Lecture 9

- Static Graph Models, continued
  - Parsimony arguments: is nature optimal?
  - (Chen et al, 1999) # Regulators is small
    - Optimizing a function: simulated annealing
  - Can we capture regulatory relationships well with correlation arguments?
  - (Wagner, 2002) # Relationships is minimal
  - Direct vs. indirect relationships
  - A perturbation model to detect direct relationships
- Linear Models
  - Definition
  - Caculating the Next State
  - Reverse Engineering the Parameters from Data
  - Normalization
  - Properties
  - Data Requirements

Simulated Annealing

- 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

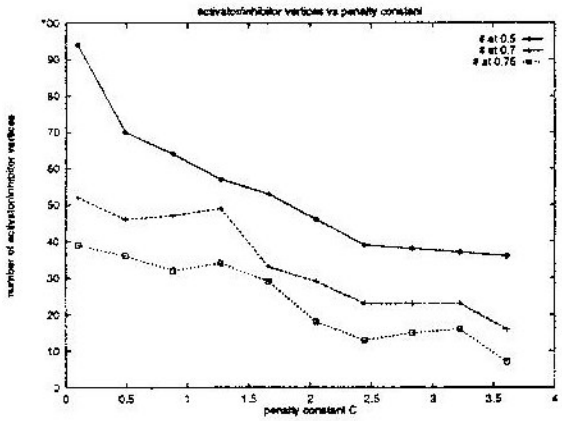
Algorithm: Given f(x) and x

1. Initialize temperature to T
2. DO: generate x′, a random transition from x
3. Calculate $\Delta f=f(x')-f(x)$
4. If $\Delta f<0$, accept x′(i.e. x=x′)
5. Else
   - accept x′ with $P = \exp(-\Delta f/T)$
   - (reject x′ with 1−P)
6. Update T, $T=\alpha T$, $\alpha=1-\varepsilon$
7. UNTIL (2) Δf converges

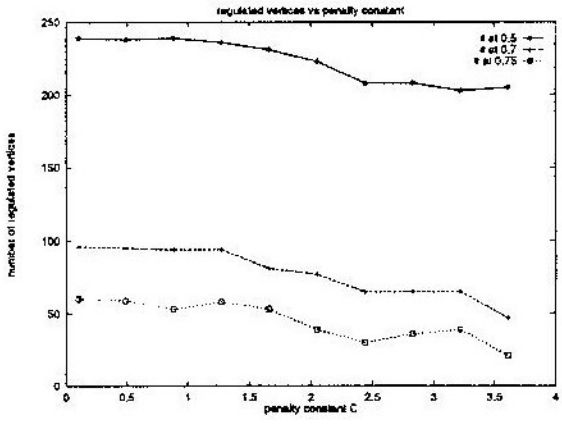
Results (Chen et al, 1999)

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

Simulated annealing performed hundreds of times for different cutoffs in edge strength and penalty constant

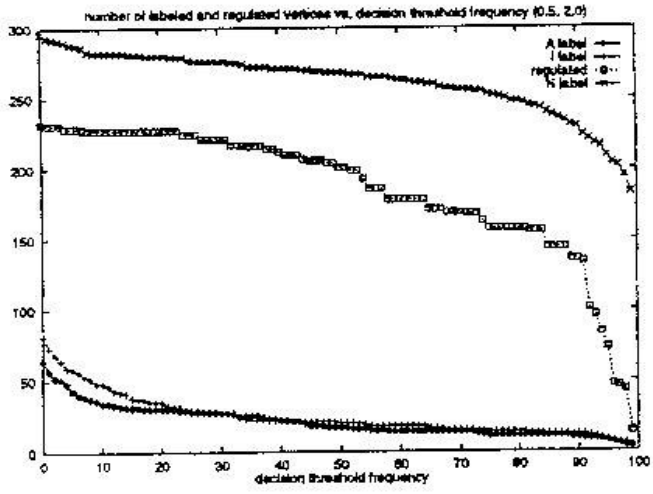


Total number of activator+inhibitor nodes vs. the penalty constant (for different edge strengths)



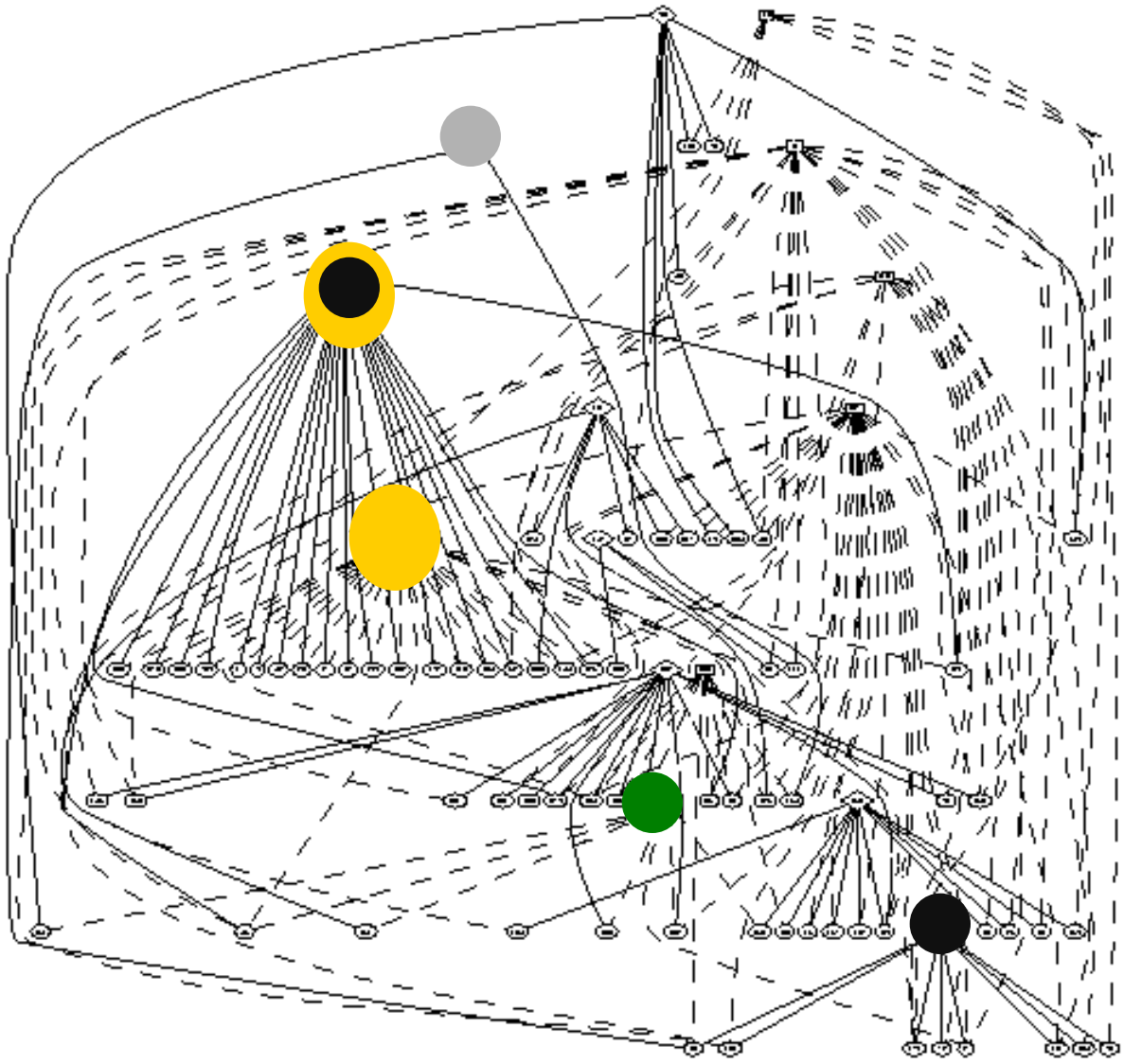
Total number of regulated nodes (out of 308)

For C=2 and cutoff=0.5, to the right is the % of consistent assignments in 100 SA runs



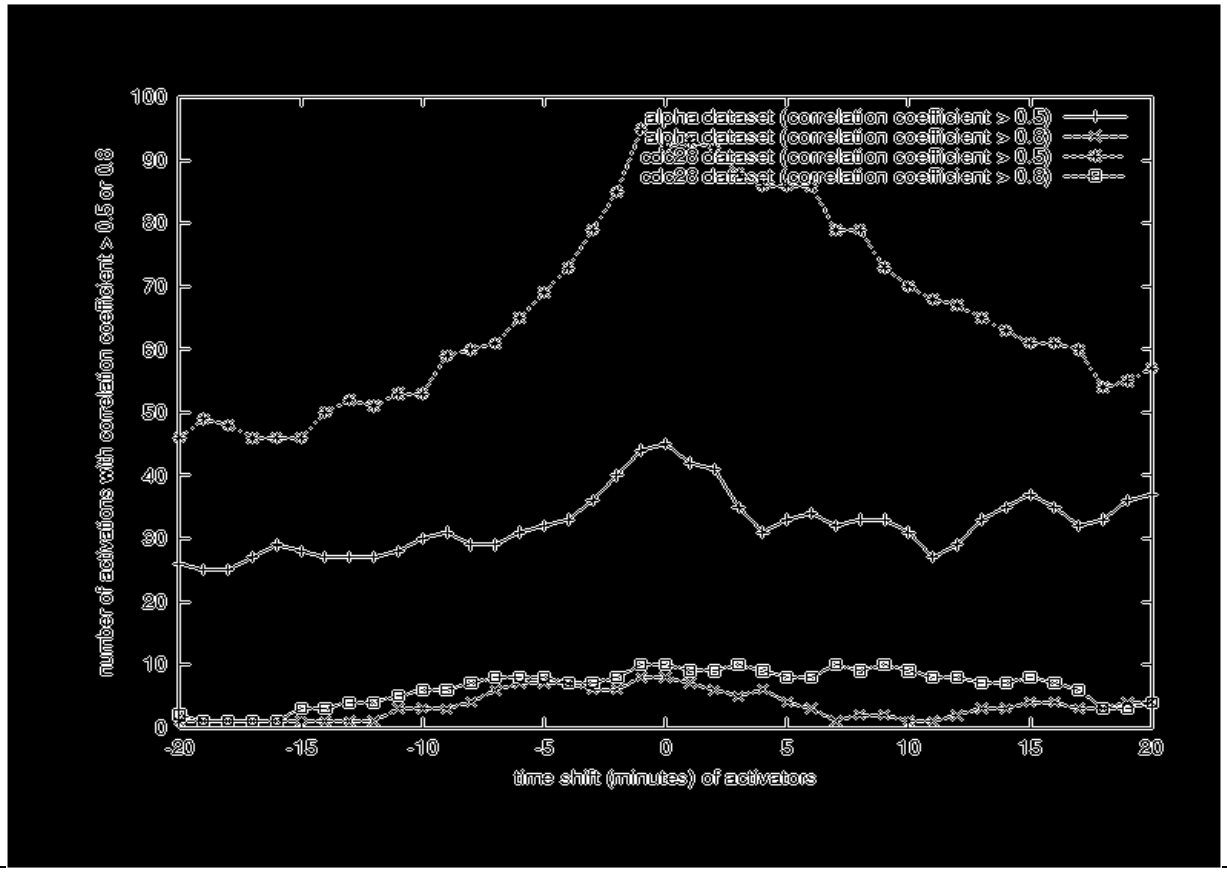
Results (Chen et al, 1999)

- A candidate network for C=2, cutoff = 0.5, and p=95%

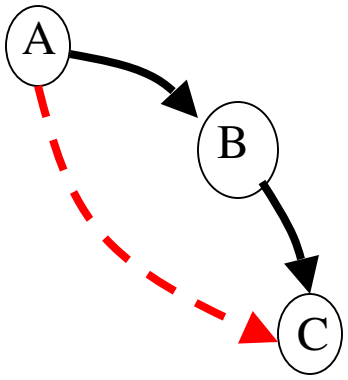


How Well Can We Capture Relationships by Correlation?

- 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)



Direct vs. Indirect Relationships



Direct:
A $\Rightarrow$B
B $\Rightarrow$C

Indirect
A $\Rightarrow$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)

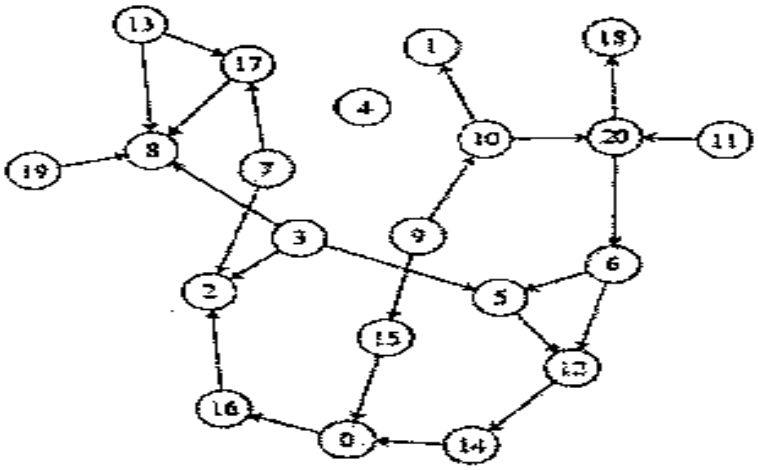
- Motivation: perturbing a gene network one gene at a time and using the effected genes in order to discriminate direct vs. indirect gene-gene relationships
- Perturbations: gene knockouts, over-expression, etc.

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

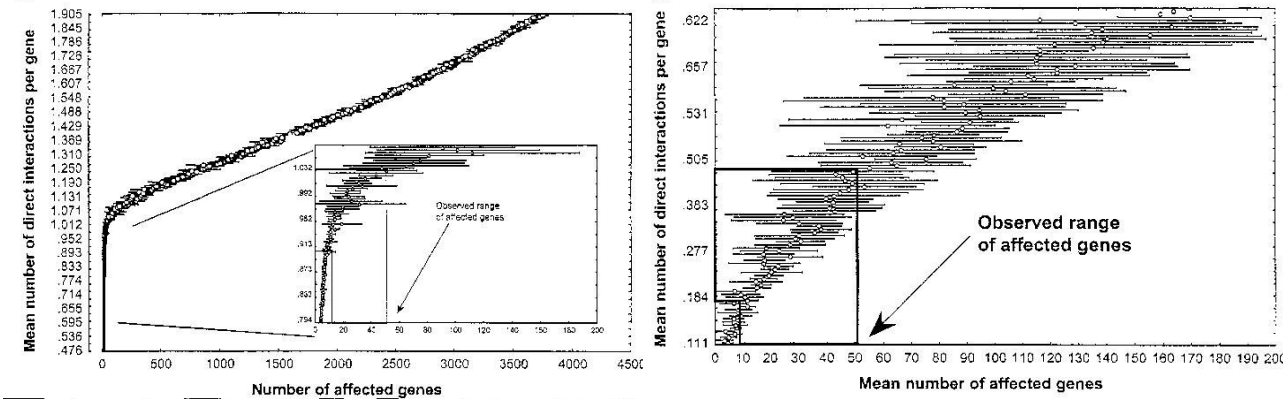
0:	16
1:	
2:	
3:	2 5 8
4:	
5:	12
6:	5 12
7:	2 17
8:	
9:	10 15
10:	1 20
11:	20
12:	14
13:	8 17
14:	0
15:	0
16:	2
17:	8
18:	
19:	8
20:	6 18

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 less if they are scale-free)



Reconstructing the Network

- Third assumption: 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!

Static (Graph) Models

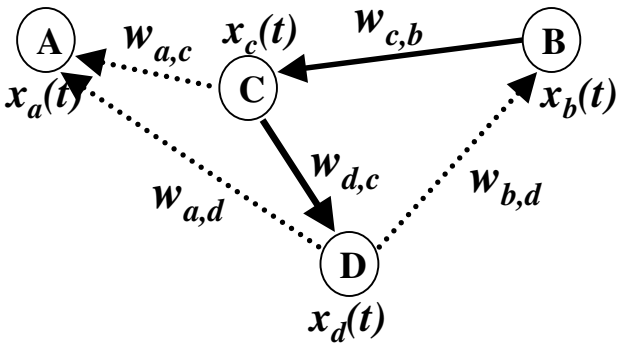
Summary

- 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

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 $\mathbf{y_i}$, $i = 1,...,k+1$, 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 :

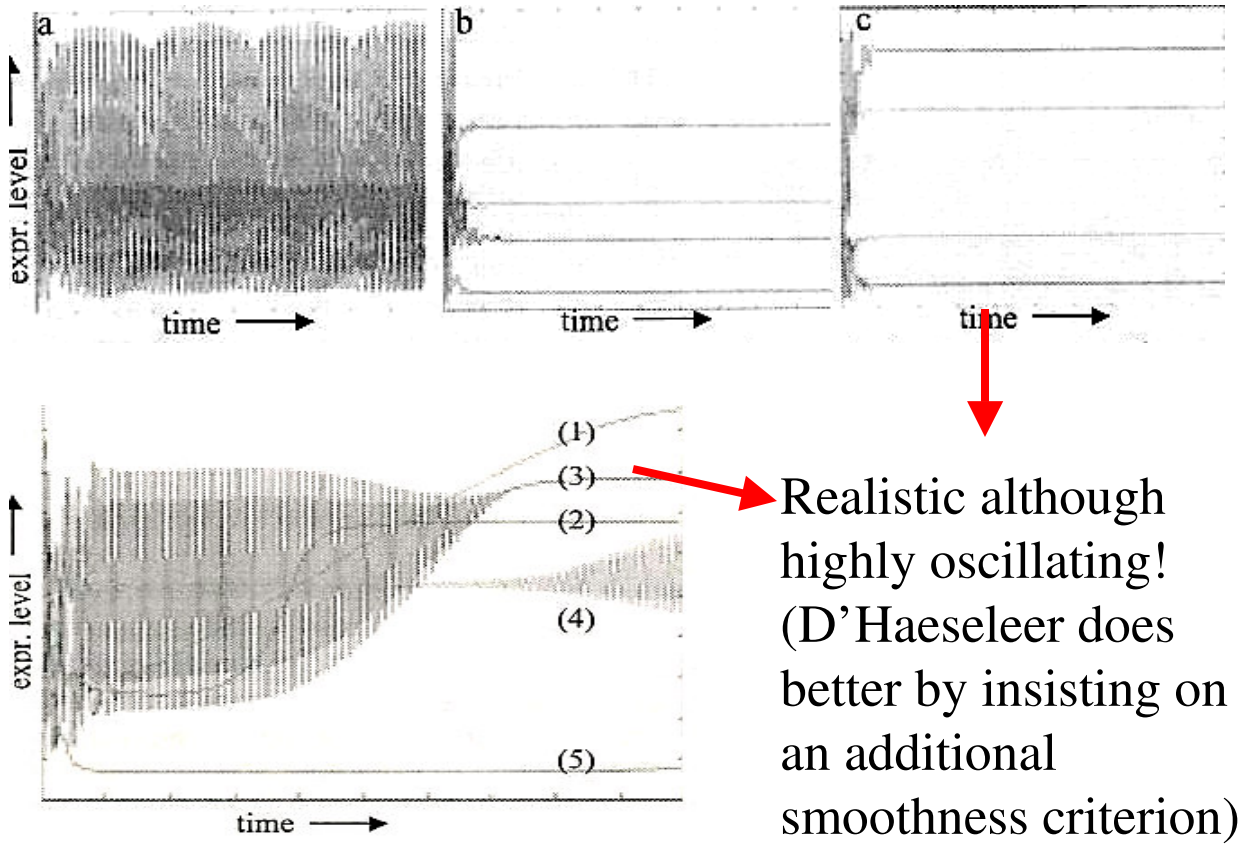
$$\mathbf{W = A^{**}A}, \quad \boxed{\mathbf{A^{**} = A^T (A A^T)^{-1}}}$$

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)

Properties of Linear Models (Weaver et al, 1999)

- Simulating Linear State Models by randomly generating the parameters
- The output of a state was used as input for the next
- The models were iterated until they reached a terminal steady state



Realistic although highly oscillating! (D'Haeseleer does better by insisting on an additional smoothness criterion)

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(K),* than it can be shown that the number of experiments needed is *O(K log(N/K))*

Summary

- Linear models yield good, realistic looking predictions
- The amount of data needed is O(n) experiments, for a fully connected network or O(klog(n/k)) for a k 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)

References

- V. Filkov and S. Skiena and J. Zhi, “Methods for Analysis of Microarray Time-Series Data,” Journal of Computational Biology, v. 9, p. 317-330, 2002
- A. Wagner, “Estimating Coarse Gene Network Structure from Large-Scale Gene Perturbation Data,” Genome Research, v. 12, p. 309-315, 2002
- D. C. Weaver and C. T. Workman and G. D. Stromo, "Modeling regulatory networks with weight matrices," Pacific Symposium on Biocomputing, 1999
- E.P. van Someren and L.F.A. Wessels and M.J.T. Reinders, “Linear Modeling of Genetic Networks from Experimental Data,” Intelligent Systems for Molecular Biology, 2000
- P. D’Haeseleer and X. Wen and S. Fuhrman and R. Somogyi, “Linear Modeling of mRNA Expression Levels During CNS Development and Injury,” Pacific Symposium on Biocomputing, 1999