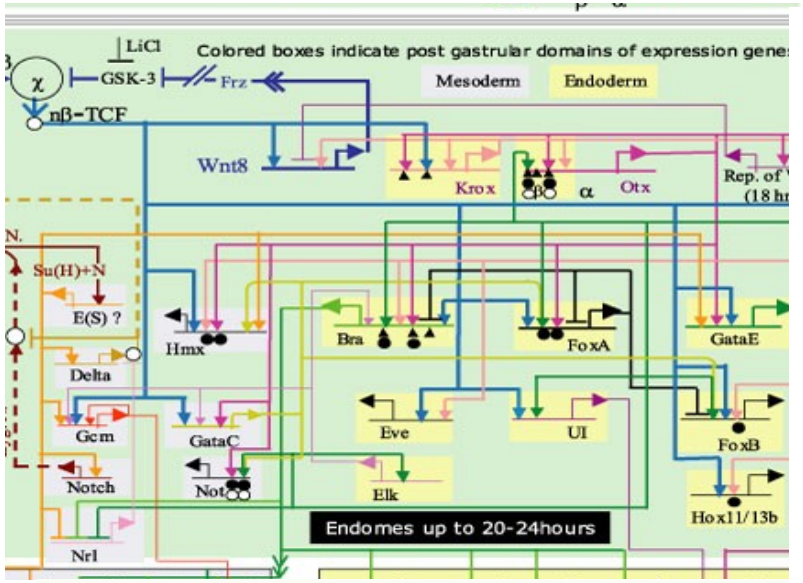
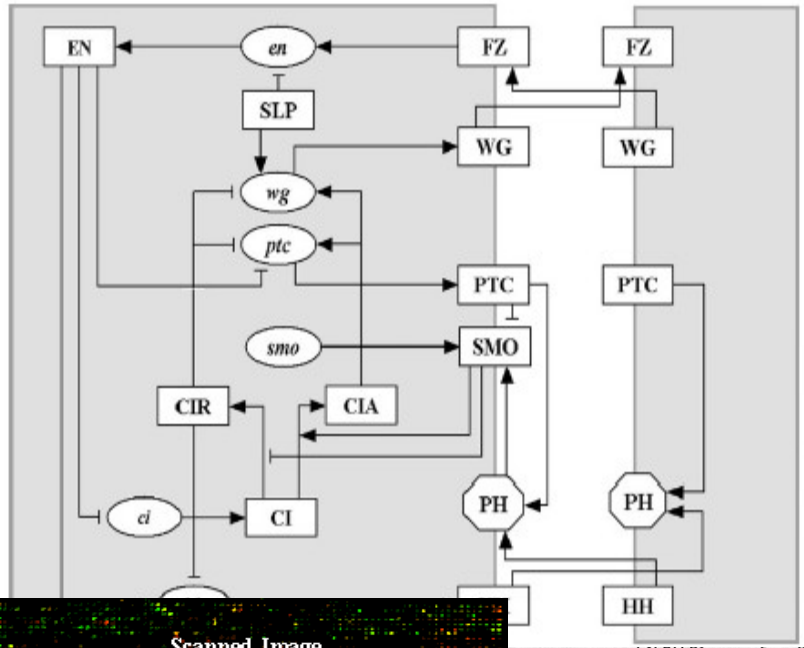
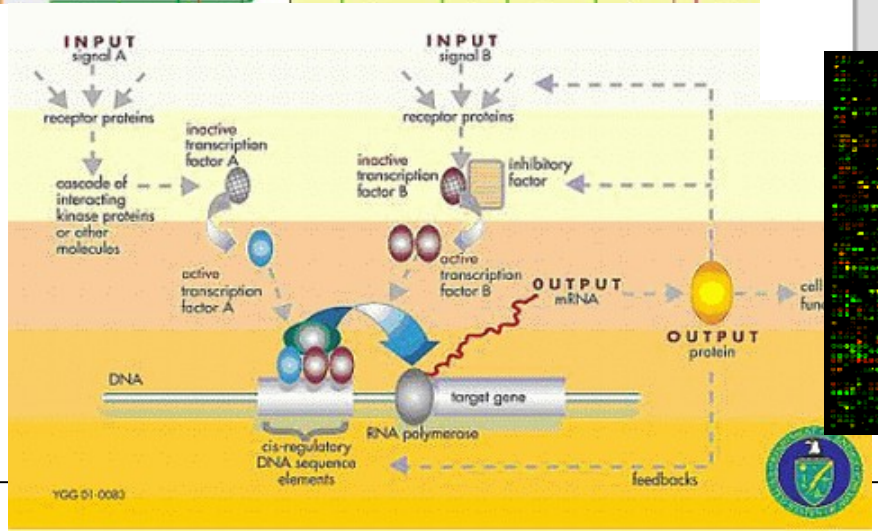
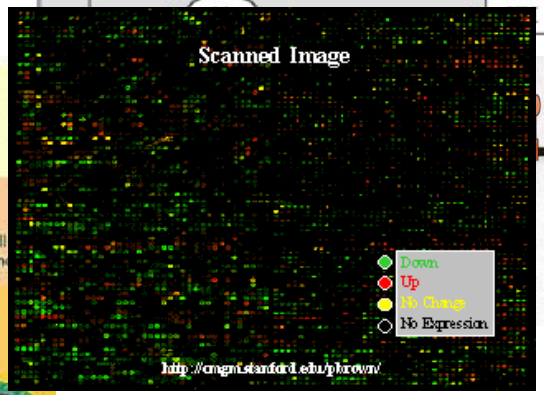
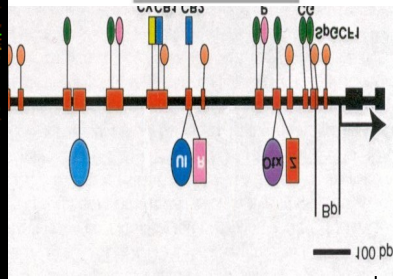
ECS 234: Combinatorial Models of Gene Regulation









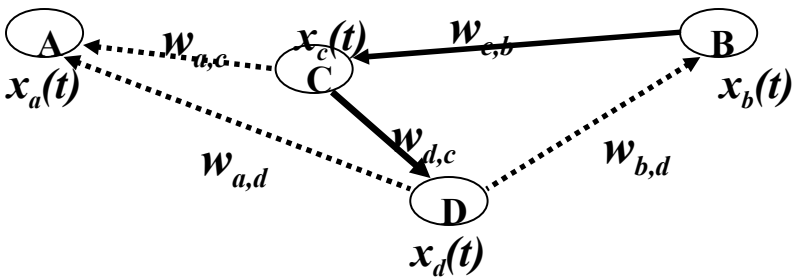


1. Linear Models
2. Graph Theoretic Models
3. Boolean Networks
4. Bayesian Networks

1. Linear (Weight Matrix) Models of Regulation

Description of the Model

- A graph model in which the nodes are genes that are in continuous states of expression (i.e. gene activities). The edges indicate the strength (weight) of the regulation relationship between two genes
- The net effect of gene *j* on gene *i* is the expression level of gene *j* multiplied by its regulatory influence on *i*, i.e. $w_{ij}x_{j.}$
- Assumptions:
  - regulators' contribution to a gene's regulation is linearly additive
  - the states of the nodes are updated synchronously



$x_i(t)$ – state of gene *i* at time *t*

w_{ij} – regulatory influence of gene *j* on gene *i*

- $w_{ij} > 0$, activation
- $w_{ij} < 0$, inhibition
- $w_{ij} = 0$, none

Calculating the Next State of the System

$$x_i(t+1) = \sum_{j=1}^{n} w_{ij} x_j(t)$$

$$x_i, w_{i,j} \in \mathbf{R}$$

Or in matrix notation :

$$\mathbf{x}_{t+1}^{(n\times 1)} = \mathbf{W}^{(n\times n)} \cdot \mathbf{x}_t^{(n\times 1)}$$

If all the weights, w_{ij} are known, then given the activities of all the genes at time t, i.e. $x_1(t), x_2(t), \ldots, x_n(t)$, we can calculate the activities of the genes at time $t+1$.

Fitting the Model to the Data

- In reality, we don't know the weights, and we would like to infer them from measurements of the activities of genes through time (microarray data)
- The weights can be found by solving a system of linear equations (multiple regression)
- Dimensionality Curse: the expression matrices, of size n x k, where n is in thousands and k is at most in hundreds
- The linear system is always under-constrained and thus yields infinitely many solutions (compare to over-constrained where we need to use least-squares fit)

Solving the Linear Model

Let the vector $\mathbf{y_i}$ represent the expressions of n genes at time point i, i.e.

$\mathbf{y_i} = [x_1(i) \quad x_2(i) \quad \cdots \quad x_n(i)]$.

Then, given $k+1$ time points, i.e. vectors $\mathbf{y_i}$, $i = 1,\ldots,k+1$, let

$\mathbf{A^{(k \times n)}}$ be a matrix with rows equal to the first k vectors, i.e. $\mathbf{A} = \begin{bmatrix} \mathbf{y_1} \\ \mathbf{y_2} \\ \cdots \\ \mathbf{y_k} \end{bmatrix}$, and

$\mathbf{B^{(k \times n)}}$ be a matrix with rows equal to the last k vectors, i.e. $\mathbf{B} = \begin{bmatrix} \mathbf{y_2} \\ \mathbf{y_3} \\ \cdots \\ \mathbf{y_{k+1}} \end{bmatrix}$.

Then, the linear system becomes :

$\mathbf{A \bullet W^T = B}$, which we want to solve for $\mathbf{W}$

- If $k > n$, the system is overconstrained, and there is no unique solution. A least squares (regression) solution :

$$\mathbf{W = A^{**}B}, \quad \boxed{\mathbf{A^{**} = (A^T A)^{-1} A^T}}$$

- If $k = n$ there is a unique solution;

- If $k < n$, the system is underconstrained, and there are infinitely many solutions. We can find a pseudo - inverse to $\mathbf{A}$ that best fits the data (Moore - Penrose), as :

$$\mathbf{W = A^{**}A}, \quad \boxed{\mathbf{A^{**} = A^T (A A^T)^{-1}}}$$

Normalization

- The input gene expressions need to be normalized at each step, so that the contributions are comparable across all genes
- The resulting (output) values are then de-normalized
- Common normalization schemes:
  - mean/variance: $x' = (x - \mu)/\sigma^2$
  - Squashing function: (neural nets)

Limitations

- Some assumptions are known to be incorrect:
  - all genetic interactions are independent events
  - synchronous dynamics
  - weight matrix
- The results may not offer insight to the problem instead they may just model the data well (the weight matrix will be chosen based on multiple regression)

How Much Data?

- If the weight matrix is dense, we need *n+1* arrays of all *n* genes to solve the linear system, assuming the experiments are independent (which is not exactly true with time-series data). In this case we say that the average connectivity is *O(n)* per node.
- If instead the average connectivity per node is fixed to *O(p),* than it can be shown that the number of experiments needed is *O(p*log(n/p))*

Summary

- Linear models yield good, realistic looking predictions, but additional assumptions are necessary
- The amount of data needed is *O(n)* experiments, for a fully connected network or *O(p*log(n/p))* for a *p*-connected network
- The weight matrix can be obtained by solving a linear system of equations
- Dimensionality curse: more genes than experiments. We have to resort to reducing the dimensionality of the problem (e.g. through clustering)

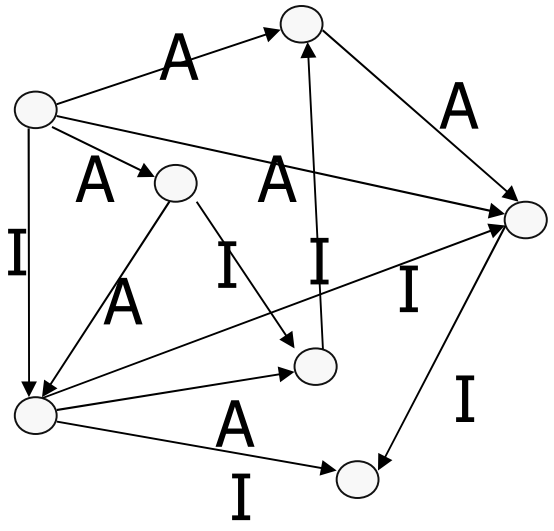
2. Graph Theoretic Models

1. Definitions
2. Chen et al 1999: Qualitative gene networks from time-series data
    1. Parsimony: # Regulators is small
    2. Can we capture regulatory relationships well with correlation arguments?
3. Wagner 2001: Causal networks from perturbation data
    1. Parsimony: # Relationships is minimal
    2. Direct vs. indirect relationships
    3. A perturbation model to detect direct relationships

Graph Theory (Static Graph) Models

Network: directed graph G=(V,E), where V is set of vertices, and E set of edges on V.

- The nodes represent genes and an edge between v_i and v_j symbolizes a dependence between v_i and v_j.
- The dependencies can be *temporal* (causal relationship) or *spatial* (cis-trans specificity).
- The graph can be annotated so as to reflect the nature of the dependencies (e.g. **promoter, inhibitor**), or their strength.



Properties:

- Fixed Topology (doesn't change with time)
- Static
- Node States: Deterministic

A Simple Static Graph Model From Microarray Data (Chen et al. 1999)

- Motivation
  - Time-series data of gene expressions in yeast
  - Is it possible to elucidate regulatory relationships from the up/down patterns in the curves?
  - Could one select a gene network from many candidates, based on a parsimony argument?
- Grand Model
  - Nodes = genes
  - Edges = relationships and labeled A, I, N from the data
  - The graph is a putative regulatory network, and has too many edges
  - Since the model over-fits the data, there is a need for additional assumptions
  - Parsimony argument: few regulators, many regulatees
  - Solve using a meta-heuristic: simulated annealing, genetic algorithms etc.

Nodes: genes

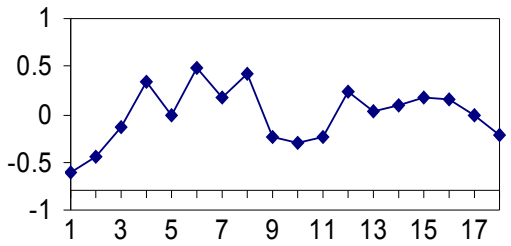
Edges: regulatory relationships (A, I, N)

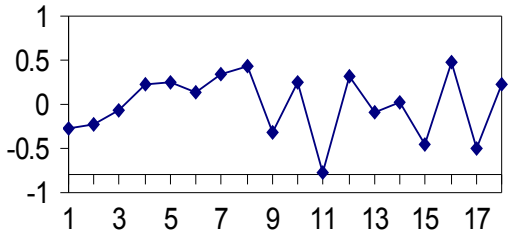
Goal: A labeling of genes as A or I

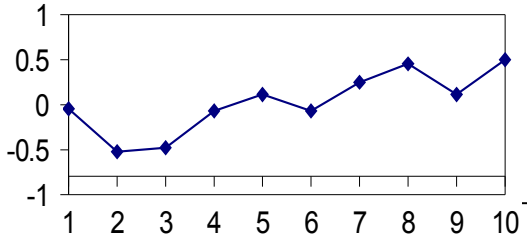
Assumptions:

- # of regulators is small
- each acts as either A or I (approx.)

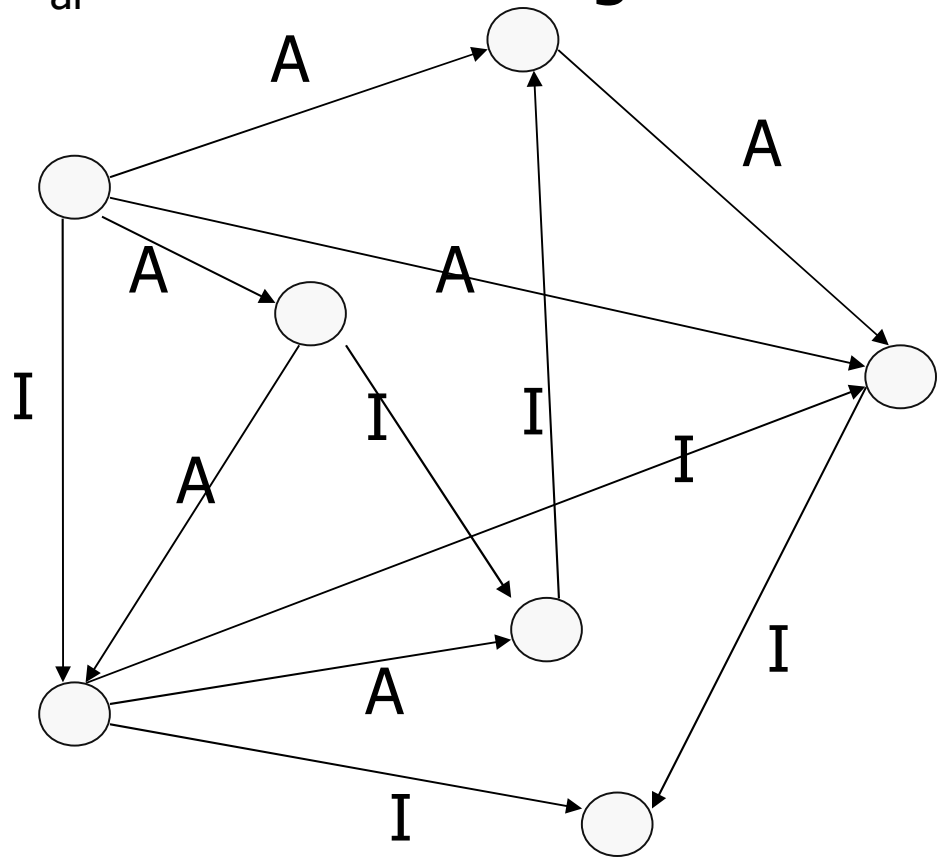
Signal Analysis: putative regulation







Reduce the time-series data set to a graph G_{ai} where each edge is labeled A, I.

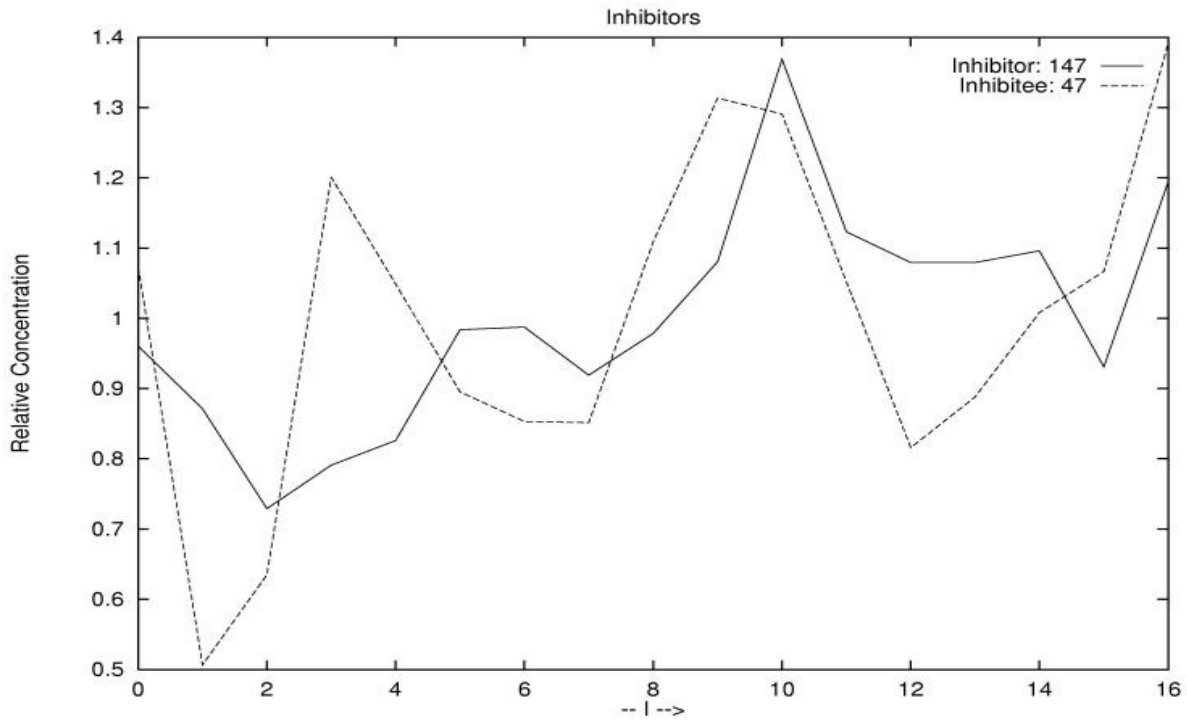


A system for inferring gene- regulation networks:

- Filter **(thresholding)**
- Cluster **(average link clustering)**
- Curve Smoothing **(peak identification)**
- Inferring Regulation Scores
- Optimizing regulation assignment

Yeast genome microarray data, Cho et al (1998)

Inferring Regulation Scores



Optimizing the Graph

Goal: Given a directed graph G with edges labeled A or I and weighted, output a labeling of vertices which optimizes:

$$f(G_{ai}) = \sum_{v_i \in V(G_{ai})} max(v_i[I] \cdot v_i[A]) - C(count(A) + count(I))$$

General optimization technique (Simulated Annealing)

Simulated Annealing

$$f(G_{ai}) = \sum_{v_i \in V(G_{ai})} \max(v_i[I]) \cdot \max(v_i[A]) - C(count(A) + count(I))$$

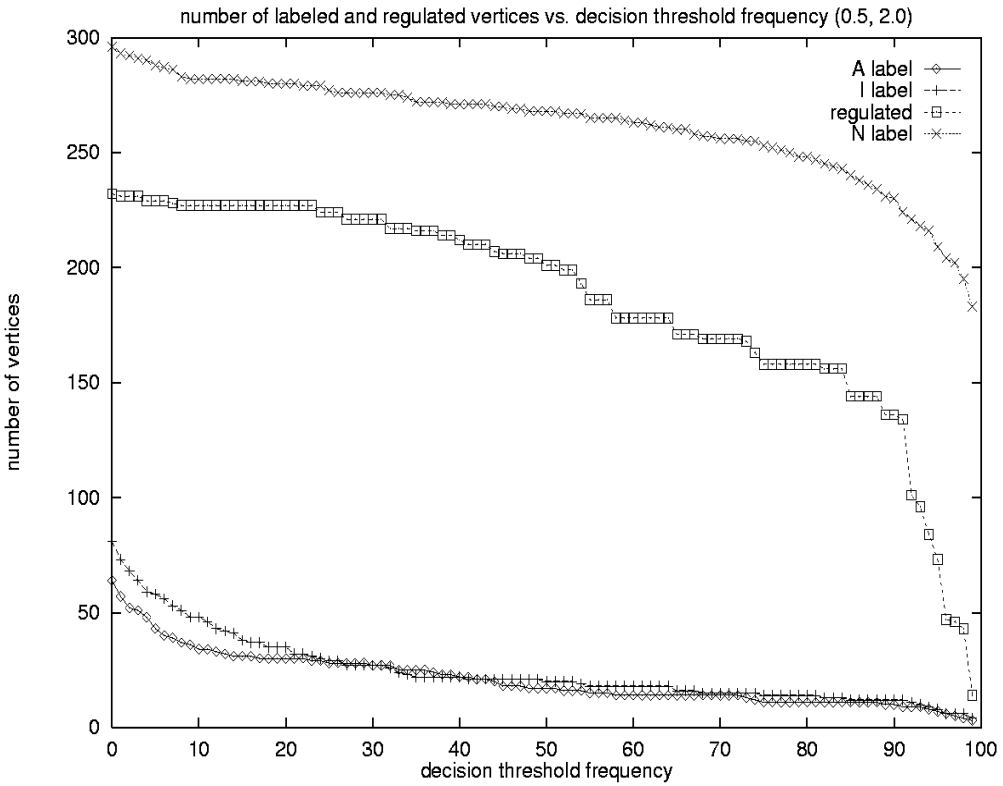
- Simulated annealing is a random, iterative search technique which simulates the natural process of metal annealing
- Problem: Minimize a function f(x)
- Solution: Get closer to the solution iteratively by randomly accepting worse solutions, with the acceptance probability decreasing with time

```
Algorithm: Given f(x) and x
1.
2.
3.
4.
5.
        -
        -
6.
7.
```

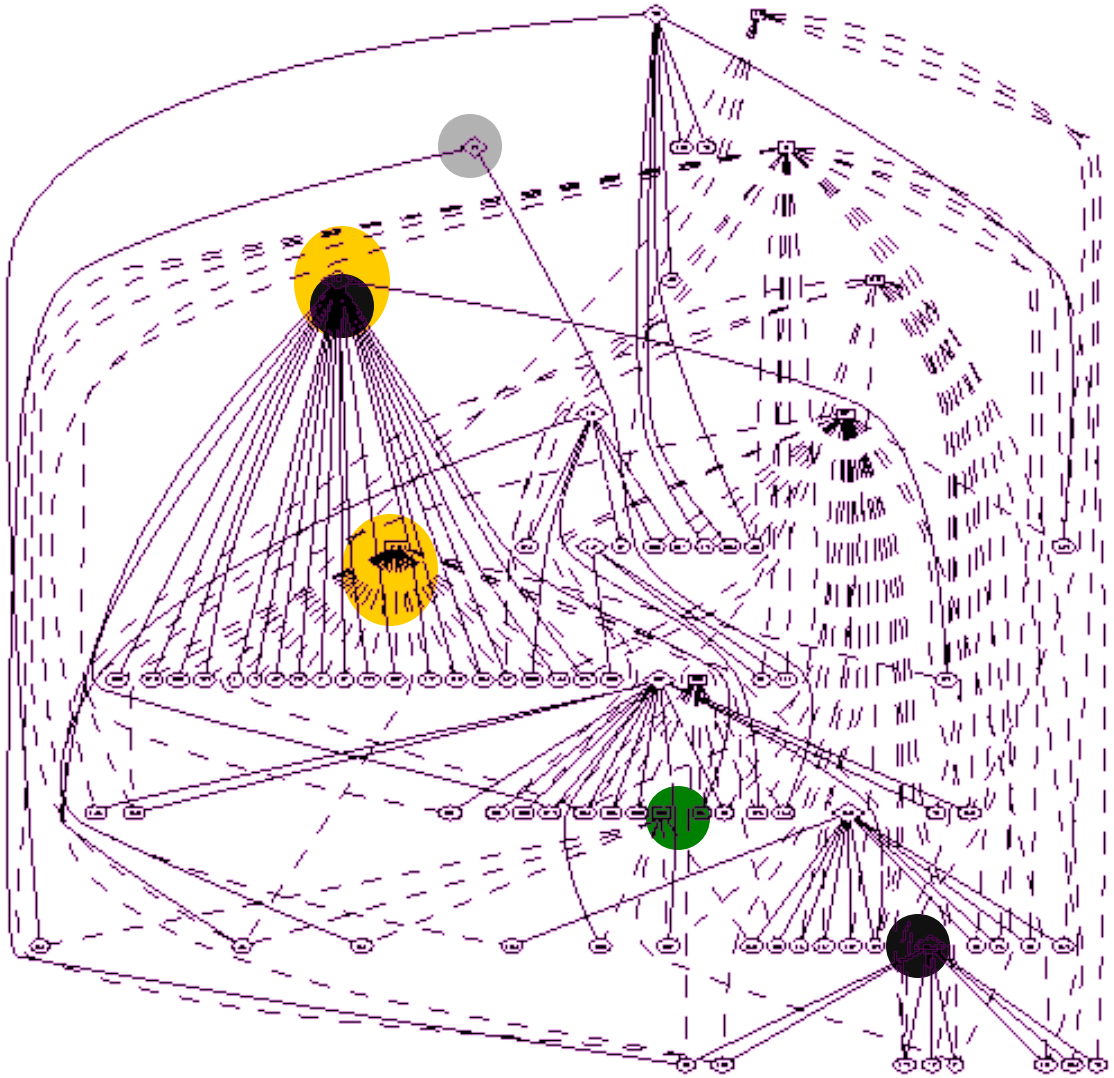
Computational Results

Repeated runs (random starts) produce different A/I networks

P is the perc. of time a network is labeled consistently



A network @ P=95%

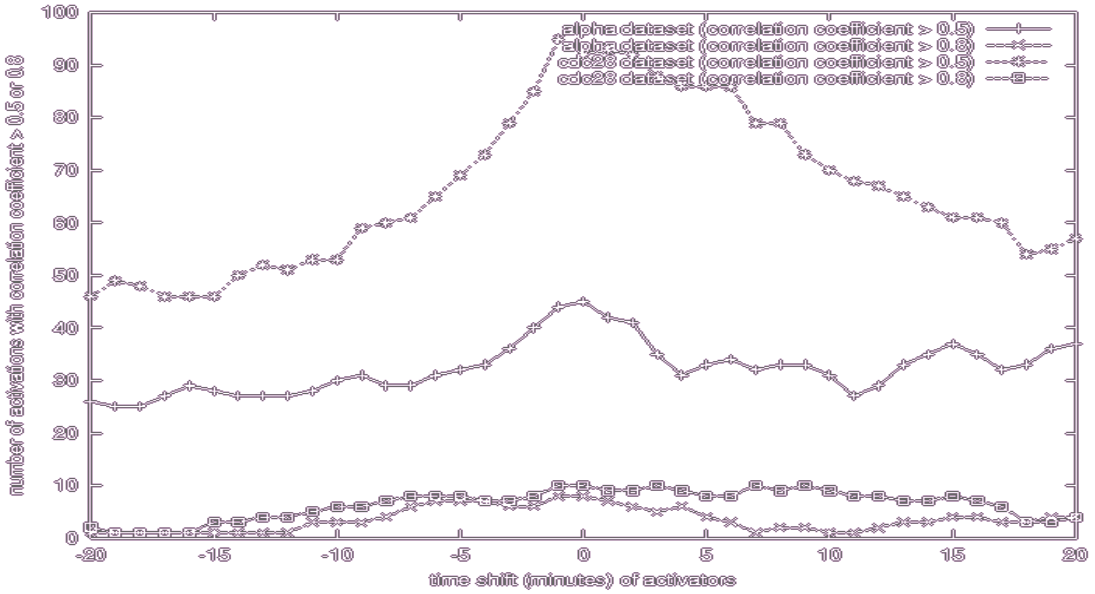


Observations:

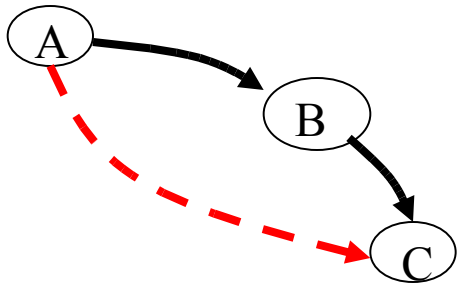
- an activator / inhibitor pair that regulate 15 other clusters
- cell cycle regulator
- DNA replication
- amino acid synthesis

How Well Can We Capture Relationships by Correlation?

- Experiments performed on 4 different data sets of time series expression
- < 20% of regulatory relationships could be predicted by correlating pairs of curves (Filkov et al. 2001)
- Time-shift between curves does not change matters



Direct vs. Indirect Relationships



Direct:
A ⇒B
B ⇒C

Indirect
A ⇒C

- How can we distinguish between direct and indirect relationships in a network based on microarray data?
- Additional assumptions needed
- In the previous model: optimize *f(grade,#regulators)*
- Next: minimize # relationships

Perturbation Static Graph Model (Wagner, 2001)

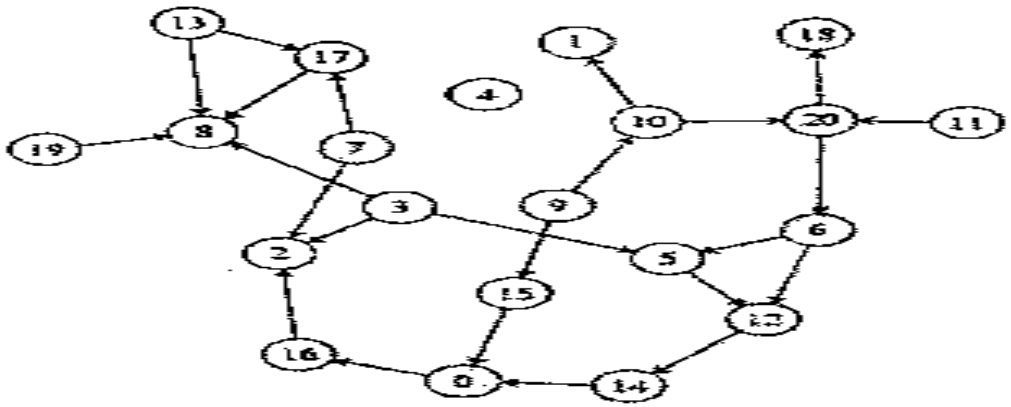
- Motivation: perturbing a gene network one gene at a time and using the effected genes in order to discriminate direct vs. indirect gene-gene relationships
- Perturbations: gene knockouts, over-expression, etc.

Method:

1. For each gene g_i, compare the control experiment to perturbed experiment (gene g_i) and identify the differentially expressed genes
2. Use the most parsimonious graph that yields the graph of 1. as its reachability graph

A single gene perturbation affects multiple genes. The question is which of them directly?

A



B

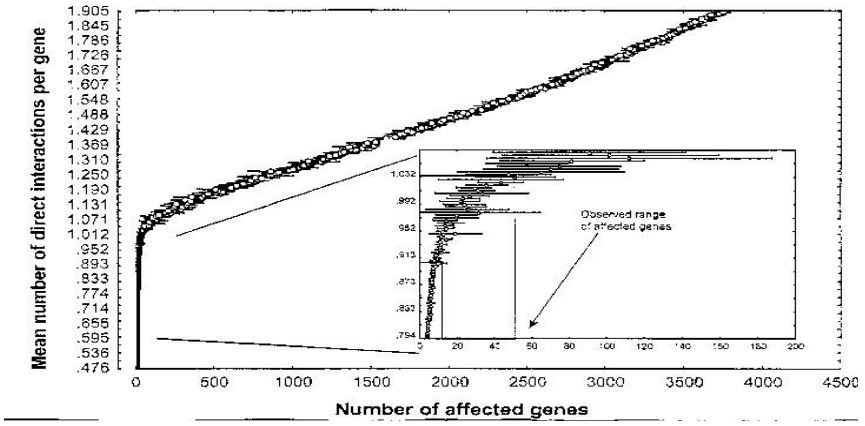
0:	16
1:	
2:	
3:	2 5 8
4:	
5:	12
6:	5 12
7:	2 17
8:	
9:	10 15
10:	1 20
11:	20
12:	14
13:	8 17
14:	0
15:	0
16:	2
17:	8
18:	
19:	8
20:	6 18

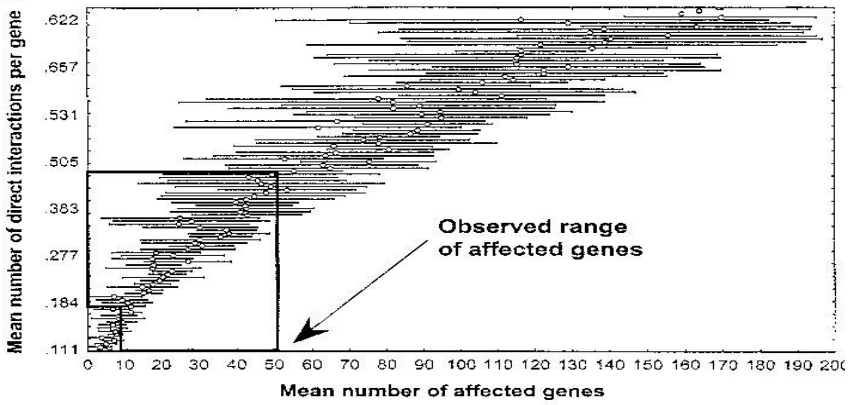
C

0:	2 16
1:	
2:	
3:	0 2 5 8 12 14 16
4:	
5:	0 2 12 14 16
6:	0 2 5 12 14 16
7:	2 8 17
8:	
9:	0 1 2 5 6 10 12 14 15 16 18 20
10:	0 1 2 5 6 12 14 16 18 20
11:	0 2 5 6 12 14 16 18 20
12:	0 2 14 16
13:	8 17
14:	0 2 16
15:	0 2 16
16:	2
17:	8
18:	
19:	8
20:	0 2 5 6 12 14 16 18

Parsimony Assumptions

- The direct relationship graph:
  - is random (ER graphs)
  - is scale-free (Power law)
  - has the smallest number of edges
- Based on the first two assumptions above, the author investigated the sparseness of the yeast gene regulatory network, based on gene knockout experiments (Hughes et al, 2000)
- Results: the yeast regulatory networks are sparse (~1 connection per gene, even fewer if they are scale-free)





Reconstructing the Network

- The best graph of all is the one with the least relationships
- Problem: Given a transitive closure of a graph calculate its transitive reduction, i.e. the graph with the same transitive closure, and the smallest number of edges
- Problem is easily solvable in polynomial time
- Data needed: n perturbation experiments. If n=6200+ this is unfeasible!

Graph Theoretic Models

Summary

- Characteristic of these models is the underlying graph structure
- The graphs may annotated to reflect the qualitative properties of the genes, i.e. activators, inhibitors
- Edges may be annotated to reflect the nature of the relationships between genes, e.g. =>,⇔, etc
- Depend on a “regulation grade” between genes
- Time-series data yield graphs of causal relationships
- Perturbation data also yield graphs of causal relationships
- Parsimony arguments allow for consideration of biological principles, e.g. small number of regulatory genes, but
- They are overall very naïve biologically

3. Boolean Network Models

- Kaufmann, 1970s studied organization and dynamics properties of (N,k) Boolean Networks
- Found out that highly connected networks behave differently than lowly connected ones
- Similarity with biological systems: they are usually lowly connected
- We study Boolean Networks as a model that yields interesting complexity of organization and leave out the philosophical context

Boolean Functions

- True, False: 1,0
- Boolean Variables: x, can be true or false
- Logical Operators: and, or, not
- Boolean Functions: k input Boolean variables, connected by logical operators, 1 output Boolean value
- Ex: ***f(x,y)=(x AND y) OR (NOT x)***
- Total number, B, of Boolean functions of k variables: 2^{2^k} (k =1, B=4; k=2, B=16; etc.)

Boolean Networks

Boolean network: a graph G(V,E), annotated with a set of states X={x_i | i=1,…,n}, together with a set of Boolean functions B={b_i | i=1,…,k}, $b_i : \{0,1\}^k \to \{0,1\}$.

Gate: Each node, v_i, has associated to it a function , with inputs the states of the nodes connected to v_i.

Dynamics: The state of node v_i at time t is denoted as $x_i(t)$. Then, the state of that node at time t+1 is given by:

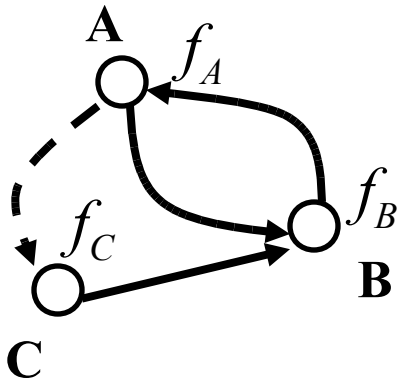
$$x_i(t+1) = b_i(x_{i1}, x_{i2}, \ldots, x_{ik})$$

where x_{ij} are the states of the nodes connected to v_i.

General Properties of BN:

- Fixed Topology (doesn't change with time)
- Dynamic
- Synchronous
- Node States: Deterministic, discrete (binary)
- Gate Function: Boolean
- Flow: Information

Wiring Diagrams and Truth Tables

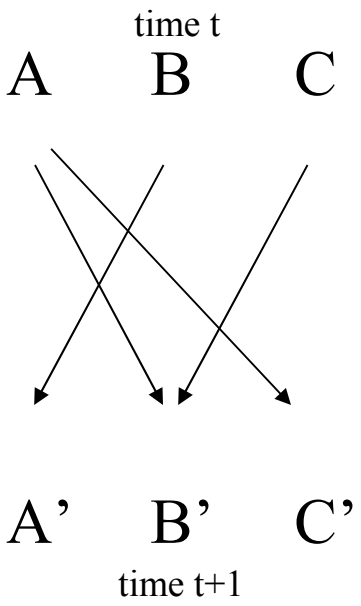


$f_A(B) = B$

$f_B(A,C) = A \text{ and } C$

$f_C(A) = \text{not } A$

Boolean Network



Wiring Diagram

State	INPUT			OUTPUT		
	A	B	C	A'	B'	C'
1	0	0	0	0	0	1
2	0	0	1	0	0	1
3	0	1	0	1	0	1
4	0	1	1	1	0	1
5	1	0	0	0	0	0
6	1	0	1	0	1	0
7	1	1	0	1	0	0
8	1	1	1	1	1	0

Truth Table

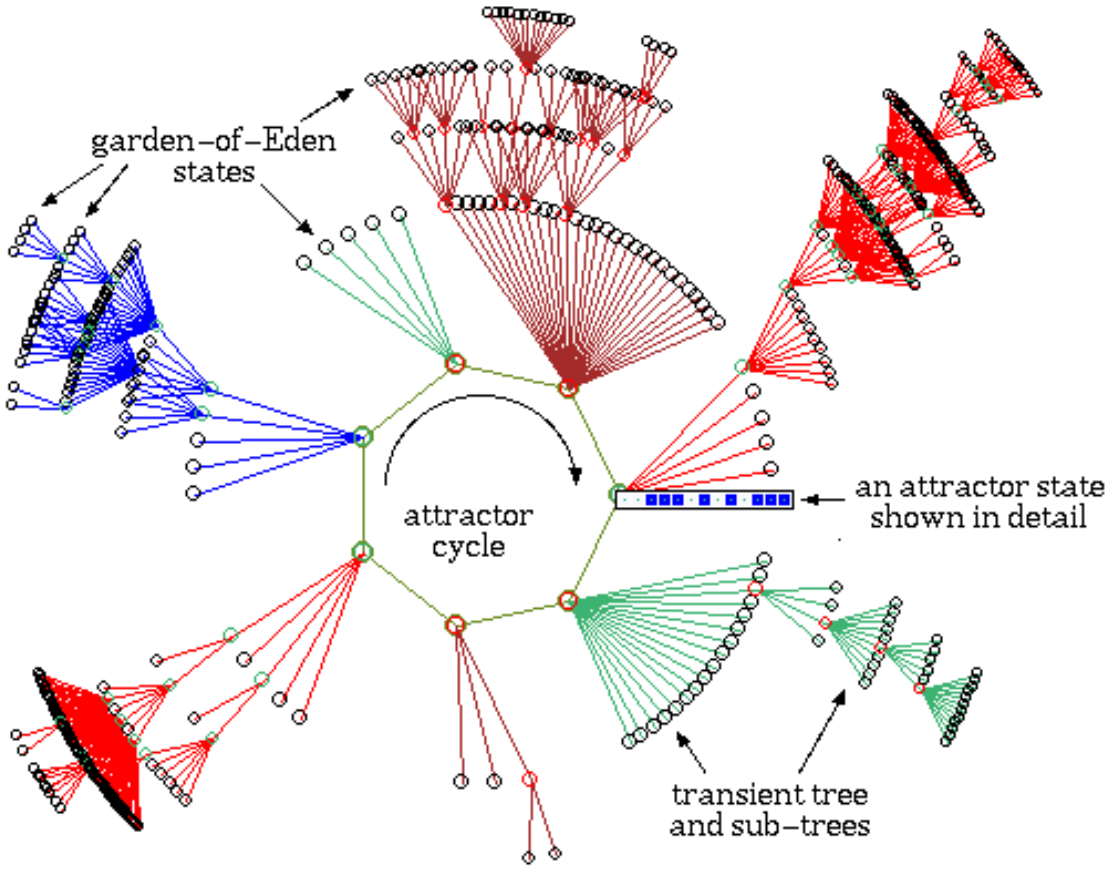
Network States and Transitions

- State: Values of all variables at given time
- Values updated synchronously
- State Transitions: Input → Output
- Ex. (100 → 000 → 001 → 001 . . .)

State	INPUT			OUTPUT		
	A	B	C	A'	B'	C'
1	0	0	0	0	0	1
2	0	0	1	0	0	1
3	0	1	0	1	0	1
4	0	1	1	1	0	1
5	1	0	0	0	0	0
6	1	0	1	0	1	0
7	1	1	0	1	0	0
8	1	1	1	1	1	0

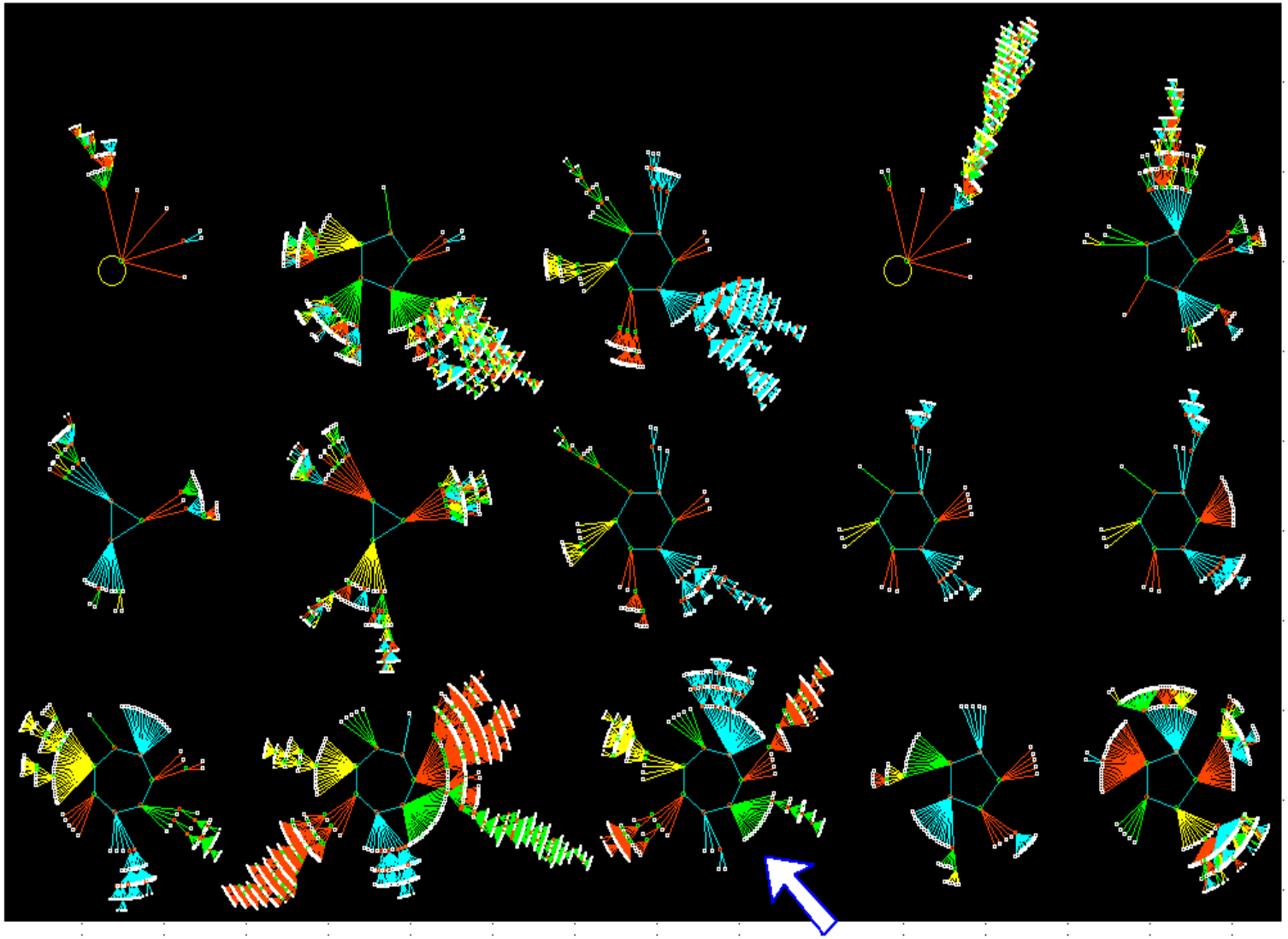
BN Dynamics

- Trajectories: series of state transitions
- Attractors: repeating trajectories
- Basin of Attraction: all states leading to an attractor



One attractor basin for a BN n=13, k=3. The cycle is of size 7

Wuensch, PSB 1998



Previous basin of attraction is one of 15 possible ones for n=13 and k=3. Total of 8192 states, and attractors with periods ranging from 1 to 7 (Pictures come from DDLab Galery, Wuensche, Santa Fe Institute)

Why Are BNs Good for Biology? Simulation

- Complex behavior (synergistic behavior)
  - Attractor steady states which can be interpreted as memory for the cell
  - Stability and reproducibility
  - Robustness
- The range of behaviors of the system is completely known and analyzable (for smaller networks) and is much smaller than the range of the individual variables
- Organizational properties:
  - high connectivity (k>5) yields chaotic behavior
  - Low connectivity (k=2) attractor number and median attractor length are *O(Sqrt(n))*
- Simple to implement and use

BN and Biology

From mRNA measures to a Regulation Network:

1 *Continuous gene expression values are discretized as being 0 or 1 (on, off), (each microarray is a binary vector of the states of the genes);*

2 *Successive measurements (arrays) represent successive states of the network i.e. X(t)->X(t+1)->X(t+2)…*

3 *A BN is reverse engineered from the input/output pairs: (X(t),X(t+1)), (X(t+1),X(t+2)), etc.*

Reverse Engineering of BNs

- Fitting the data: given observations of the states of the BN, find the truth table
- In general, many networks will be found
- Available algorithms:
  - Akutsu et al.
  - Liang et al. (REVEAL)

Formal Problem

- An example is a pair of observations (I_j, O_j).
- A node is consistent with an example, if there is a Boolean function such that $O_j = f(I_j)$
- A **BN** is consistent with (I_j, O_j) if all nodes are consistent with that example. Similarly, a **BN** is consistent with $\boldsymbol{EX} = \{(I_1, O_1), \ldots, (I_m, O_m)\}$ if it is consistent with each example
- Problem: **Given EX, n the number of nodes in the network, and k (constant) the max indegree of a node, find a BN consistent with the data**.

Algorithm (Akutsu et al, 1999)

The following algorithm is for the case of $k=2$, for illustration purposes. It can easily be extended to cases where $k>2$

- For each node v_i
  - For each pair of nodes v_k and v_h and
    - For each Boolean function ***f*** of 2 variables (16 poss.)
      - Check if $O_j(v_i)=f(I_j(v_k),I_j(v_h))$ holds for all j.

Analysis of the Algorithm

- Correctness: Exhaustive
- Time: Examine all Boolean functions of 2 inputs, for all node triplets, and all examples $O(2 \cdot 2^{2^2} \cdot n^3 \cdot m)$
- For k inputs (k in front is the 2 above, time to access the k input observations) $O(k \cdot 2^{2^k} \cdot n^{k+1} \cdot m)$
- This is polynomial in n, if k is constant.

Better Algorithms?

- If in-degree is fixed to at most k,
  - the best known deterministic algorithms run in $O(mn^k)$ time
  - Randomized: $O(m^{w-2}n^k+mn^{k+w-3})$, where w is the exponent in matrix multiplication, currently $w<2.376$ (Akutsu et al., 2000)
- If in-degree is close to n, the problem is NP-complete (Akutsu et al., 2000)

Data Requirement

- How many examples (I,O) do we need to reconstruct a Boolean Network?
- If in-degree unbounded 2^n
- If in-degree<k, information theoretic aruments yield the following bounds:
  - Upper bound $O(2^{2k} \cdot (2k + \alpha) \cdot \log n)$
  - Lower bound $\Omega\,(2^k + K \log n)$
- Experiments show that the constant in front of the log n is somewhere in between, i.e. $k2^k$

Limitations

- BNs are Boolean! Very discrete
- Updates are synchronous
- Only small nets can be reverse engineered with current state-of-the-art algorithms

Summary

- BN are the simplest models that offer plausible real network complexity
- Can be reverse engineered from a small number of experiments *O(log n)* if the connectivity is bounded by a constant. 2^n experiments needed if connectivity is high
- Algorithms for reverse engineering are polynomial in the degree of connectivity

4. Bayesian (Belief) Network Models

Bayes Logic

- Given our knowledge that an event may have been the result of two or more causes occurring, what is the probability it occurred as a result of a particular cause?

- We would like to predict the unobserved, using our knowledge, i.e. assumptions, about things

Why Bayesian Networks?

- Bayesian Nets are graphical (as in graph) representations of precise statistical relationships between entities
- They combine two very well developed scientific areas: Probability + Graph Theory
- Bayesian Nets are graphs where the nodes are random variables and the edges are directed causal relationships between them, $A \rightarrow B$
- They are very high level qualitative models, making them a good match for gene networks modeling

Bayesian Networks (G,θ)

(1) An annotated directed acyclic graph G(X,E), where the nodes are random variables Xi ∈ X
(2) conditional distributions θi= P(Xi | ancestors(Xi)) defined for each Xi.

A Bayesian network uniquely specifies a joint distribution:

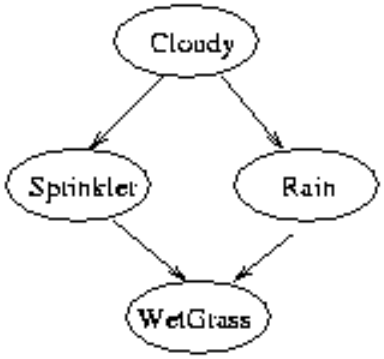
$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid \text{Parents}(X_i))$$

From the joint distribution one can do inferences, and choose likely causalities

Example

KP Murphy's web site:
http://www.cs.berkeley.edu/~murphyk/Bayes/bayes.html

P(C=F)	P(C=T)
0.5	0.5



P(S|C)

C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

P(R|C)

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

P(W|S,R)

Joint: *P(C,R,S,W)=P(C)*P(S|C)*P(R|C)*P(W|S,R)*

Independencies: ***I(S;R|C), I(R;S|C)***

Dependencies: *P(S|C), P(R|C), P(W|S,R)*

Example, contd.

Which event is more likely, wet grass observed and it is because of

- sprinkler:

$$P(S=1|W=1)=P(S=1,W=1)/P(W=1)=0.430$$

- rain:

$$P(R=1|W=1)=P(R=1,W=1)/P(W=1)=0.708$$

Algorithms exist that can answer such questions given the Bayesian Network

General Properties of BNs

- Fixed Topology (static BNs)
- Nodes: Random Variables
- Edges: Causal relationships
- DAGs
- Allow testing inferences from the model and the data

Learning the Network

- Given data we would like to come up with Bayesian Network(s) that fit that data well

- *Problem:* *Given a training set* $D=(x^1,x^2,\ldots,x^n)$ *of independent instances of the random variables* $(X_1,X_2,\ldots,X_n)$*, find a network G (or equivalence class of networks) that best matches D.*

- Algorithms exist that can do this efficiently (though the optimal ones are NP-complete)
- Heuristics are typically used

Parameter Fitting and Model Selection

- Parameter Fitting: If we know G and we want θ
  - Parametric assignments optimized by Maximum Likelihood
- Model Selection: G is unknown
  - Discrete optimization by exploring the space of all Gs
  - Score the Gs

Choosing the Best Bayesian Network: Model Discrimination

- Many Bayesian Networks may model given data well
- In addition to the data fitting part, here we need to discriminate between the many models that fit the data
- Scoring function: Bayesian Likelihood
- More on this later (maybe)

E1: Bayesian Networks and Expression Data

- Friedman et al., 2000
- Learned pronounced features of equivalence classes of Bayesian Networks from time-series measurements of microarray data
- The features are:
  - Order
  - Markov Blanket

Data and Methods

- Data set used: Spellman et al., 1998
  - Objective: Cell cycling genes
  - Yeast genome microarrays (6177 genes)
  - 76 observations at different time-points
- They ended up using 800 genes (250 for some experiments)
- Learned features with both the multinomial and the linear gaussian probability models
- They used no prior knowledge, only the data

Robustness Analysis: Thresholds

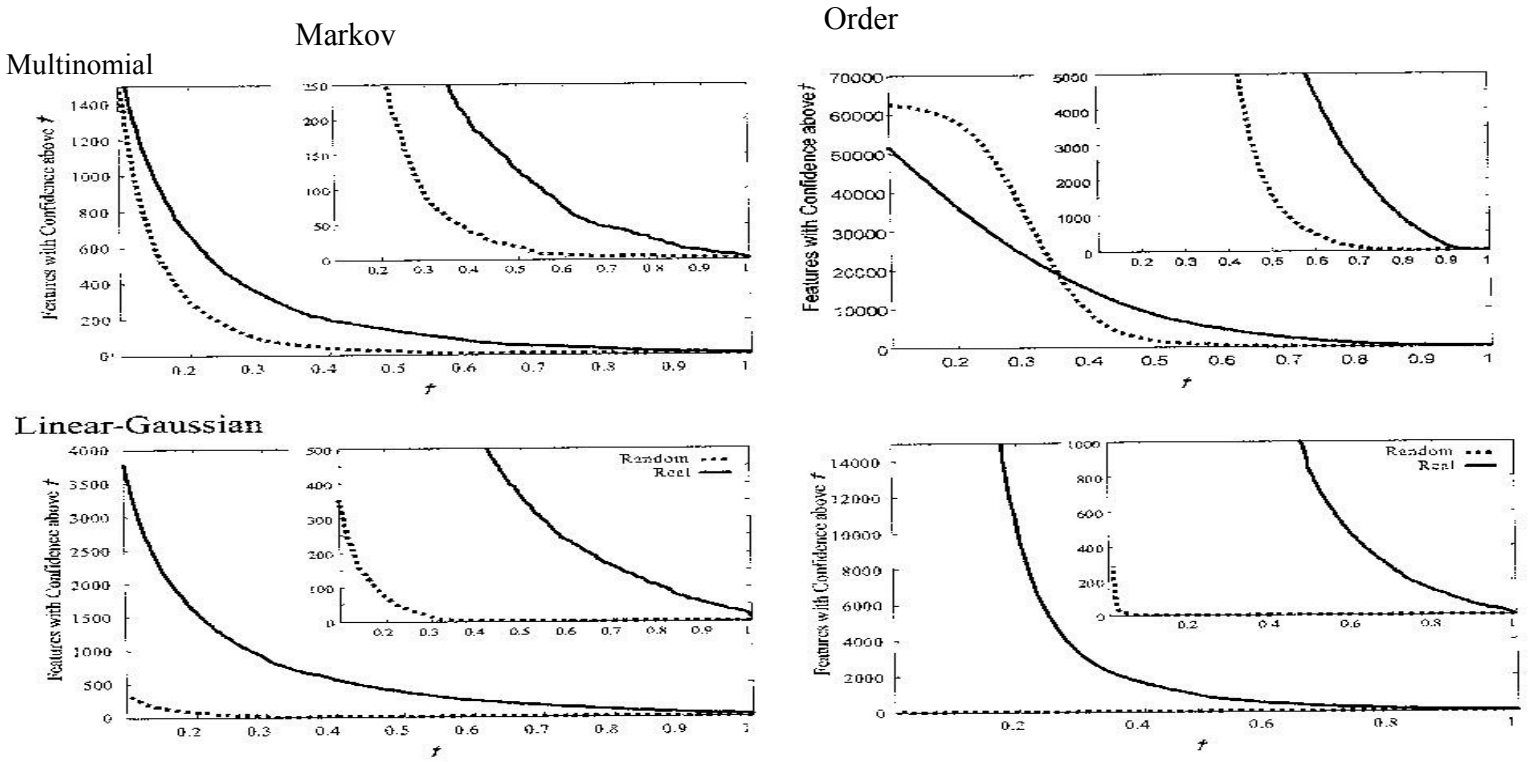


Figure 3: Plots of abundance of features with different confidence levels for the cell cycle data set (solid line), and the randomized data set (dotted line). The x-axis denotes the confidence threshold, and the y-axis denotes the number of features with confidence equal or higher than the corresponding x-value. The graphs on the left column show Markov features, and the ones on the right column show Order features. The top row describes features found using the multinomial model, and the bottom row describes features found by the linear-Gaussian model. Inset in each graph is plot of the tail of the distribution.

Comparing results on real vs. random data

Robustness: Scaling

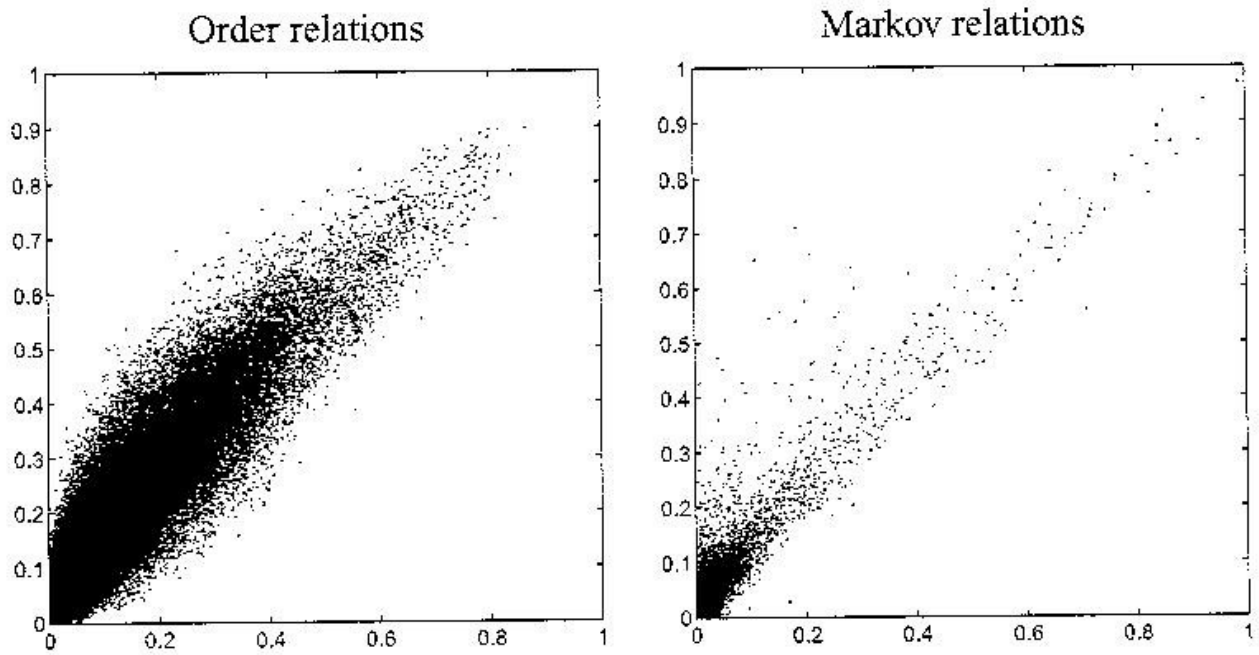


Figure 4: Comparison of confidence levels obtained in two datasets differing in the number of genes, on the multinomial experiment. Each relation is shown as a point, with the x-coordinate being its confidence in the the 250 genes data set and the y-coordinate the confidence in the 800 genes data set. The left figure shows order relation features, and the right figure shows Markov relation features.

Robustness: Discrete vs. Linear

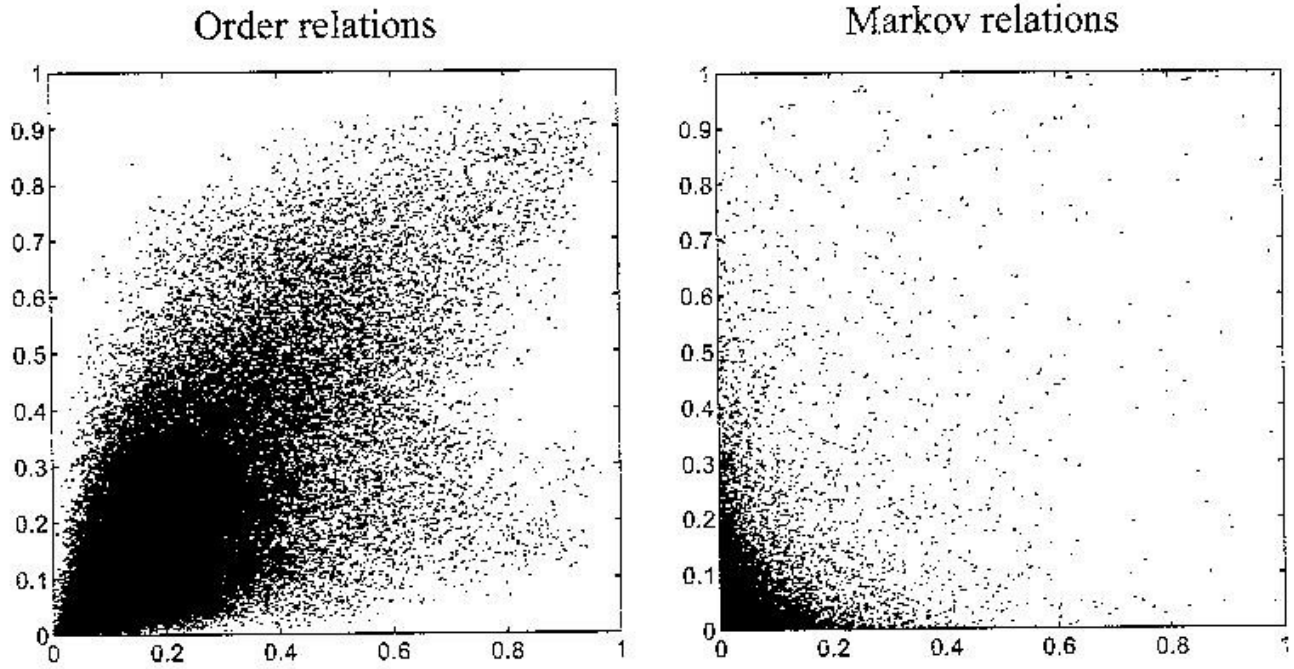


Figure 5: Comparison of of confidence levels between the multinomial experiment and the linear-Gaussian experiment. Each relation is shown as a point, with the x-coordinate being its confidence in the multinomial experiment, and the y-coordinate its confidence in the linear-Gaussian experiment. The left figure shows order relation features, and the right figure shows Markov relation features.

Biological Analysis

- Order relations and Markov relations yield different significant pairs of genes
- Order relations: strikingly pronounced dominant genes, many with interesting known (or even key) properties for cell functions
- Markov relations: all top pairs of known importance, some found that are beyond the reach of clustering (see CLN2 fig.2 for example)

Table 1: List of dominant genes in the ordering relations. Included are the top 10 dominant genes for each experiments.

Gene/ORF	Score in Experiment Multinomial	Gaussian	Notes
MCD1	550	525	Mitotic Chromosome Determinant,null mutant is inviable
MSH6	292	508	Required for mismatch repair in mitosis and meiosis
CSI2	444	497	cell wall maintenance, chitin synthesis
CLN2	497	454	Role in cell cycle START, null mutant exhibits G1 arrest
YLR183C	551	448	Contains forkheaded associated domain, thus possibly nuclear
RFA2	456	423	Involved in nucleotide excision repair, null mutant is inviable
RSR1	352	395	GTP-binding protein of the RAS family involved in bud site selection
CDC45	-	394	Required for initiation of chromosomal replication, null mutant lethal
RAD53	60	383	Cell cycle control, checkpoint function, null mutant lethal
CDC5	209	353	Cell cycle control, required for exit from mitosis, null mutant lethal
POL30	376	321	Required for DNA replication and repair, null mutant is inviable
YOX1	400	291	Homeodomain protein
SRO4	463	239	Involved in cellular polarization during budding
CLN1	324	-	Role in cell cycle START, null mutant exhibits G1 arrest
YBR089W	298	-	

Table 2: List of top Markov relations, multinomial experiment.

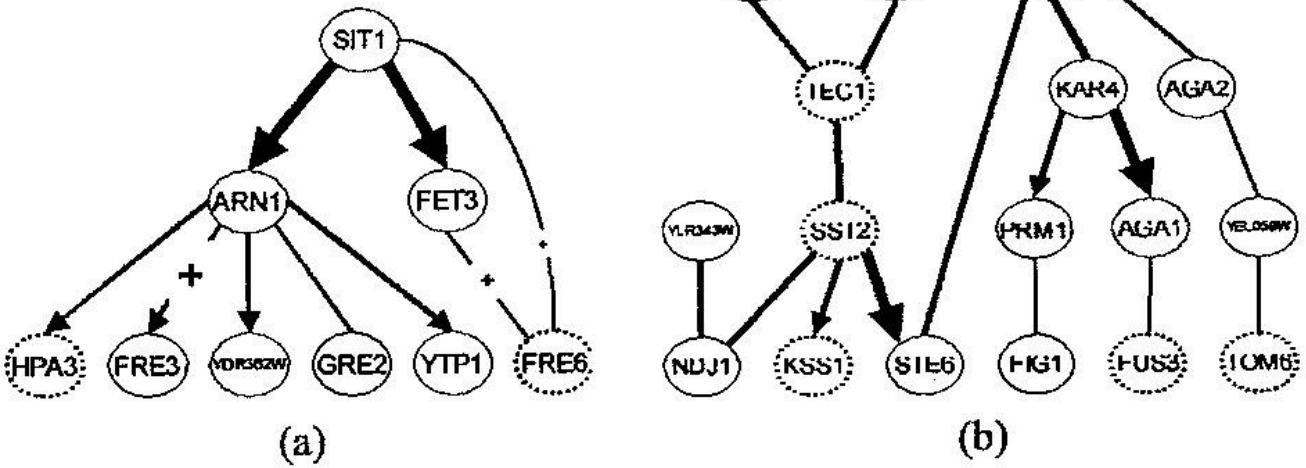
Confidence	Gene 1	Gene 2	Notes
1.0	YKL163W-PIR3	YKL164C-PIR1	Close locality on chromosome
0.985	PRY2	YKR012C	Close locality on chromosome
0.985	MCD1	MSH6	Both bind to DNA during mitosis
0.98	PHO11	PHO12	Both nearly identical acid phosphatases
0.975	HHT1	HTB1	Both are Histones
0.97	HTB2	HTA1	Both are Histones
0.94	YNL057W	YNL058C	Close locality on chromosome
0.94	YHR143W	CTS1	Homolog to EGT2 cell wall control, both involved in Cytokinesis
0.92	YOR263C	YOR264W	Close locality on chromosome
0.91	YGR086	SIC1	Homolog to mammalian nuclear ran protein, both involved in nuclear function
0.9	FAR1	ASH1	Both part of a mating type switch, **expression uncorrelated**
0.89	CLN2	SVS1	Function of SVS1 unknown
0.88	YDR033W	NCE2	Homolog to transmembrame proteins suggest 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

Ex2: Bayesian Networks and Perturbation Data

- Pe'er et al. 2001
- Similar study as above, but on a different, and bigger data set.
- Goal: identify network AND nature of interactions
- Data: Hughes et al. 2000
  - 6000+ genes in yeast
  - 300 full-genome perturbation experiments
    - 276 deletion mutants
    - 11 tetracycline regulatable alleles of essential genes
    - 13 chemically treated yeast cultures
- Pe'er et al. chose 565 significantly differentially expressed genes in at least 4 profiles

Results

- Biologically meaningful pathways learned from the data!



Iron homeostasis

Mating response

Read the paper.....

Limitations

- Bayesian Networks:
  - Causal vs. Bayesian Networks
  - What are the edges really telling us?
  - Dependent on choice of priors
  - Simplifications at every stage of the pipeline: analysis impossible
- Friedman et al. approach:
  - They did what they knew how to do: priors and other things chosen for convenience
  - Meaningful biology?
  - Do we need all that machinery if what they discovered are only the very strong signals?

Background:

1. Bayes Logic
2. Graphs
3. Bayes Nets
4. Learning Bayes Nets
5. BN and Causal Nets
6. BN and Regulatory Networks

BONUS

1. Bayes Logic

If two events, A and B are independent:

$$P(AB)=P(A)P(B)$$

If they are not independent:

$$P(B|A)=P(AB)/P(A)$$

or

$$P(AB)=P(B|A)*P(A)$$

Bayes Formula

- $P(AB)=P(B|A)*P(A)$

Joint distributions

- $P(AB)=P(A|B)*P(B)$

Thus $P(B|A)*P(A)=P(A|B)*P(B)$ and

$$P(B|A)=P(A|B)*P(B)/P(A)$$

If B_i, $i=1,...,n$ are mutually exclusive events, then

Possible causes

Posterior $P(B_i|A) = \frac{P(A|B_i)\cdot P(B_i)}{\sum_{j=1}^{n} P(A|B_j)\cdot P(B_j)}$ Prior probabilities (assumptions)

Observation

Joint Probability

- The probability of all events:
  $P(AB)=P(A)*P(B|A)$ *or*
  $P(ABCD)=P(A)*P(B|A)*P(C|AB)*P(D|ABC)$
  …

- For n variables it can take up to 2^n terms to write it out!

Conditional Independencies

- Recall P(AB)=P(A)*P(B) if A is independent of B
- Similarly, we have the conditional Independence: A is independent of B, given C

  P(A;B|C) =P(A|C)*P(B|C)

2. Graphs

- $\boldsymbol{X}=\{X_1, X_2, ..., X_n\}$ – the set of nodes
- Pa(Xi) – the set of parents of node Xi
- Des(Xi) $\subseteq \boldsymbol{X}$– the set of descendant nodes Y s.t. there is a directed path from Xi to Y
- Xi is an ancestor to all Des(Xi)
- NonDes(Xi) $\subseteq \boldsymbol{X}$ – non descendant nodes of Xi, i.e. $\boldsymbol{X}$\Des(Xi)
- Note that all ancestors of Xi are also in NonDes(Xi)

3. Bayesian Nets (BNs)
(G,θ)

- BNs Encode Joint Prob. Distribution on all nodes
- The joint distribution follows directly from the graph
- **Markov Assumption:** Each variable is independent of its non-descendents, given its parents
- Bayesian Networks implicitly encode the Markov assumption.

The joint probability can be expanded by the Bayes chain rule as follows:

$$P(X_1, X_2, \ldots, X_n) = P(X_n \mid X_1, X_2, \ldots, X_{n-1}) \times P(X_1, X_2, \ldots, X_{n-1})$$

$$= P(X_n \mid X_1, X_2, \ldots, X_{n-1}) \times P(X_{n-1} \mid X_1, X_2, \ldots, X_{n-2}) \times P(X_1, X_2, \ldots, X_{n-2}) = \ldots$$

$$= \prod_{i=1}^{n} P(X_i \mid X_1, X_2, \ldots, X_{i-1})$$

Let $X_1, X_2, ..., X_n$ be topologically sorted, i.e. Xi is before all its children. Then, the joint probability becomes:

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid x_1, x_2, ..., x_{i-1}) = \prod_{i=1}^{n} P(x_i \mid Pa(x))$$

which is what the joint distribution simplifies to.

Notice that if the parents (fan in) are bound by k, the complexity of this joint becomes $n2^{k+1}$

4. Learning Bayesian Networks

- Problem: Given a training set $D=(x^1,x^2,...,x^n)$ of independent instances of the random variables $(X_1,X_2,...,X_n)$, find a network G (or equivalence class of networks) that best matches D.

Equivalence Classes of Bayesian Networks

- A Bayesian Network G implies a set of independencies, I(G), in addition to the ones following from Markov assumption
- Two Bayesian Networks that have the same set of independencies are equivalent
- Example G: $X \rightarrow Y$ and G': $X \leftarrow Y$ are equivalent, since $I(G)=I(G')=\varnothing$

Equivalence Classes, contd.

- v-structure: two directed edges converging into the same node, i.e. $X \rightarrow Z \leftarrow Y$
- Thm: Two graphs are equivalent iff their DAGs have the same underlying undirected graphs and the same v-structures
- Graphs in an equivalence class can be represented simply by Partially Directed Graph, PDAG where
  - a directed edge, $X \rightarrow Y$ implies all members of the equivalence class contain that directed edge
  - an undirected edge, X—Y implies that some DAGs in the class contain $X \rightarrow Y$ and others $X \leftarrow Y$.
- Given a DAG, a PDAG can be constructed efficiently

Model Selection

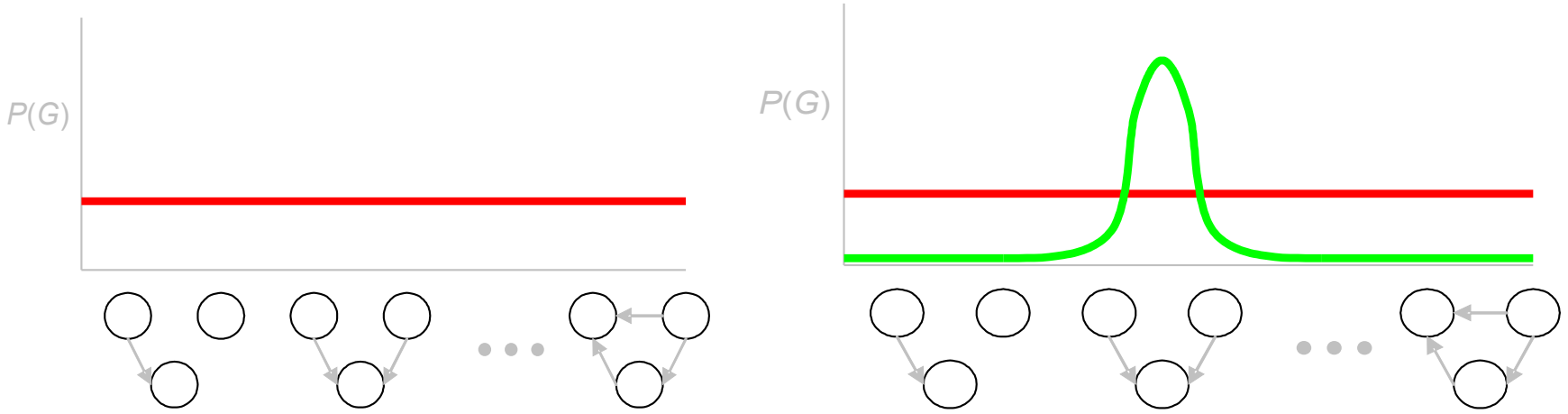
- Propose and compare models for G
- Comparison based on a scoring function
- A commonly used scoring function is the Bayesian Score which has some very nice properties.
- Finding G that maximizes the Bayesian Score is NP-hard; heuristics are used that perform well in practice.

Scoring Bayesian Networks

- The Scoring Measure is the posterior probability of the graph, given the data ($D=\{x^1,...,x^n\}$:

$$S(G:D)=\log P(G|D)=\log P(D|G)+\log P(G)+C$$

- *P(G) (resp. P(D|G))* averages the probability of the data over all possible Gs (resp. over all possible parametric assignments to G)



Example for maximizing P(G) (from David Danks, IHMC)

Properties of a Well-Chosen S(G:D)

$$P(D \mid G) = \int P(D \mid G, \theta) P(\theta \mid G) d\theta$$

- graphs that capture the exact properties of the network (i.e. all dependencies in the distribution) very likely score higher than ones that do not (given large # of samples)
- Score is decomposable:

$$S(G : D) = \sum_{i=1}^{n} ScoreContribution(Xi, Pa(Xi) : D)$$

Issues in Scoring

- Which metric:
  - Bayesian Dirichlet equivalent: captures P(G|D),
  - Bayesian Information Criterion: approx.
- Which data discretization? Hard, 2,3,4?
- Which Priors?
- Which heuristics?
  - simulated annealing
  - hill climbing
  - GA, etc.

Optimizing S(G:D)

- Once the priors are specified, and the data is given, the Bayesian Network is learned, i.e. the network with the highest score is chosen
- But Maximizing this scoring function is an NP-hard problem
- Heuristics: local search of the space of all Gs by adding/subtracting edges, and reversing directions

5. Closer to the Goal: Causal Networks

Bayesian vs. Causal Nets

1. We want "A is a cause for B" (Causal)
2. We have "B independent of non-descendants given A" (Bayesian)
3. So, we want to get from the second to the first, i.e. from Bayesian to stronger, causal networks

Difference between Causal and Bayesian Networks:

- $X \rightarrow Y$ and $X \leftarrow Y$ are equivalent Bayesian Nets, but very different causally
- Causal Networks can be interpreted as Bayesian if we make another assumption
- Causal Markov Assumption: given the values of a variable's immediate causes, it is independent of its earlier causes (Example: Genetic Pedigree)
- Rule of thumb: In a PDAG equivalence class, $X \rightarrow Y$ can be interpreted as a causal link

6. Putting it All Together: BNs and Regulatory Networks

Spellman et al., 2000

Pe'er et al. 2001

How do We Use BNs for Microarray Data?

- Random Variables denote expression levels of genes
- The result is a joint probability distribution over all random variables
- The joint can be used to answer queries:
  - Does the gene depend on the experimental conditions?
  - Is this dependence direct or not?
  - If it is indirect, which genes mediate the dependence?

Putting it all Together: Issues

In learning such models the following issues come up:

1. Dimensionality curse: statistical robustness
2. Algorithmic complexities in learning from the data
3. Choice of local probability models (priors)

Dimensionality Curse

Problem: We are hurt by having many more genes than observations (6200 vs. tens or hundreds)

Solution:

- **Bootstrap:** features robust to perturbations
- **Partial Models:** features present in many models
- Combine the two

Partial Models

- Instead of trying to learn a model that explains the whole data characterize features common to high-scoring models
- The intuition is that preserved features in many high-scoring networks are biologically important
- Simple features considered: pair-wise relations

Partial Model Features

- Markov Relations
  - Is Y in the Markov Blanket of X?
  - Markov Blanket is the minimal set of variables that shield X from the rest of the variables in the model
  - Formally, X is independent from the rest of the network given the blanket
  - It can be shown that X and Y are either directly linked or share parenthood of a node
  - In biological context, a MR indicates that X and Y are related in some joint process
- Order Relations
  - Is X an ancestor of Y in all networks of a given class?
  - An indication of causality!

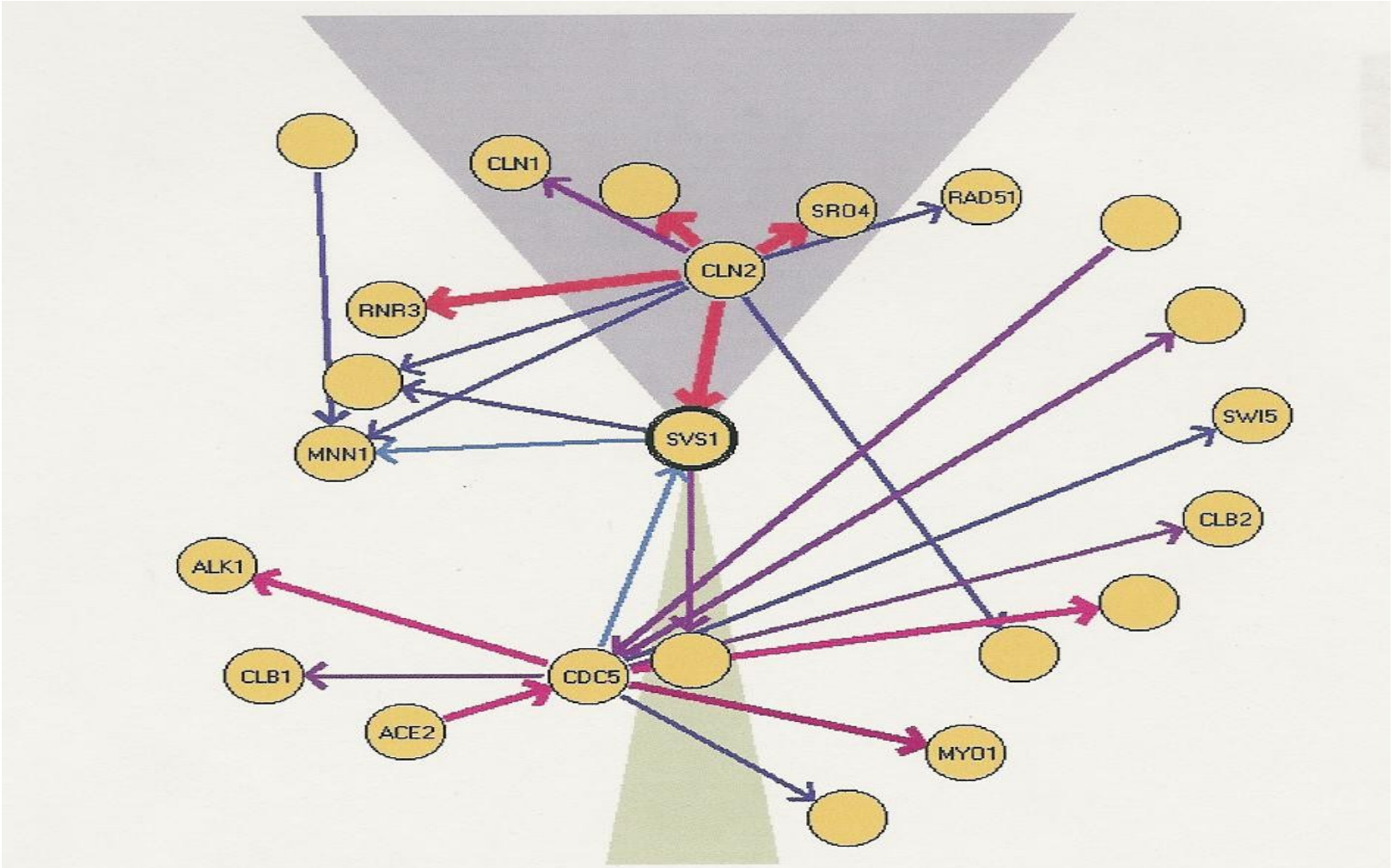


Figure 2: An example of the graphical display of Markov features. This graph shows a "local map" for the gene SVS1. The width (and color) of edges corresponds to the computed confidence level. An edge is directed if there is a sufficiently high confidence in the order between the genes connected by the edge. This local map shows that CLN2 separates SVS1 from several other genes. Although there is a strong connection between CLN2 to all these genes, there are no other edges connecting them. This indicates that, with high confidence, these genes are conditionally independent given the expression level of CLN2.

Bootstrap: Are the Features Trustworthy?

- To what extent does the data support a given feature?
- The authors develop a measure of confidence in features as the likelihood that a given feature is actually true
- Confidence is estimated by generating slightly "perturbed" versions of the original data set and learning from them
- Thus, any false positives should disappear if the features are truly strong
- This is the case in their experiments

Efficient Learning Algorithms

- The solution space for all these problems is huge: super-exponential
- Thus some additional simplification is needed
- Assumption: Number of parents of a node is limited
- Trick: Initial guesses for the parents of a node are genes whose temporal curves cluster well

Local Probability Models

- Multinomial and linear gaussian
- These models are chosen for mathematical convenience
- Pros. et cons.:
  - Former needs discretization of the data. Gene expression levels are {-1,0,1}. Can capture combinatorial effects
  - Latter can take continuous data, but can only detect linear or close to linear dependencies

References

- Akutsu et al., *Identification of Genetic Networks From a Small Number of Gene Expression Patterns Under the Boolean Network Model*, Pacific Symposium on Biocomputing, 1999
- Akutsu et al., *Algorithms for Identifying Boolean Networks and Related Biological Networks Based on Matrix Multiplication and Fingerprint Function*, RECOMB 2000
- Liang et al., *REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures*, Pacific Symposium on Biocomputing, 1998
- Wuensche, *Genomic Regulation Modeled as a Network With Basins of Attraction*, Pacific Symposium on Biocomputing, 1998
- D'Haeseleer et al., *Tutorial on Gene Expression, Data Analysis, and Modeling*, PSB, 1999
- Chen et al, *Identifying gene regulatory networks from experimental data*, RECOMB 1999
- Wagner, *How to reconstruct a large genetic network of n genes in n^2 easy steps*, Bioinformatics, 2001

References:

- Friedman et al., Using Bayesian Networks to Analyze Expression Data, RECOMB 2000, 127-135.
- Pe'er et al., Inferring Subnetworks from Perturbed Expression Profiles, Bioinformatics, v.1, 2001, 1-9.
- Ron Shamir's course, Analysis of Gene Expression Data, DNA Chips and Gene Networks, at Tel Aviv University, lecture 10 http://www.math.tau.ac.il/~rshamir/ge/02/ge02.html
- Yu, J. et al. "Using Bayesian Network Inference Algorithms to Recover Molecular Genetic Regulatory Networks." ICSB02.
- Spellman et al., Mol. Bio. Cell, v. 9, 3273-3297, 1998
- Hughes et al., Cell, v. 102, 109-26, 2000