ GO:0003673 : Gene_Ontology (6448)
  - ⊟ Ⓟ GO:0008150 : biological_process (6447)
    - ⊞ Ⓘ GO:0007610 : behavior (1)
    - Ⓘ GO:0000004 : biological_process unknown (1893)
    - ⊟ Ⓘ GO:0009987 : cellular process (2257)
      - ⊞ Ⓘ GO:0007154 : cell communication (175)
      - ⊞ Ⓘ GO:0008219 : cell death (24)
      - ⊞ Ⓘ GO:0030154 : cell differentiation (0)
      - ⊞ Ⓘ GO:0008151 : cell growth and/or maintenance (2168)
      - ⊞ Ⓘ GO:0006928 : cell motility (0)
      - ⊞ Ⓘ GO:0006944 : membrane fusion (33)
    - ⊞ Ⓘ GO:0007275 : development (341)
    - ⊞ Ⓘ GO:0008371 : obsolete biological process (0)
    - ⊞ Ⓘ GO:0007582 : physiological processes (4520)
    - ⊞ Ⓘ GO:0016032 : viral life cycle (1)
  - ⊞ Ⓟ GO:0005575 : cellular_component (6436)
  - ⊞ Ⓟ GO:0003674 : molecular_function (6435)

Formalization

- GO attributes j. X_j indicate function. $X_j(i)$=1 if gene is annotated with j.
- Attribution set nad(x_j): neither ancestor nor descendant attribute of one attribute j in the GO DAG.

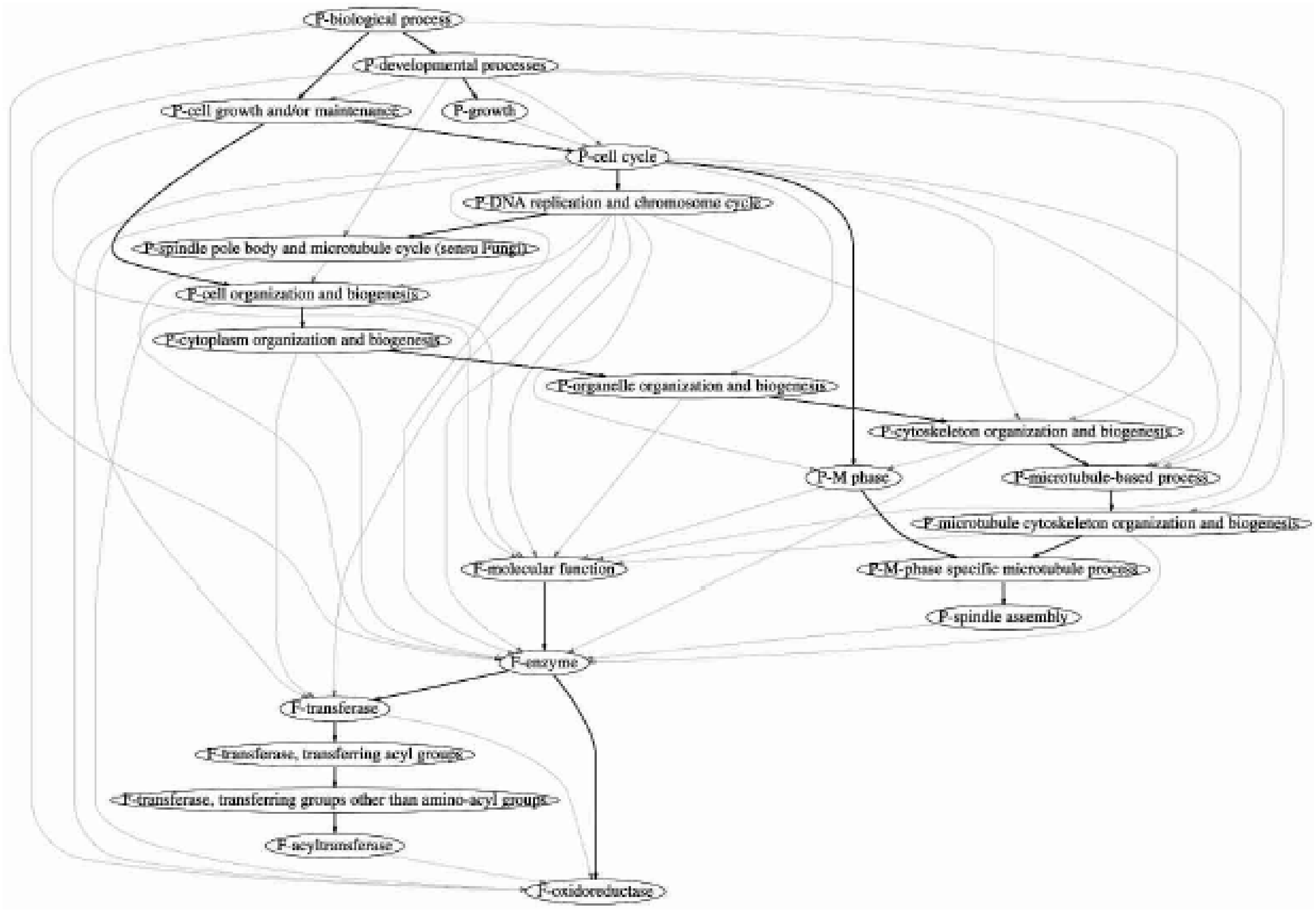
- The task is to estimate the probability

$$q(i,j) = Pr(X_j = 1 | nad(X_i) = nad(X_i)(i))$$

Bayesian Network Model

- Nodes: GO attribute covers more than 10 genes, and no descendant covers more than 10 genes.
  - SGD, 170.
  - FlyBase, 218.
- Constraints: just considering those structures logically consist with GO DAG.

Fragment of Learned Bayesian Network


P-biological process
P-developmental processes
P-cell growth and/or maintenance
P-growth
P-cell cycle
P-DNA replication and chromosome cycle
P-spindle pole body and microtubule cycle (sensu Fungi)
P-cell organization and biogenesis
P-cytoplasm organization and biogenesis
P-organelle organization and biogenesis
P-cytoskeleton organization and biogenesis
P-M phase
P-microtubule-based process
P-microtubule cytoskeleton organization and biogenesis
F-molecular function
P-M-phase specific microtubule process
P-spindle assembly
F-enzyme
F-transferase
F-transferase, transferring acyl groups
F-transferase, transferring groups other than amino-acyl groups
F-acyltransferase
F-oxidoreductase

Further Reading

- N.Friedman et al. A structural EM algorithm for phylogenetic inference. *RECOMB2001*.
- E.Segal et al. From promoter sequence to gene expression data. *RECOMB2002*.
- E.Segal. Regulatory module. Nature Genetics 34: 2003.

Bayesian Network Sourses

- Peoples
  - N.Friedman http://www.cs.huji.ac.il/~nir/
  - D.Heckman http://www.research.microsoft.com/~heckerman/
  - J. PEARL http://bayes.cs.ucla.edu/jp_home.html
  - F.V.Jensen http://www.cs.auc.dk/~fvj/
  - ……

Bayesian Network Sourses

- Bayesian Network Repository
  http://www.cs.huji.ac.il/labs/compbio/Repository/.
- Systems
  - Bayesian Networks Software Package listing
    http://www.cs.berkeley.edu/~zuwhan/bn.html.
  - Microsoft Belief Network Tools
    http://www.research.microsoft.com/research/dtg/msbn/
  - Hugin http://hugin.dk/
  - ……

Part IV Gaussian Graphical Model

Multivariate Gaussian Distribution

- Multivariate Gaussian over all continuous expressions

$$p(x) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right)$$

$$x = (x_1, x_2, \cdots, x_p)^T$$

- MRF表示

$$p(x) \propto \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x + (\Sigma^{-1}\mu)^T x \right)$$

Multivariate Gaussian Distribution

- 定理：如果 A, D 可逆，那么

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -S^{-1}BD^{-1} \\ -D^{-1}CS^{-1} & D^{-1} + D^{-1}CS^{-1}BD^{-1} \end{pmatrix}$$

其中 $S = A - BD^{-1}C$

- 证明：由

$$\begin{pmatrix} I & -BD^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} I & 0 \\ -D^{-1}C & I \end{pmatrix}$$

$$= \begin{pmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{pmatrix}$$

Multivariate Gaussian Distribution

- 得到

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} I & 0 \\ -D^{-1}C & I \end{pmatrix} \begin{pmatrix} (A-BD^{-1}C)^{-1} & 0 \\ 0 & D^{-1} \end{pmatrix} \begin{pmatrix} I & -BD^{-1} \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} (A-BD^{-1}C)^{-1} & -S^{-1}BD^{-1} \\ -D^{-1}CS^{-1} & D^{-1}+D^{-1}CS^{-1}BD^{-1} \end{pmatrix}$$

- 其中 $S = A - BD^{-1}C$

Multivariate Gaussian Distribution

- 定理：对于多元高斯模型，有如下结论

(1). $(\Sigma)^{-1}_{ij} = 0 \Leftrightarrow x_i \perp x_j|_{V/\{i,j\}}$

(2). $(\Sigma)^{-1}_{ij} = 0 \Leftrightarrow \beta_{ij} = 0$

$$\text{in the regression } E(X_j|X_{V/j}) = \sum_{k \neq j} \beta_{jk} X_k$$

Multivariate Gaussian Distribution

- 证明：考虑变量的划分i和V$_2$， $V_2 = V/\{i\}$, 设

$$\Sigma = \begin{pmatrix} \sigma_{ii} & \Sigma_{i,V_2} \\ \Sigma_{V_2,i} & \Sigma_{V_2,V_2} \end{pmatrix}$$

并令

$$\Sigma^{-1} = \begin{pmatrix} d_{ii} & D_{i,V_2} \\ D_{V_2,i} & D_{V_2,V_2} \end{pmatrix}$$

Multivariate Gaussian Distribution

- 由分块矩阵求逆得到

$$D_{i,V_2} = -d_{ii}\Sigma_{i,V_2}\Sigma^{-1}_{V_2,V_2}$$

- 进一步将V$_2$划分为j和B, $B = V/\{i,j\}$, 并设

$$\Sigma_{V_2,V_2} = \begin{pmatrix} \sigma_{jj} & \Sigma_{jB} \\ \Sigma_{Bj} & \Sigma_{BB} \end{pmatrix}, [\Sigma_{V_2,V_2}]^{-1} = \begin{pmatrix} \tilde{d}_{jj} & \tilde{D}_{jB} \\ \tilde{D}_{Bj} & \tilde{D}_{BB} \end{pmatrix}$$

- 可得

$$\tilde{D}_{Bj} = -\tilde{d}_{jj}[\Sigma_{BB}]^{-1}\Sigma_{Bj}$$

Multivariate Gaussian Distribution

- 所以

$$
\begin{aligned}
(\Sigma^{-1})_{ij} &= D_{i,V_2}[j] \\
&= -d_{ii}\left(\Sigma_{ij}\tilde{d}_{jj} + \Sigma_{iB}\tilde{D}_{Bj}\right) \\
&= -d_{ii}\left(\sigma_{ij}\tilde{d}_{jj} - \tilde{d}_{jj}\Sigma_{iB}[\Sigma_{BB}]^{-1}\Sigma_{Bj}\right) \\
&= -d_{ii}\tilde{d}_{jj}\left(\sigma_{ij} - \Sigma_{iB}[\Sigma_{BB}]^{-1}\Sigma_{Bj}\right)
\end{aligned}
$$

Multivariate Gaussian Distribution

- 另一方面，对于联合高斯模型，条件 $X_A|X_B$ 的分布仍然是高斯分布，而且

$$Var(V_A|V_B) = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$$

- 所以

$$\begin{aligned} X_i \perp X_j|V/\{i,j\} \quad &\Leftrightarrow \quad (cov(X_A|X_B))_{ij} = 0 \\ &\Leftrightarrow \quad \sigma_{ij} - \Sigma_{iB}[\Sigma_{BB}]^{-1}\Sigma_{Bj} = 0 \end{aligned}$$

Multivariate Gaussian Distribution

- 因此

$$E(X_i|X_{V/i}) = \sum_{k\neq i} \beta_{ik} X_k$$

$$\Leftrightarrow \beta_{ik} = \mathrm{Argmin} E(X_i - \sum_{k\neq i} \beta_{ik} X_k)^2$$

- 而

$$\beta_{iV_2} = \Sigma_{iV_2}\Sigma^{-1}_{V_2,V_2} = -D_{iV_2}/d_{ii}$$

- 因此 $(\Sigma^{-1})_{ij} = 0$ 等价于上述回归方程中的系数

$$\beta_{ij} = 0$$

Precision Matrix

- The precision matrix $\Omega = \Sigma^{-1}$ reveals the topology of the (undirected) network G=(V, E)

$$\omega_{ij} = 0 \quad \Leftrightarrow \quad e_{ij} \notin E$$

$$\omega_{ij} = 0 \quad \Leftrightarrow \quad x_i \perp x_j \big|_{V/\{i,j\}}$$

Neighborhood Selection

- For each j, let

$$\theta^{j,\lambda} = \arg\min_{\theta} \|X_j - X\theta\|_2^2 + \lambda\|\theta\|_1$$

$$ne(j,\lambda) = \{j : \theta^{j,\lambda} \neq 0\}$$

L1-loglikelihood

- Likelihood

$$L(\Sigma) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \right)$$

- Log-likelihood

$$l(\Sigma) = \log(|\Sigma^{-1}|) - tr(\Sigma^{-1} S)$$

- Penalized log-likelihood Maximization

$$\hat{\Omega} = \underset{\Omega}{\text{Argmin}} \left(-\log|\Omega| + tr(\Omega S) + \lambda \sum_{i \neq j} |\Omega_{ij}| \right)$$

References

- D.Heckman. A tutorial on learning with Bayesian Network.
- N.Friedman. Learning bayesian networks with local structure.
- D.Heckman. Bayesian Networks for data mining. Data Mining and Knoledge Dicovery **1**: 79-119, 1997.
- N.Friedman. Using bayesian networks to analyze expression data. J. Comp. Biol. 2002.
- A.Hartemink Combining location and expression data for principled discovery of genetic regulatory network models. *PSB2002*.
- O.D.King et al. *Genome Res.* **13**: 896-904. 2003.
- Novershtern N, Subramanian A, Lawton LN, Mak RH, Haining WN, McConkey ME, Habib N, Yosef N, Chang CY, Shay T, Frampton GM, Drake AC, Leskov I, Nilsson B, Preffer F, Dombkowski D, Evans JW, Liefeld T, Smutko JS, Chen J, Friedman N, Young RA, Golub TR, Regev A, Ebert BL. Densely interconnected transcriptional circuits control cell states in human hematopoiesis. *Cell*. 144(2):296-309. 2011.