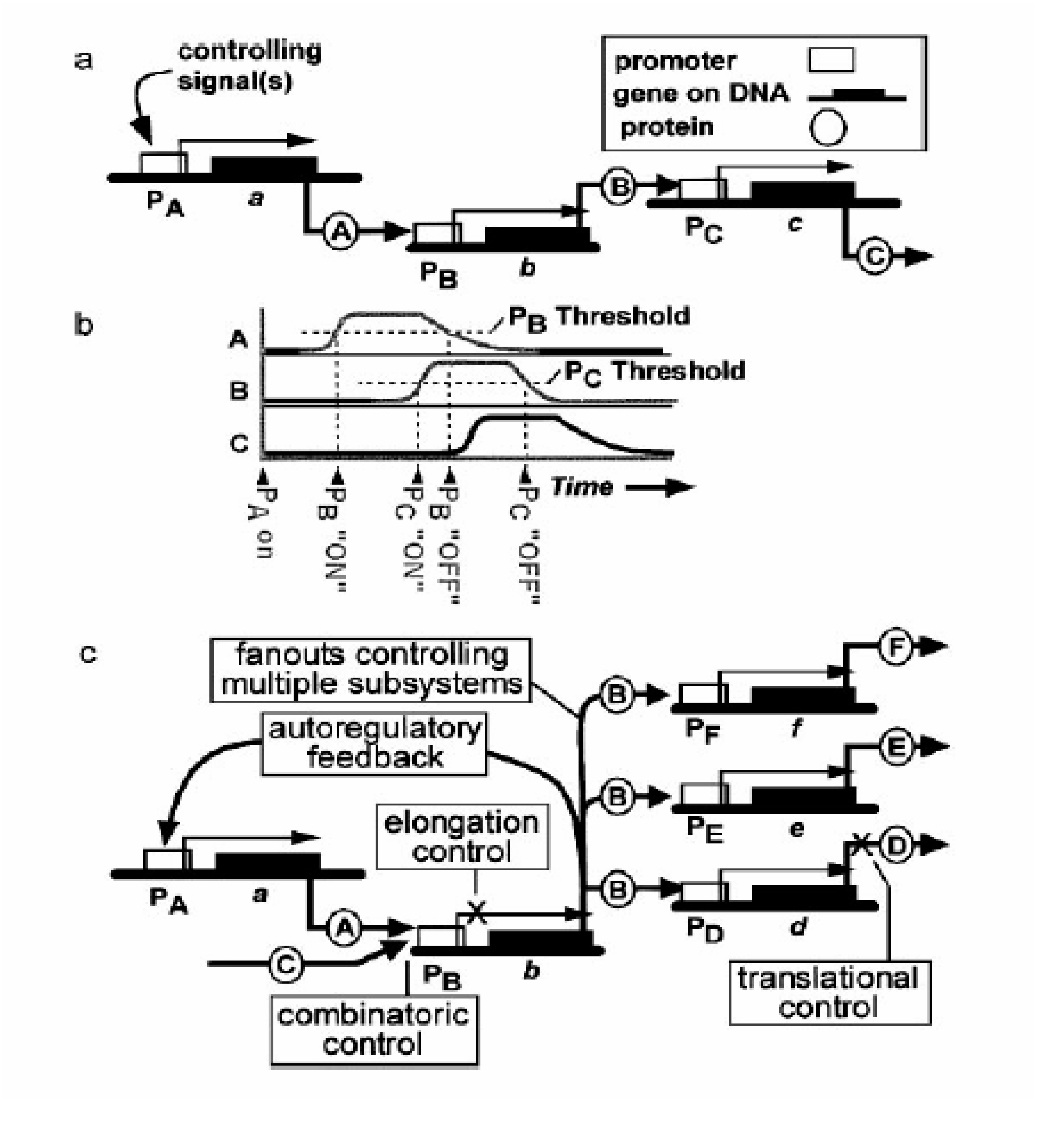
Biological Networks

- Gene Networks
- Metabolic Networks
- Signaling Pathways
- Others …

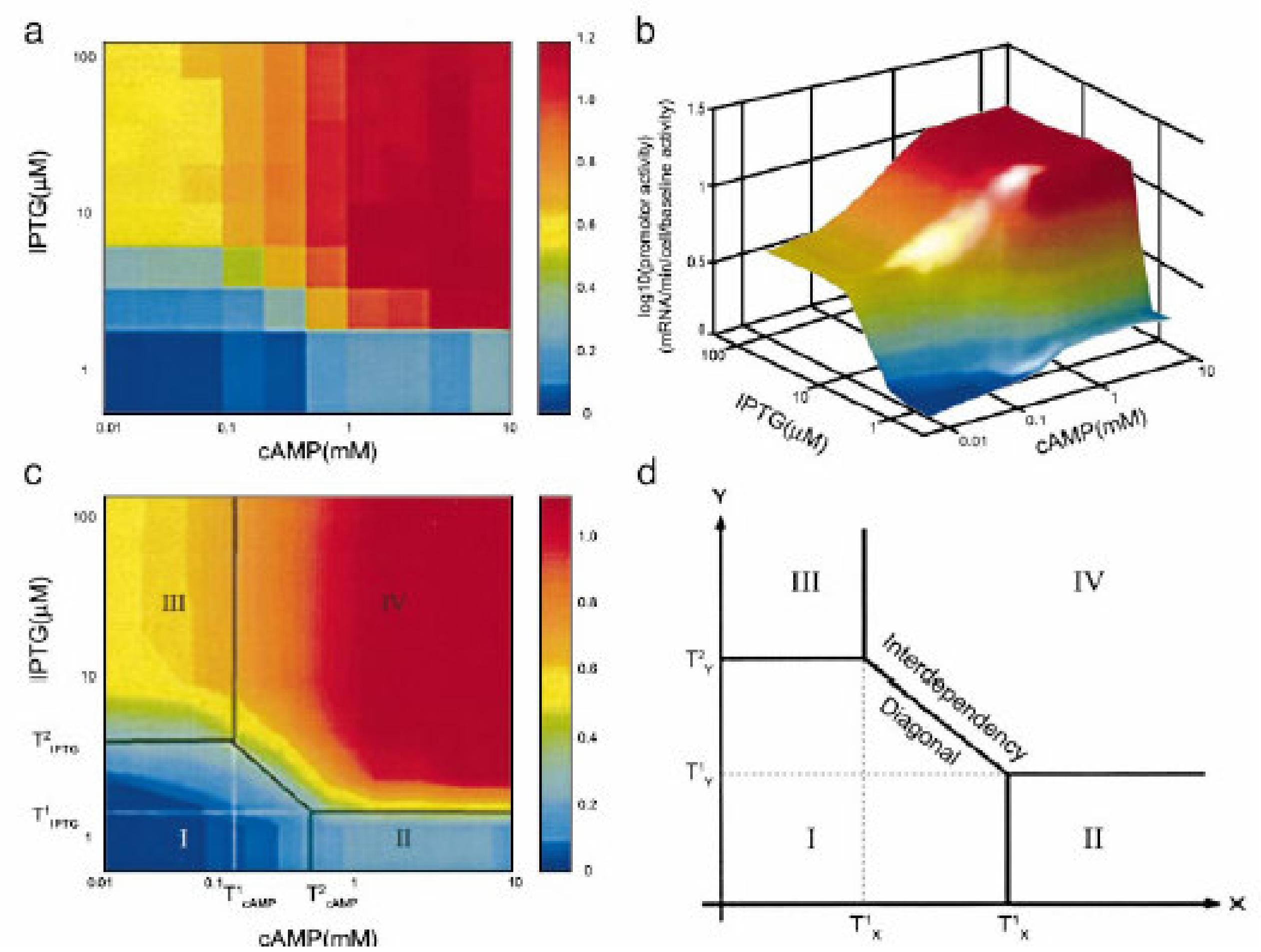
- Modeling
- Inference

Simple Genetic Circuits



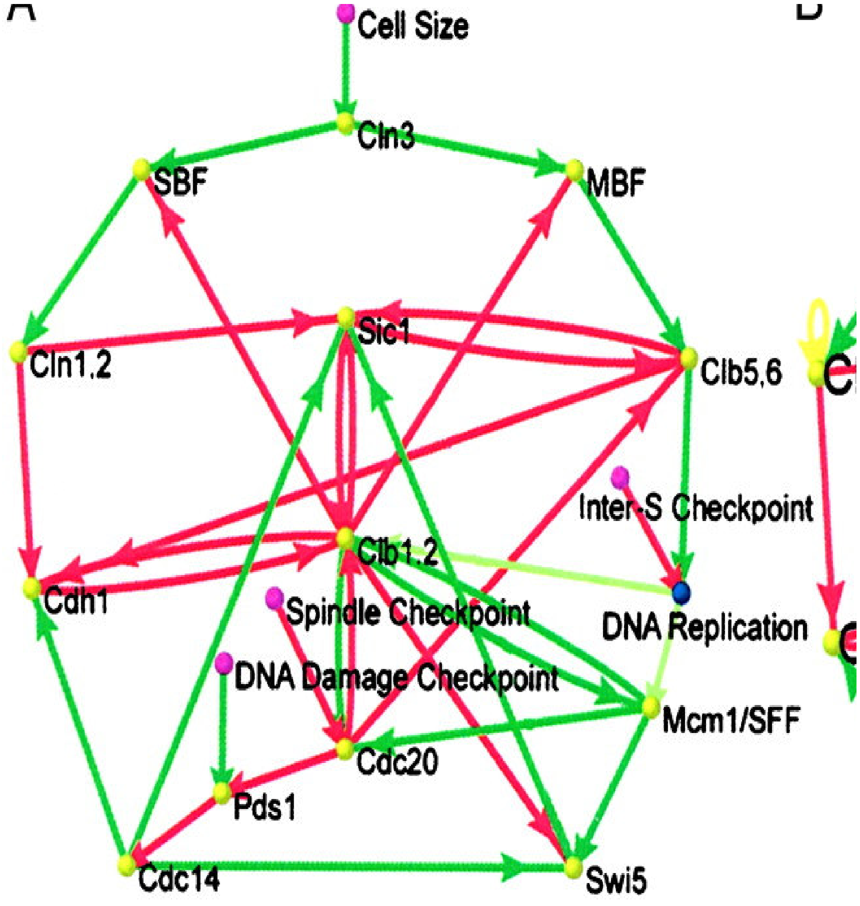
McAdams and Arkin et al 1998

Cis-regulatory input of lacZYA operon in E. coli



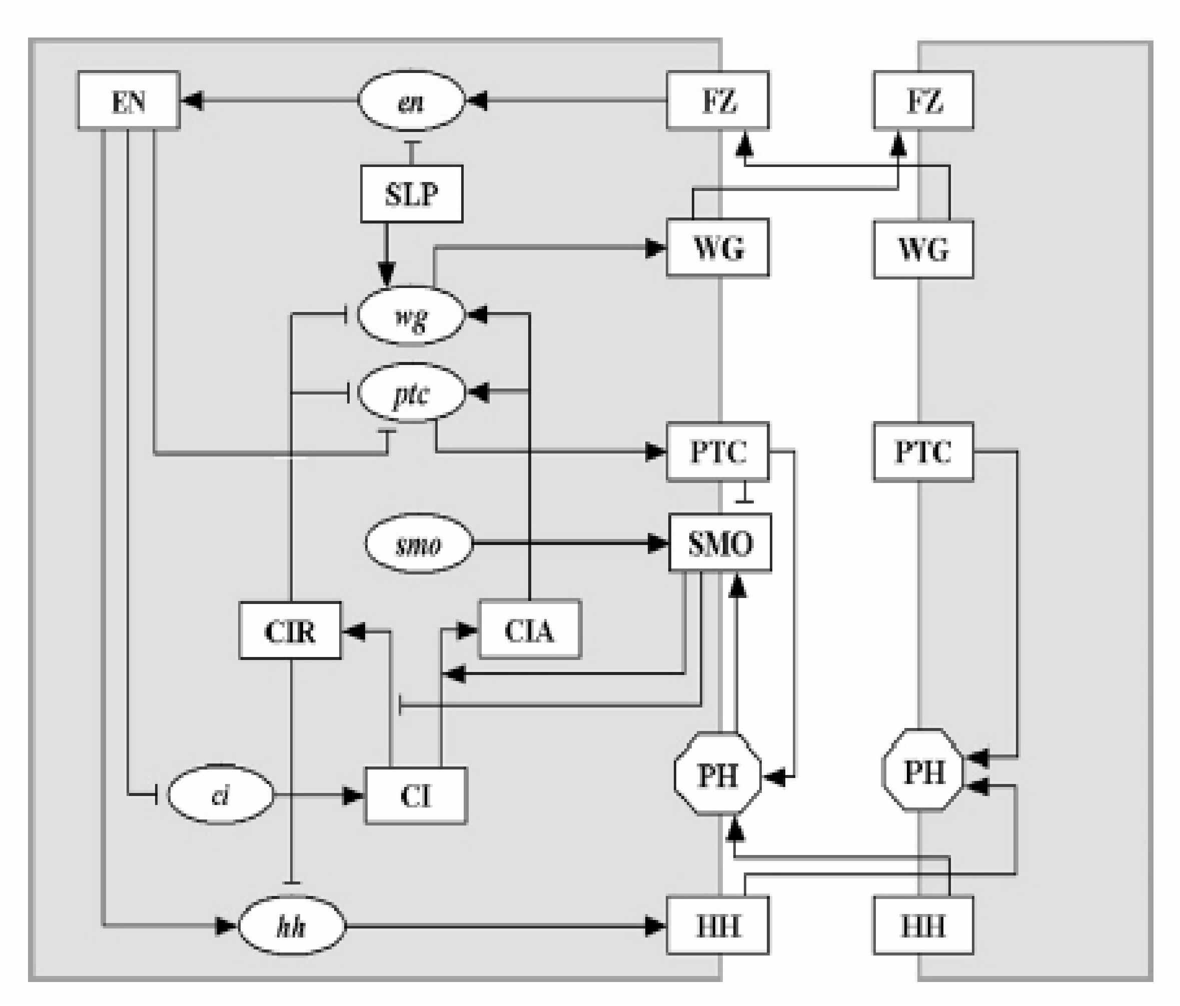
Setty et al. PNAS 2003

Cell cycle network in S. Cerevisiae



Li, Fangting et al. (2004) Proc. Natl. Acad. Sci.

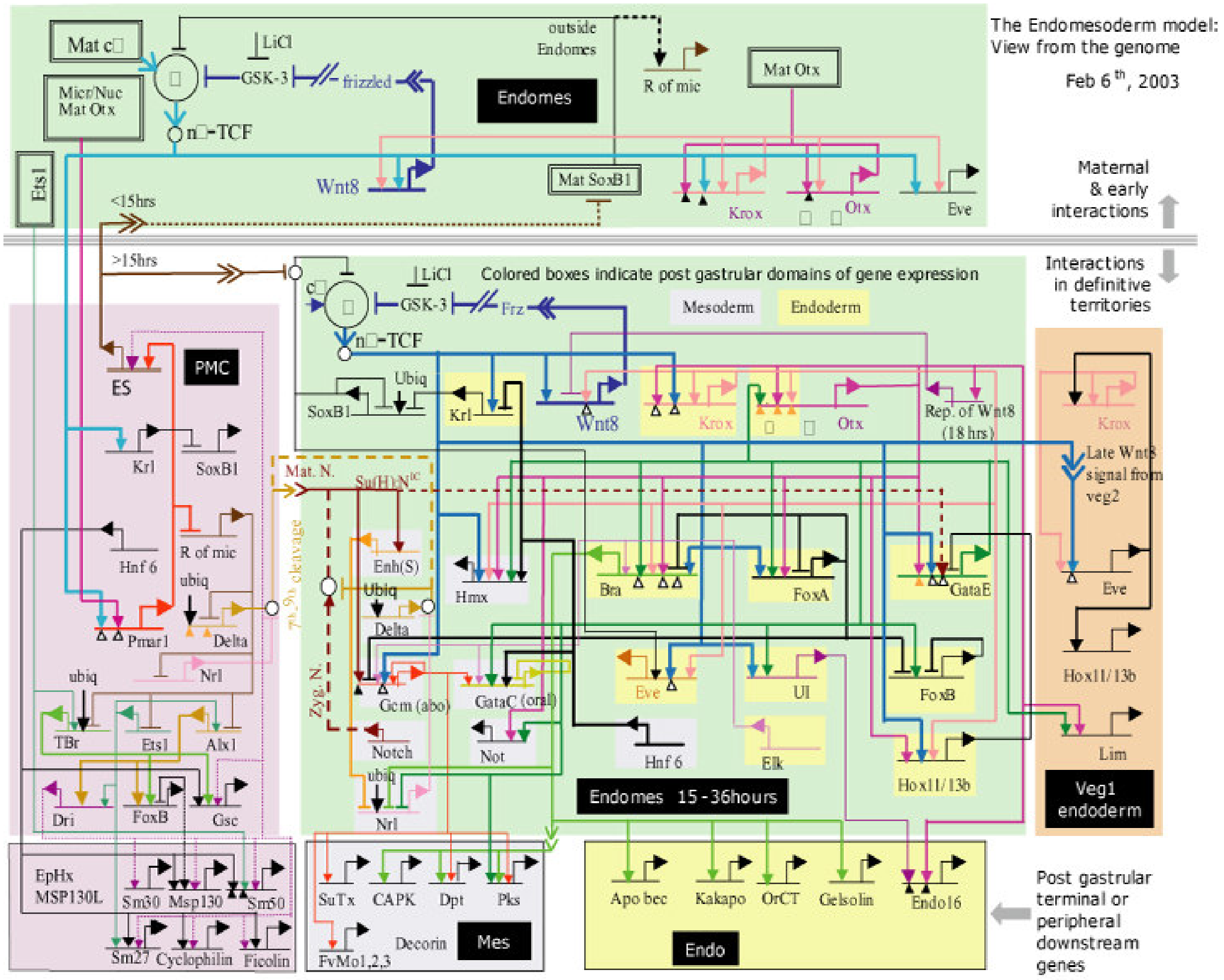
Segment polarity network in Drosophila



Albert and Otmer, JTB 2003

Nodes: mRNA (round), protein (rectangle), prot. Complex (octagon)
Edges: biochemical interactions or regulatory relationships

Gene network of endomeso development in Sea Urchin

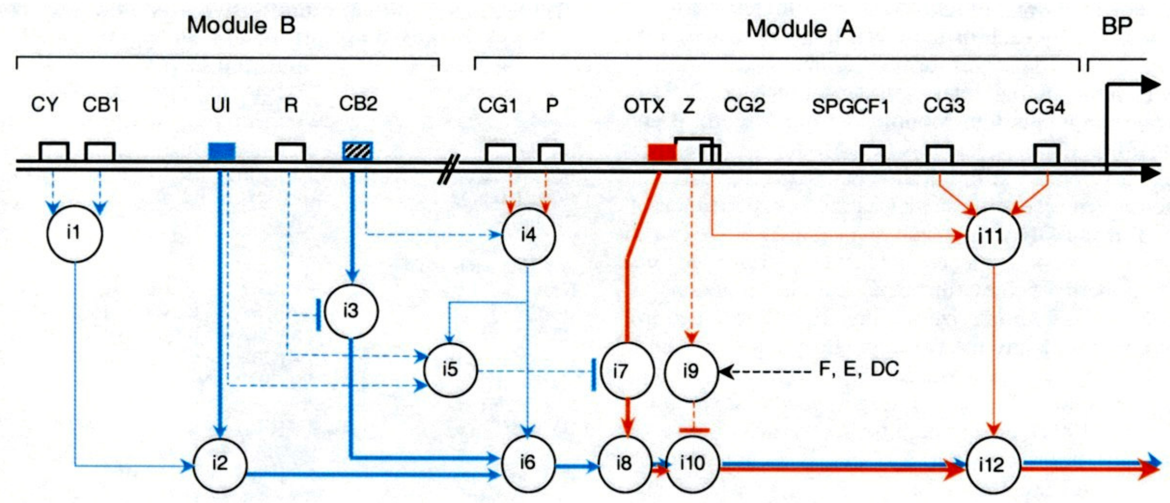


The Endomesoderm model: View from the genome

Feb 6th, 2003

Davidson et al. Science 2002

Logic of Cis-regulation



if CY & CB1	i1 = 1
else	i1 = 0.5
	i2 = i1 • UI (t)
if R	i3 = CB2(t)
else	i3 = k • CB2(t) (1<k<2)
if P & CG1 & CB2	i4 = 2
else	i4 = 0
if UI (t)>threshold & R & i4‡0	i5 = 1
else	i5 = 0
	i6 = i4 • (i2+i3)

if i5=0	i7 = OTX(t)
else	i7 = 0
	i8 = i6 + i7
if (F or E or DC) & Z	i9 = 1
else	i9 = 0
if i9=1	i10 = 0
else	i10 = i8
if (CG2 & CG3 & CG4)	i11 = 2
else	i11 = 1
	i12 = i11•i10

Transcriptional Regulatory Systems

- *Cis regulatory elements*: DNA sequence (specific sites)
    - promoters;
    - enhancers;
    - silencers;
- *Trans regulatory factors*: products of regulatory genes
    - generalized
    - specific (Zinc finger, leucine zipper, etc.)

Known properties of real gene regulatory systems:

- cis-trans specificity
- small number of trans factors to a cis element: 8-10
- cis elements are programs
- regulation is event driven (asynchronous)
- regulation systems are noisy environments
- Protein-DNA and protein-protein regulation
- regulation changes with time

Gene Regulatory Networks

Gene Networks: models of measurable properties of Gene Regulatory Systems.

Gene networks model functional elements of a Gene Regulation System together with the regulatory relationships among them in a computational formalism.

Types of relationships: causal, binding specificity, protein-DNA binding, protein-protein binding, etc.

Modeling Formalisms

Combinatorial (Qualitative)

- Static Graph Models
- Boolean Networks
- Weight Matrix (Linear) Models
- Bayesian Networks

Physical (Quantitative or Continuous)

- Stochastic Models
- Difference / Differential Equation Models
- Chemical/Physical Models
- Concurrency models

Continuous Models of Gene Regulation

Outline

- Quantitative Modeling
- Discrete vs. Continuous
- Modeling problems
- Models:
  - ODE
  - PDE
  - Stochastic
- Conclusions

Quantitative Modeling in Biology

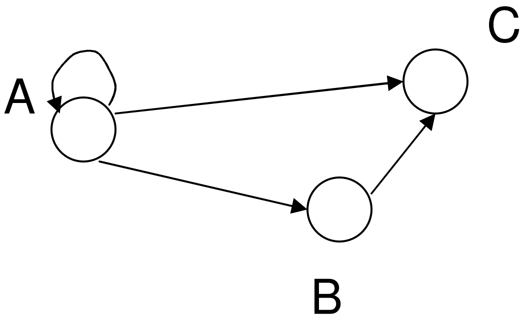
- State variables: concentrations of substances, e.g. proteins, mRNA, small molecules, etc.
- Knowing a system means being able to predict the concentrations of all key substances (state variables)
- Quantitative Modeling is the process of connecting the components of a system in a mathematical equation
- Solving the equations yields testable predictions for all state variables of the system

Discrete vs. Continuous

- Here we will talk about continuous models, where values of variables change continuously in time (and/or space)
- On a molecular scale things are discrete, but on a macro scale they blend in and look continuous
- Next class we’ll discuss discrete models

Why Continuous?

- Continuous models are appealing because they allow for instantaneous change
- Continuous models let us express the <u>precise</u> relationships between instantaneous states of variables in a system



vs.

$$\frac{dA}{dt} = 1 - 2A$$

$$\frac{dB}{dt} = 0.5A$$

$$\frac{dC}{dt} = 2A + B$$

Problems

When modeling with differential equations we face all the same problems as in the discrete models:

- Posing the equations. This presumes we understand the underlying phenomenon
- Data Fitting. How do we learn the model from the data?
- Solving the equations. Means we can do the math
- Model Behavior. Analyzing the fitted model to understand its behavior

Recall the Modeling Process…

1. Knowledge
2. Modeling Objectives
3. Construct and Revise Models
4. Model behavior and predictions
5. Compare to new data
6. Better Models, goto 3
7. Learn…

1. Ordinary Differential Equations

Rate equation:

$$\frac{dx_i}{dt} = f_i(\mathbf{x}), \quad 1 \le i \le n$$

where

$\mathbf{x} = [\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}]$ is a vector of n concentrat ions

$f_i(x) : \mathbf{R}^n \to \mathbf{R}$ is a function

Systems of ODEs: There are n such equations

Solving the rate equations depends on f, but what is the form of the function f ?

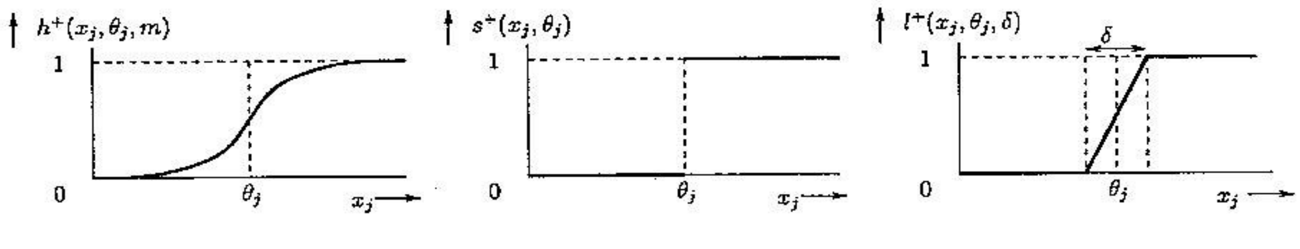
The answer is: as simple as possible.

The Rate Function and Regulation

- The rate function specifies the interactions between the state variables.
- Its input are the concentrations, and the output is indicative (i.e. a function of) the change in a gene’s regulation
- The regulation function describes how the concentration is related to regulation

$$h^{+}(x_j, \theta_j, m) = \frac{x_j^m}{x_j^m + \theta_j^m}, \tag{8}$$

- This is a typical regulation function, called a sigmoid, bellow compared to similar ones



Non-linear ODEs

The rate function is nonlinear!

Eg.

1. Sigmoidal
2. Nonlinear, additive. Summarizes all pair wise (and nothing but pair wise) relationship

$$\frac{dX_i}{dt} = \sum_j T_{ij} f_j(X_j)$$

3. Nonlinear, non-additive. Summarizes all pairs and triplets of relationships

$$\frac{dX_i}{dt} = \sum_{jk} T_{ijk} f_j(X_j) f_k(X_k) + \sum_j T_{ij} f_j(X_j)$$

Solving

- In general, these equations are difficult to solve analytically when $f_i(\mathbf{x})$ are non-linear
- Numerical Simulators/Solvers work by numerically approximating the concentration values at discretized, consecutive time-points. Popular software for biochemical interactions:
  - DBsolve
  - GEPASI
  - MIST
  - SCAMP
- Although analytical solutions are impossible, we can learn a lot from general analyses of the behavior of the models, which some of the packages above provide

Model Behavior:

- Feedback is essential in biological systems. The following is known about feedback:
  - negative feedback loops: system approach or oscillate around a single steady state
  - positive feedback loops: system tends to settle in one of two stable states
  - in general: a negative feedback loop is necessary for stable oscillation, and a positive feedback loop is necessary for multistationarity

Data Fitting

- Fitting the parameters of a non-linear system is a difficult problem.
- Common solution: non-linear optimization scheme
  - explore the parameter space of the system
  - for each choice of parameters the models are solved numerically (e.g. Runge-Kutta)
  - the parameterized model is compared to the data with a goodness of fit function. It is this function that is optimized
- Genetic Algorithms and Simulated Annealing, with proper transition functions have been used with promising results

Linear and Piecewise Linear ODEs

Linear

- These are much easier to deal with: if the input variables are limited by a constant, they can be solved and learned polynomially, depending on the amount of data available

$$\frac{dX_i}{dt} = \sum_j w_{ij} X_j$$

- One way to learn them is by approximating them with linear weight models

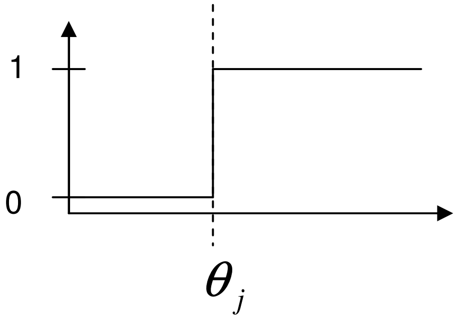
Piecewise linear

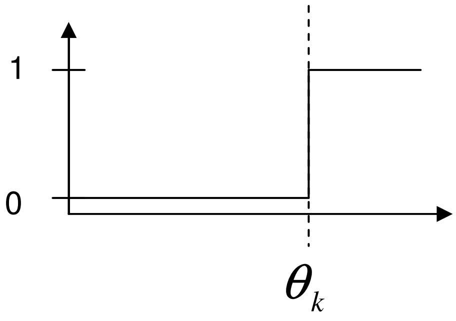
- Approximating the sigmoid regulatory function with a step function

$$\frac{dX_i}{dt} = g_i(\mathbf{x}) - \gamma_i x_i,\ 1 \le \mathrm{i} \le \mathrm{n}$$

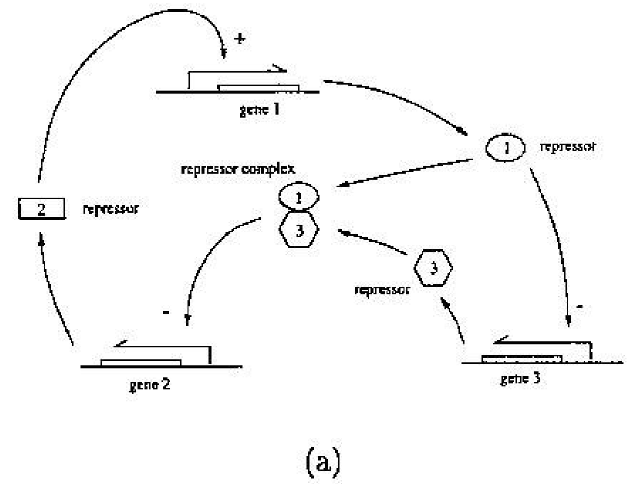
$$g_i(\mathbf{x}) = \sum_{l \in L} k_{il} b_{il}(\mathbf{x}) \ge 0$$

- Here the function b_{il} is a function of n variables, defined in terms of sums and products of step functions:





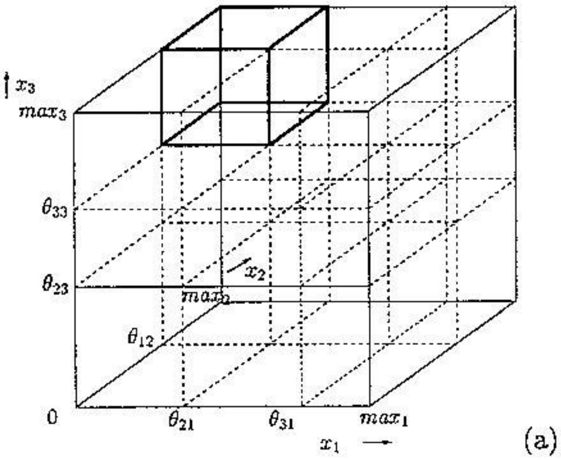
- This amounts to subdividing n-dimensional space into “orthants”, and in each of the orthants the PLODEs reduce to ODEs



$$\dot{x}_1 = \kappa_1\, s^+(x_2, \theta_{21}) - \gamma_1\, x_1$$
$$\dot{x}_2 = \kappa_2\, (1 - s^+(x_1, \theta_{11})\, s^+(x_3, \theta_{31})) - \gamma_2\, x_2$$
$$\dot{x}_3 = \kappa_3\, s^-(x_1, \theta_{12}) + \kappa_4\, s^-(x_3, \theta_{32}) - \gamma_3\, x_3$$

(b)

FIG. 9. **(a)** Example regulatory network of three genes and **(b)** corresponding piecewise-linear differential equations; x_1, x_2, and x_3 represent protein or mRNA concentrations, respectively, $\kappa_1, \ldots, \kappa_4$ production constants, $\gamma_1, \ldots, \gamma_3$ degradation constants, and $\theta_{11}, \theta_{12}, \theta_{21}, \theta_{31}, \theta_{32}$ threshold constants.



$$\dot{x}_1 = \kappa_{12} - \gamma_1 x_1$$
$$\dot{x}_2 = -\gamma_2 x_2$$
$$\dot{x}_3 = \kappa_{31} - \gamma_3 x_3$$

(b)

FIG. 10. (a) The phase space box of the model in Fig. 9, divided into $2 \cdot 3 \cdot 3 = 18$ orthants by the threshold planes. **(b)** The state equations for the orthant $0 \le x_1 < \theta_{21}$, $\theta_{12} < x_2 \le max_2$, and $\theta_{33} < x_3 \le max_3$ (the orthant demarcated by bold lines).

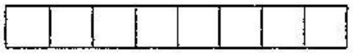
de Jong, JCB 2002

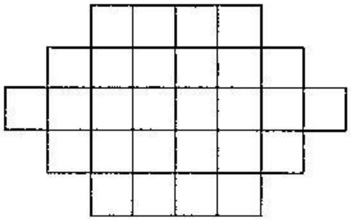
2. PDES

- ODEs count on spatial homogeneity
- In other words, ODEs don’t care where the processes take place
- But in some real situation this assumption clearly does not hold
  - Diffusion
  - Transcription factor gradients in development
  - Multicelular organisms

Example: Reaction-Diffusion Equations

$$\frac{dx_i^{(l)}}{dt} = f_i(\boldsymbol{x}^{(l)}) + \delta_i \left(x_i^{(l+1)} - 2x_i^{(l)} + x_i^{(l-1)} \right), \ 1 \le i \le n, \ 1 < l < p. \tag{16}$$





The equation above describes the change in conc. for all state variables, in all cells of the line above. When the number of cells is large, this becomes a PDE:

$$\frac{\partial x_i}{\partial t} = f_i(\boldsymbol{x}) + \delta_i \frac{\partial^2 x_i}{\partial l^2}, \ 0 \le l \le \lambda, \ 1 \le i \le n. \tag{17}$$

If it is assumed that no diffusion occurs across the boundaries $l = 0$ and $l = \lambda$, the boundary conditions become

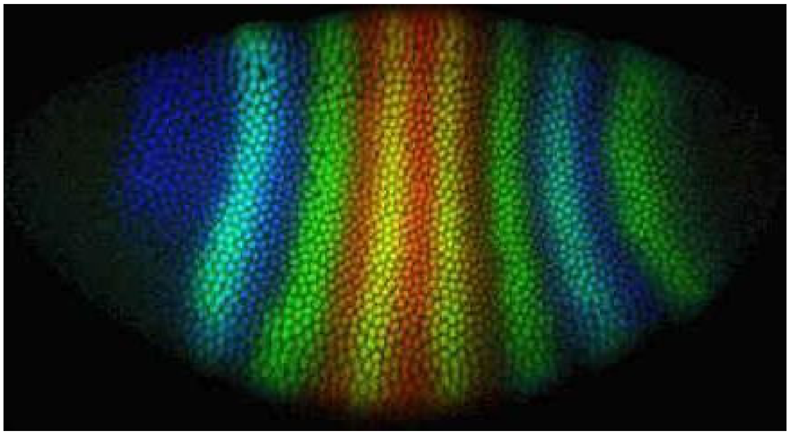
$$\frac{\partial^2}{\partial l^2} x_i(0, t) = 0 \text{ and } \frac{\partial^2}{\partial l^2} x_i(\lambda, t) = 0. \tag{18}$$

These equations were first introduced in the study of developmental phenomena and pattern formation by Turing.

Direct analytical solutions are impossible even for two variables (n=2)

Drosophila Example

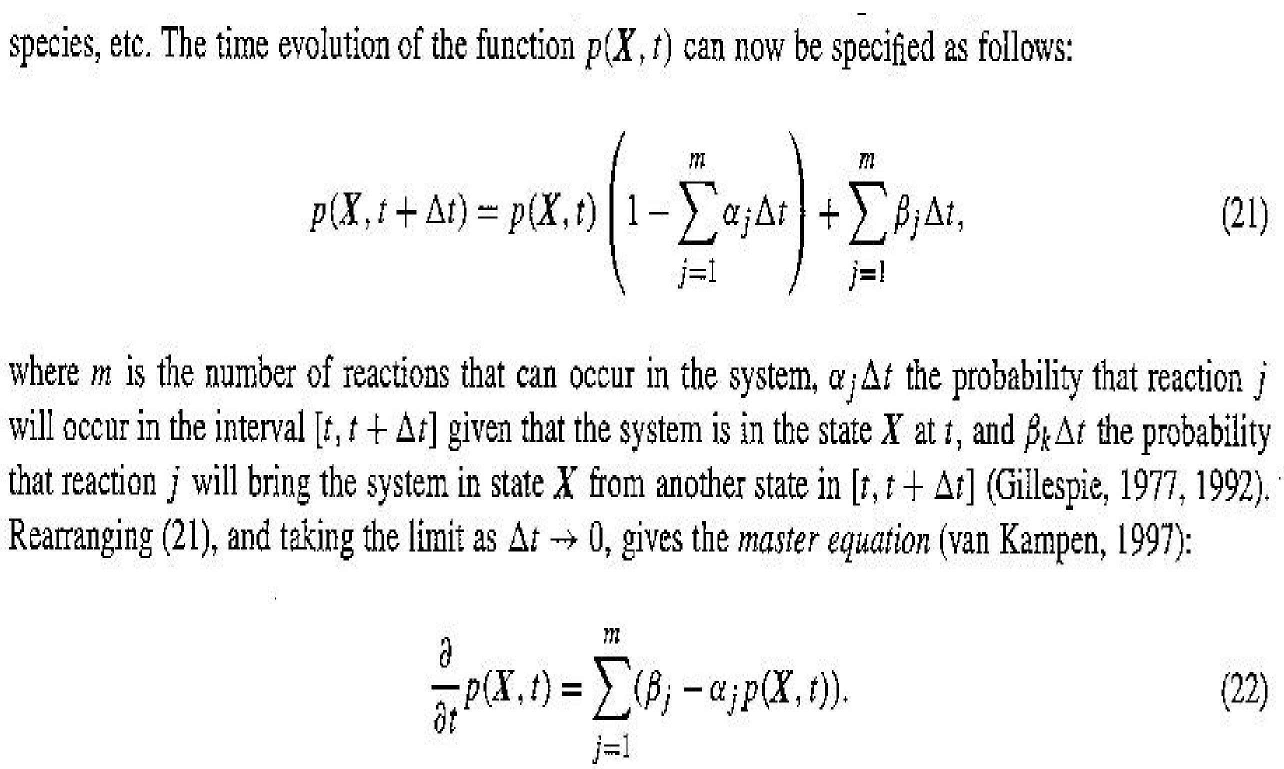
- These PDE models have been used repeatedly to model developmental examples in the fruit fly
- Instances of the reaction-diffusion equations (only more specific) have been used to model the striped patterns in a drosophila embryo



3. Stochastic Master Equations

- Deterministic modeling is not always possible, but also sometimes incorrect
- Assumptions of deterministic, continuous models:
  - Concentrations of substances vary deterministically
  - Conc. Of subst. vary continuously
- On molecular level, both assumptions may not be correct
- Solution: Instead of deterministic values, accept a joint probability distribution, similar to the one discussed in the Bayesian Network lectures.

Equation:



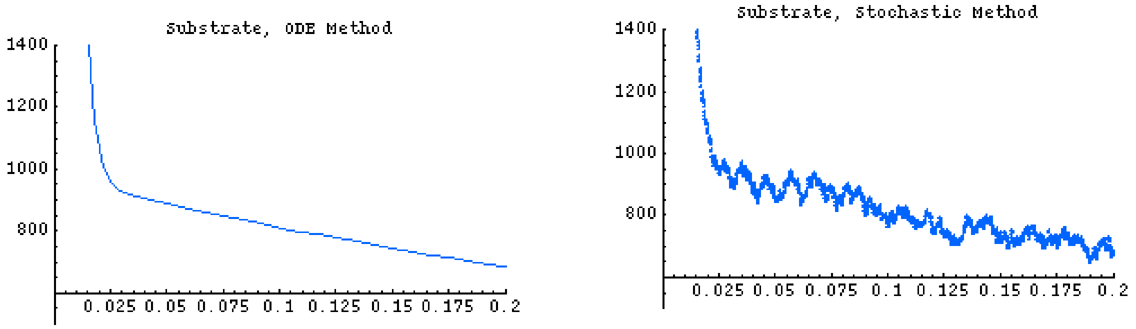
species, etc. The time evolution of the function $p(X,t)$ can now be specified as follows:

$$p(X,t+\Delta t)=p(X,t)\left(1-\sum_{j=1}^{m}\alpha_j\Delta t\right)+\sum_{j=1}^{m}\beta_j\Delta t, \tag{21}$$

where m is the number of reactions that can occur in the system, $\alpha_j\Delta t$ the probability that reaction j will occur in the interval $[t,t+\Delta t]$ given that the system is in the state X at t, and $\beta_k\Delta t$ the probability that reaction j will bring the system in state X from another state in $[t,t+\Delta t]$ (Gillespie, 1977, 1992). Rearranging (21), and taking the limit as $\Delta t\rightarrow 0$, gives the *master equation* (van Kampen, 1997):

$$\frac{\partial}{\partial t}p(X,t)=\sum_{j=1}^{m}(\beta_j-\alpha_j p(X,t)). \tag{22}$$

These equations are very difficult to solve and simulate!



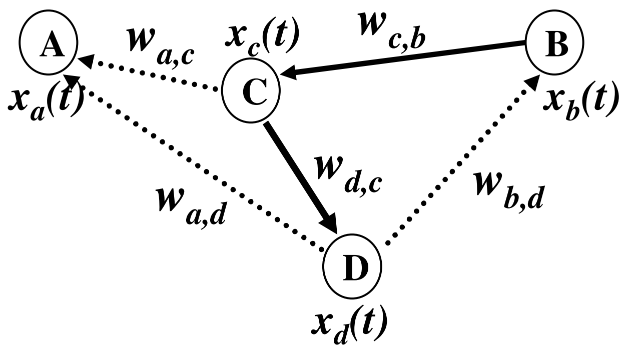
(c) Jason Kastner and Caltech

ODE vs. Stochastic solutions

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



$\boldsymbol{x_i(t)}$ – state of gene *i* at time *t*
$\boldsymbol{w_{ij}}$ – regulatory influence of gene *j* on gene *i*
- $\boldsymbol{w_{ij} > 0}$, activation
- $\boldsymbol{w_{ij} < 0}$, inhibition
- $\boldsymbol{w_{ij} = 0}$, 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 <u>all</u> 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.

$$\underline{\mathbf{y_i} = \begin{bmatrix} x_1(i) & x_2(i) & \cdots & x_n(i) \end{bmatrix}.}$$

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

$\mathbf{A}^{(\mathbf{k}\times\mathbf{n})}$ 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

$\mathbf{B}^{(\mathbf{k}\times\mathbf{n})}$ 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:

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

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

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

- If $\underline{k = n}$ there is a unique solution;

- If $\underline{k < n}$, 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:

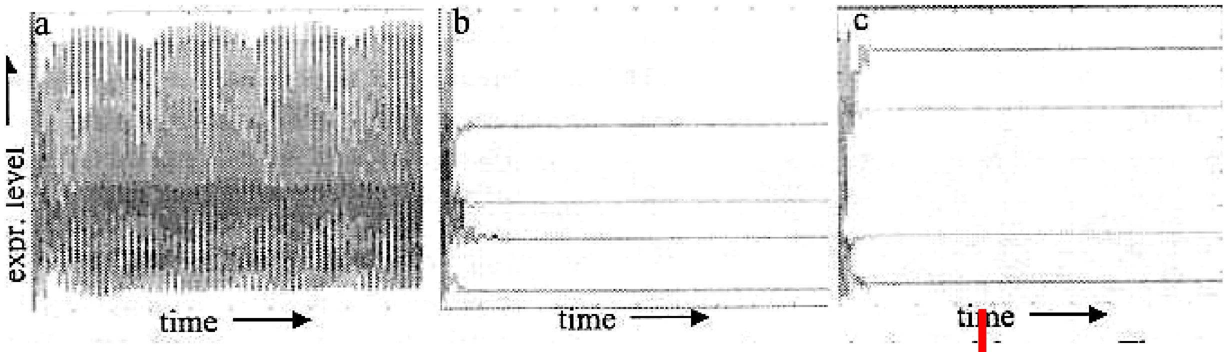
$$\mathbf{W} = \mathbf{A}^{**}\mathbf{A}, \quad \boxed{\mathbf{A}^{**} = \mathbf{A}^{\mathbf{T}}(\mathbf{A}\mathbf{A}^{\mathbf{T}})^{-\mathbf{1}}}$$

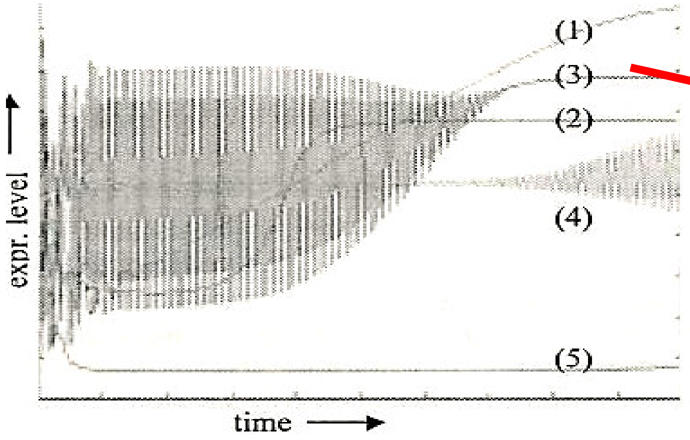
Normalization

- 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





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(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
- 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)

Biography

- Hidde de Jong, Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. Journal of Computational Biology 9(1): 67-103 (2002).
- McAdams and Arkin, Annu. Rev. Biophys. Biomol. Struct. 1998 (27)
- Setty, Y. et al., Detailed map of a cis-regulatory input function, PNAS, 100:7702-7707 (2003)
- Fangting et al., PNAS 2004 (101)
- Albert, R. and Othmer, H.G. Journal of Theoretical Biology 223, 1-18 (2003).
- Davidson, E.H. et al., Science 295, 1669-1678, 2002
- D. C. Weaver and C. T. Workman and G. D. Stromo, Pacific Symposium on Biocomputing, 1999.
- D'Haeseleer et al., Pacific Symposium on Biocomputing, 1999.