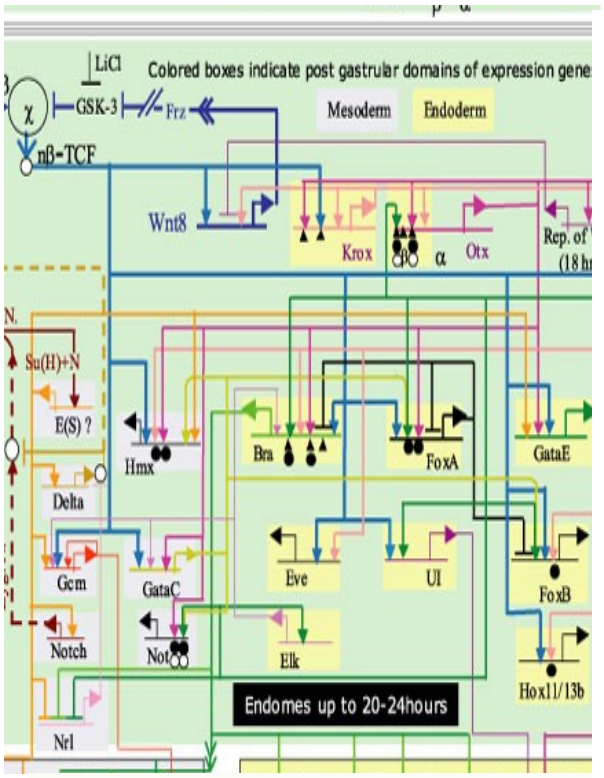
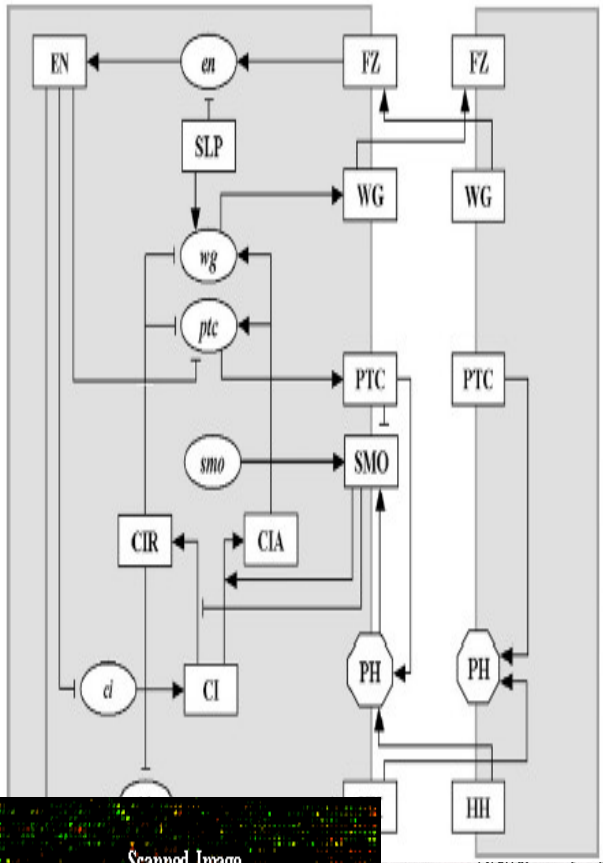
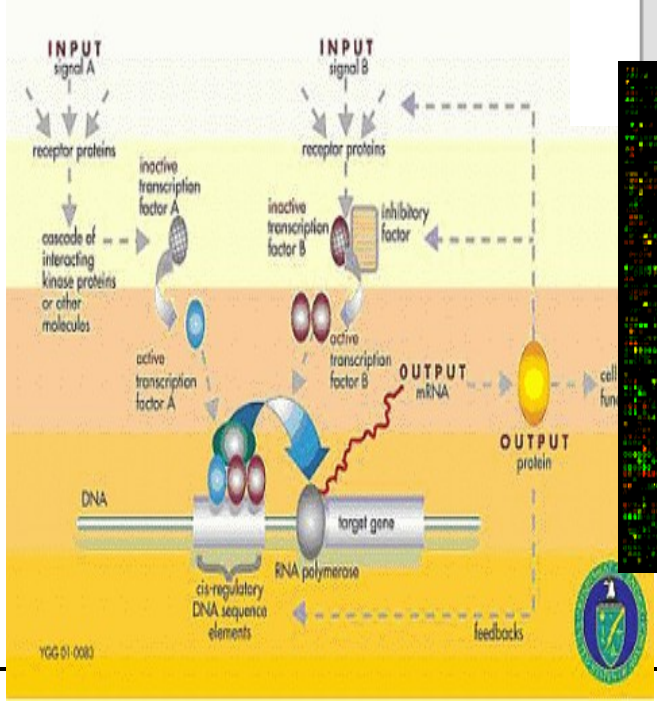
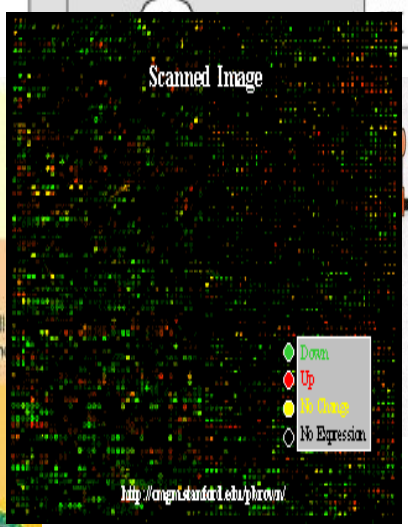
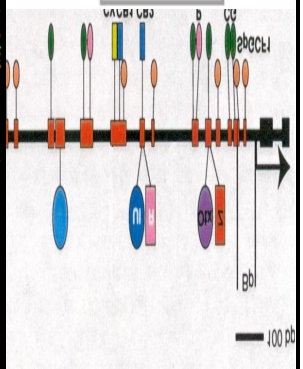
ECS 234: Gene Regulation Modelling: Continuous Models











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:

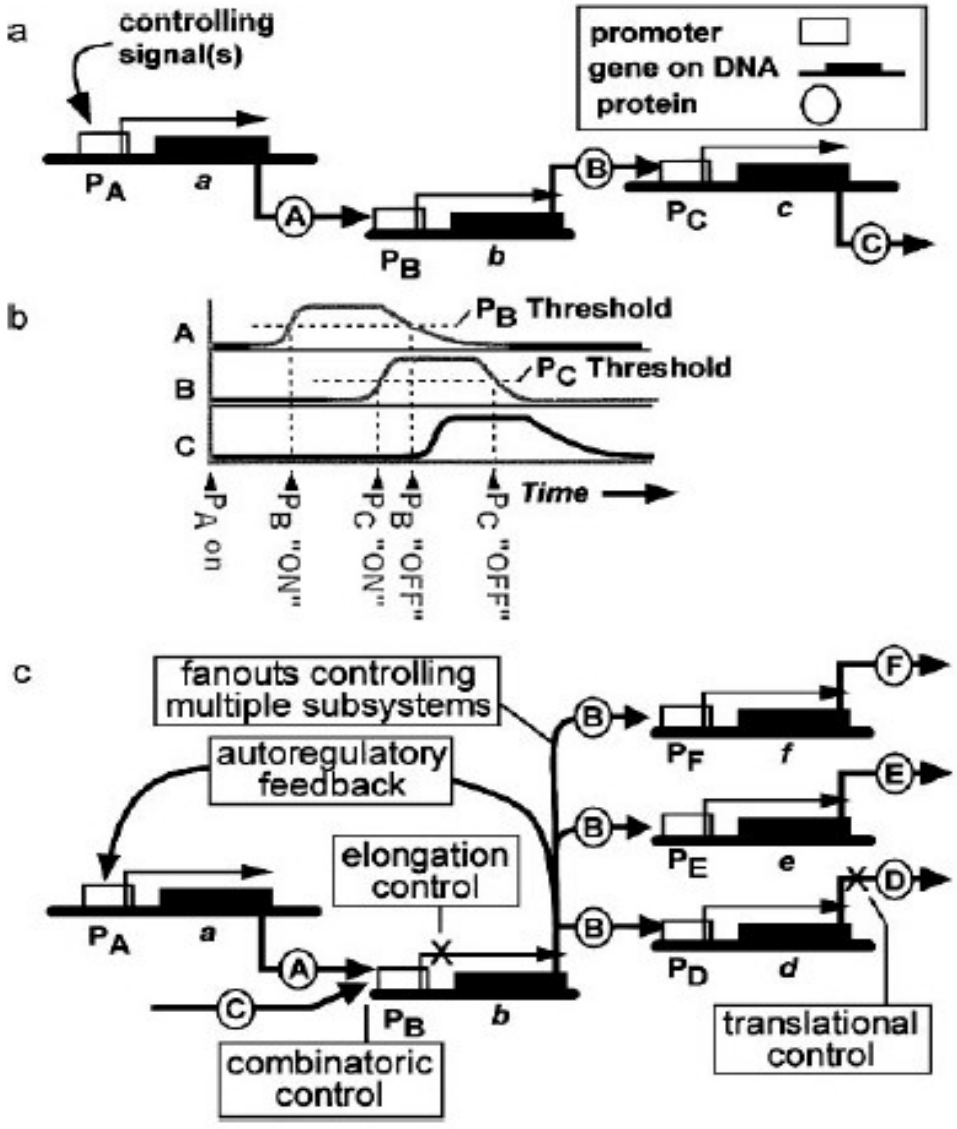
- cis-trans specificity
- small number of trans factors to a cis element: 8-10
- cis elements are programs
- regulation is event driven (asynchronous)
- regulation systems are noisy environments
- Protein-DNA and protein-protein regulation
- regulation changes with time

Gene 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.

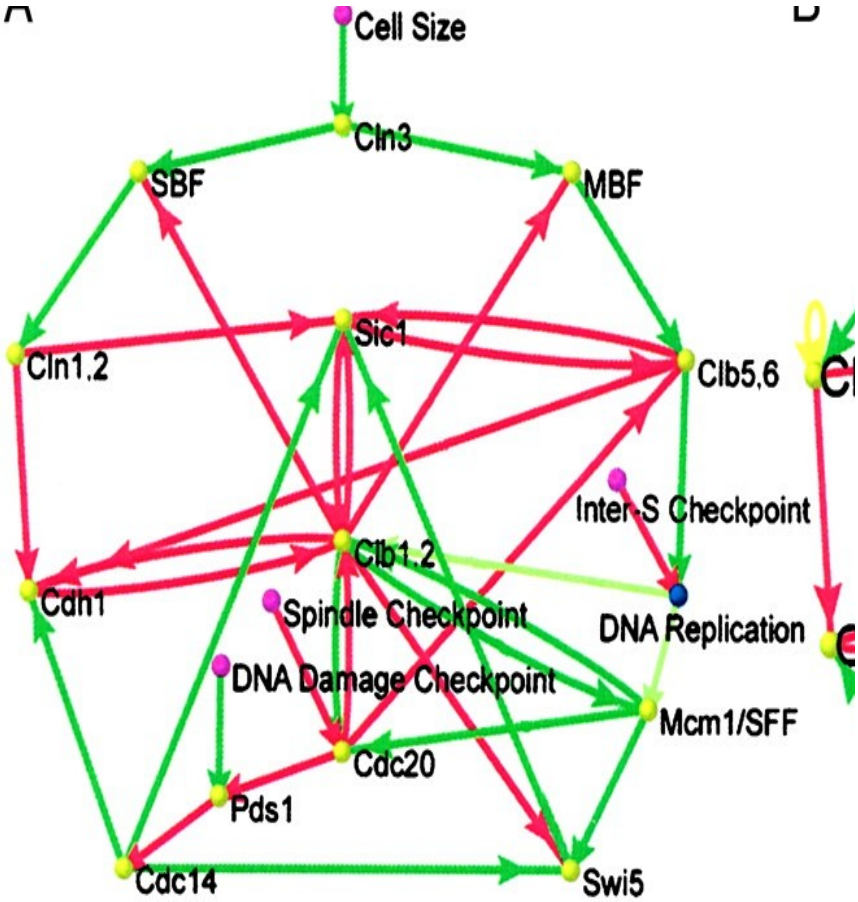
Types of relationships: causal, binding specificity, protein-DNA binding, protein-protein binding, etc.

Simple Genetic Circuits



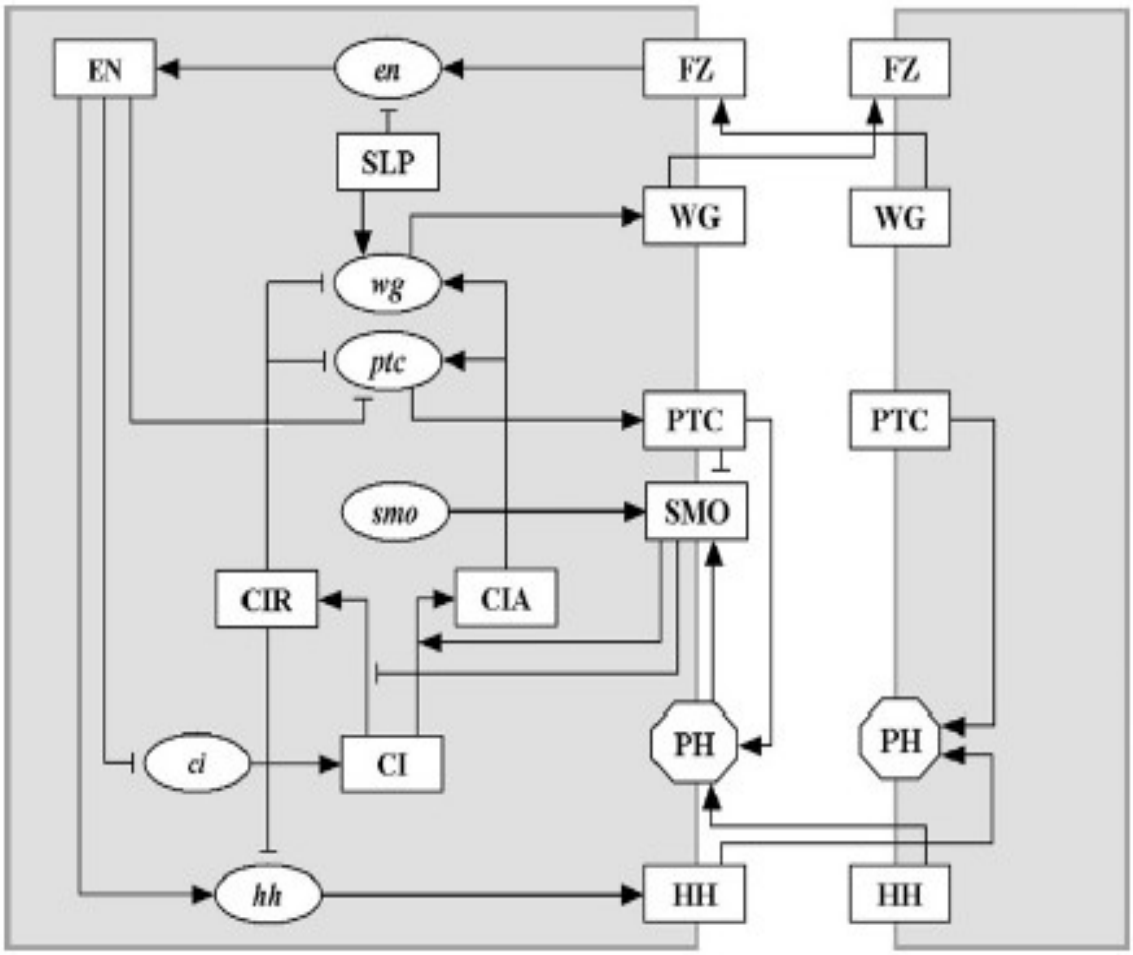
McAdams and Arkin et al 1998

Cell cycle network in S. Cerevisiae



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

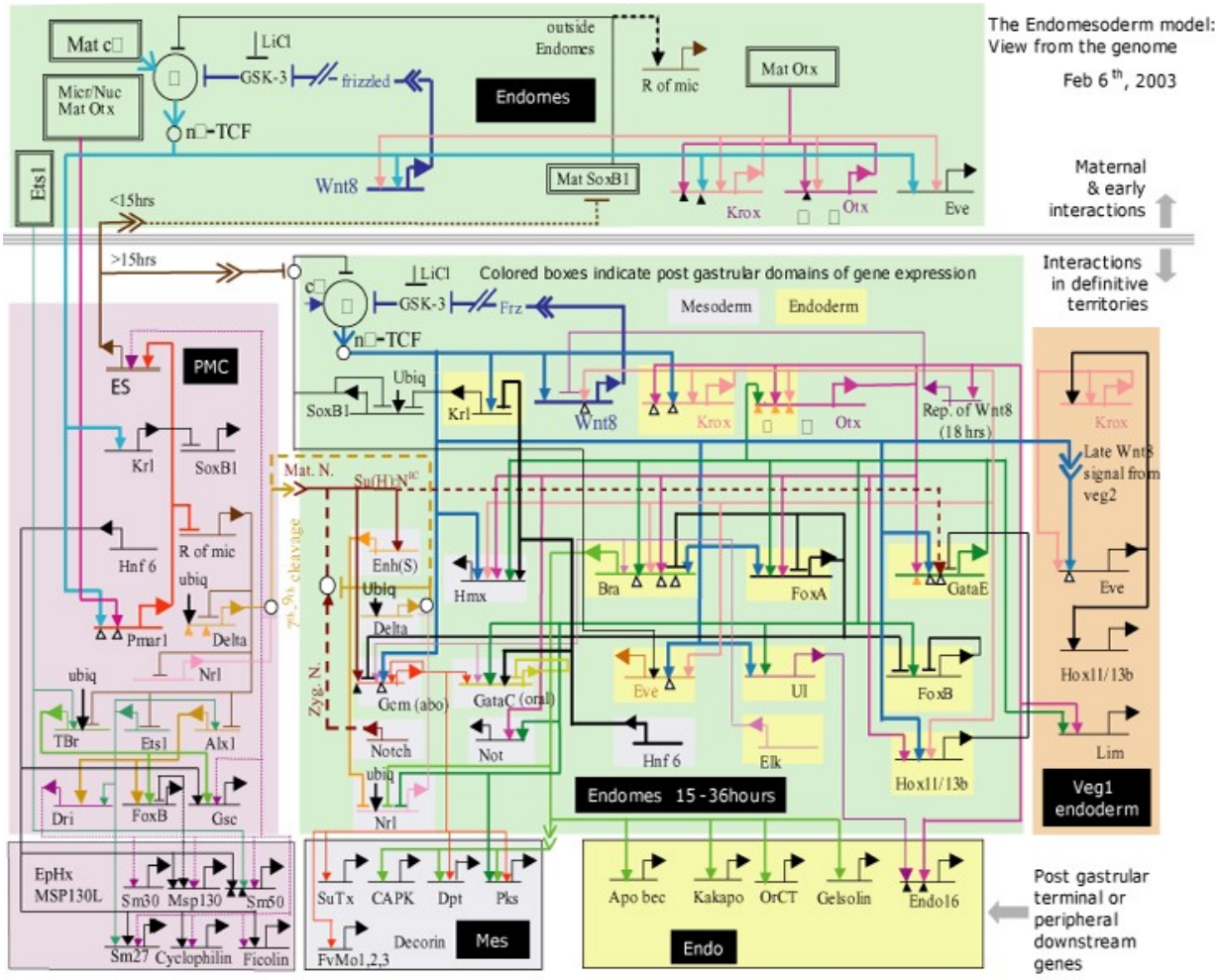
Segment polarity network in Drosophila



Albert and Otmer, JTB 2003

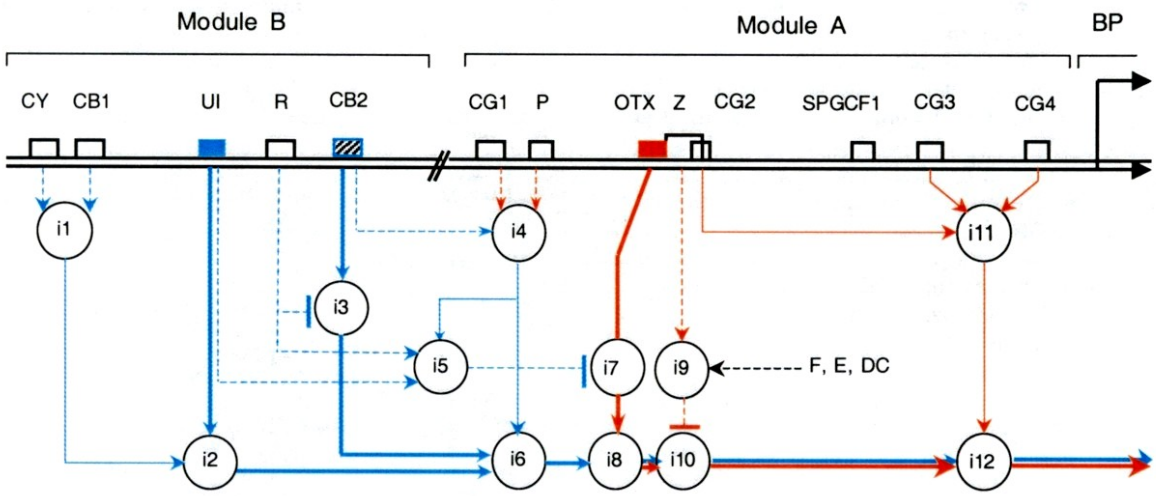
Nodes: mRNA (round), protein (rectangle), protein complex (octagon)
Edges: biochemical interactions or regulatory relationships

Gene network of endomeso development in Sea Urchin



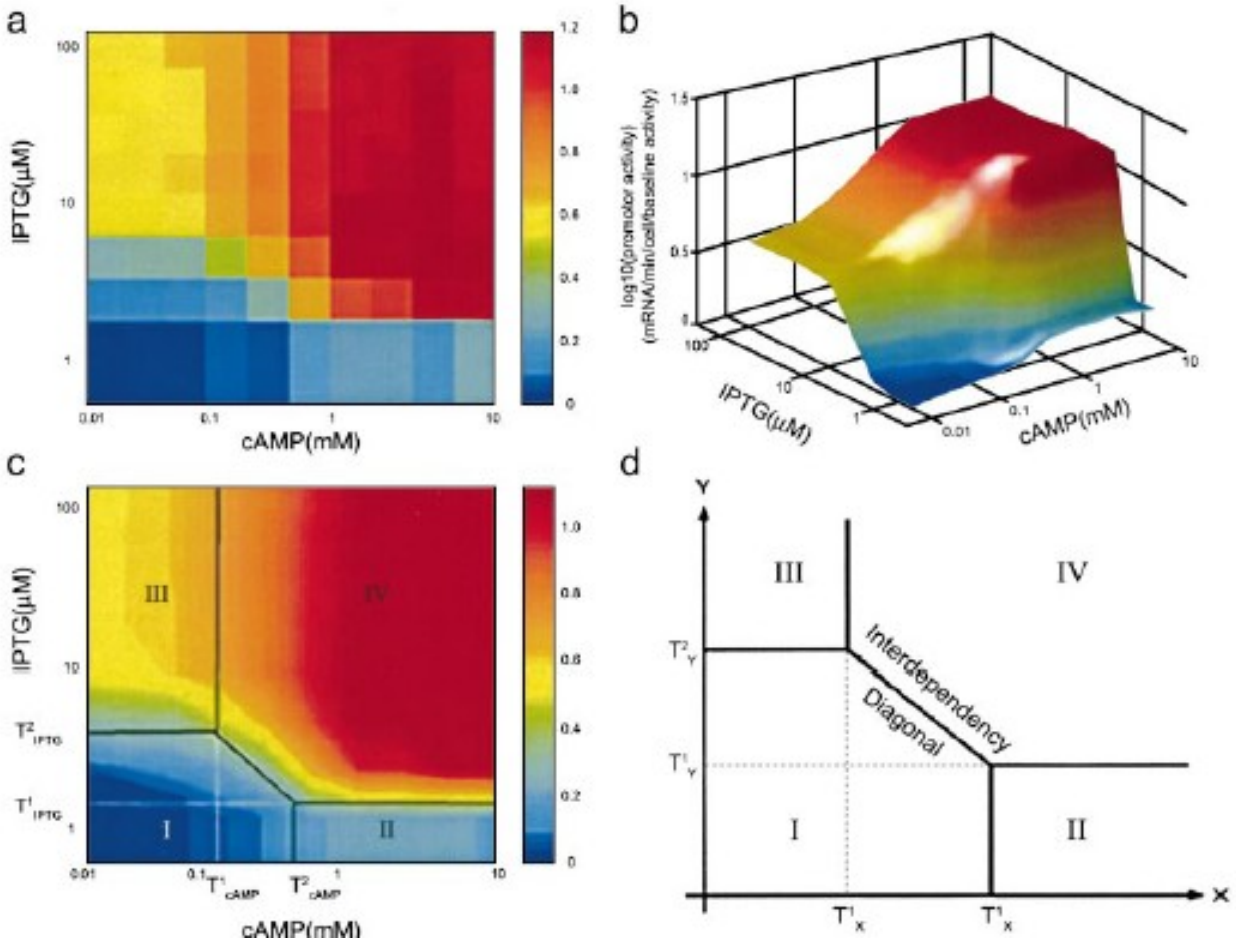
Davidson et al. Science 2002

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

Cis-regulatory input of lacZYA operon in E. coli



Setty et al. PNAS 2003

Modeling Formalisms

Combinatorial (Qualitative)	**Physical (Quantitative or Continuous)**
• Static Graph Models • Boolean Networks • Weight Matrix (Linear) Models • Bayesian Networks	• 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

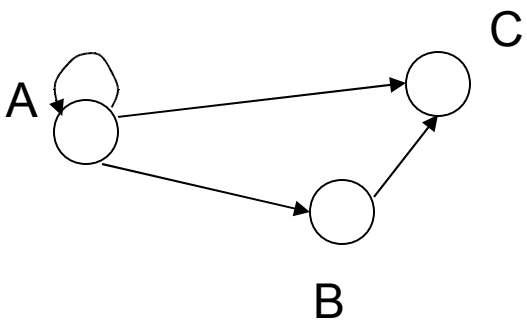
- State variables: concentrations of substances, e.g. proteins, mRNA, small molecules, etc.
- Knowing a system means being able to predict the concentrations of all key substances (state variables)
- Quantitative Modeling is the process of connecting the components of a system in a mathematical equation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

$\mathbf{x} = [x_1, \ldots, x_n]$ is a vector of n concentrations

$f_i(x) : \mathbf{R}^n \rightarrow \mathbf{R}$ 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?

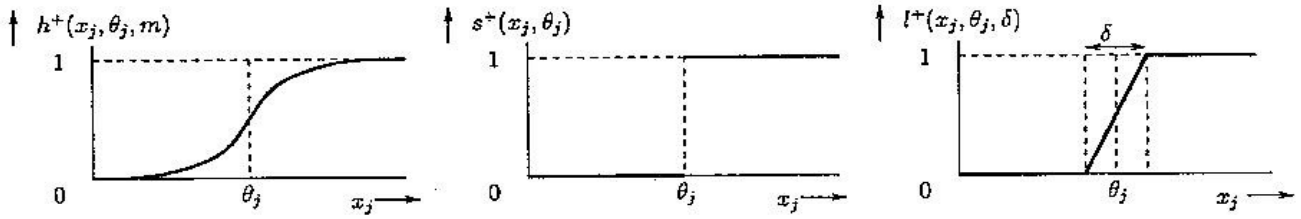
The answer is: as simple as possible.

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

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

- 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

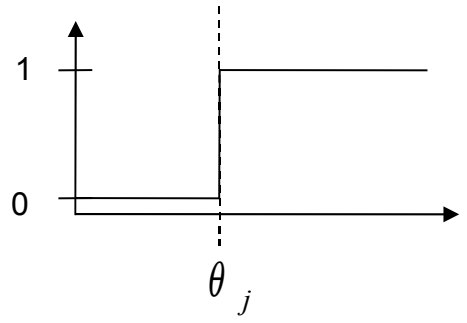
Piecewise linear

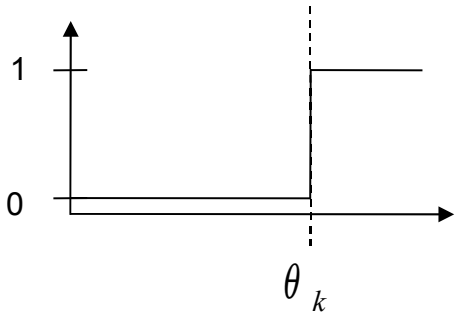
- Approximating the sigmoid regulatory function with a step function

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

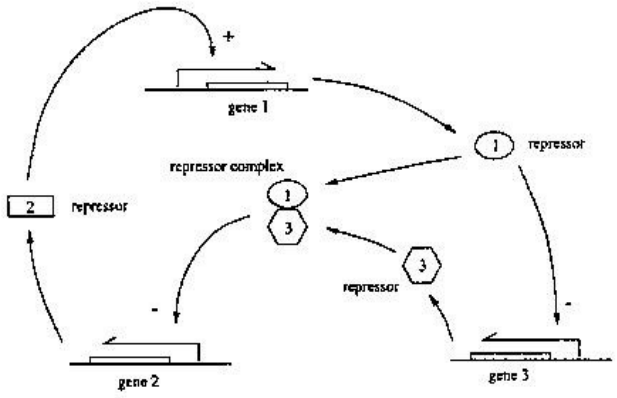
$$g_i(\mathbf{x}) = \sum_{l \in L} k_{il} b_{il}(\mathbf{x}) \ge 0$$

- 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

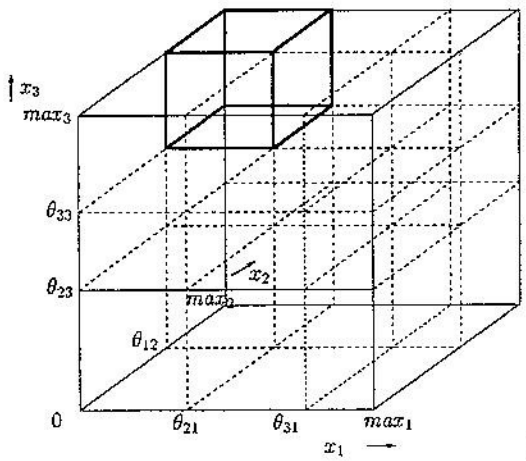


(a)

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

(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.



(a)

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

(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).

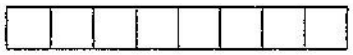
de Jong, JCB 2002

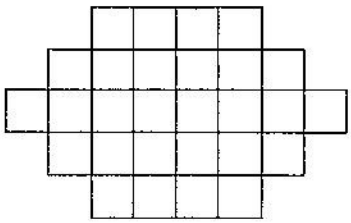
2. PDES

- 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

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





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:

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

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

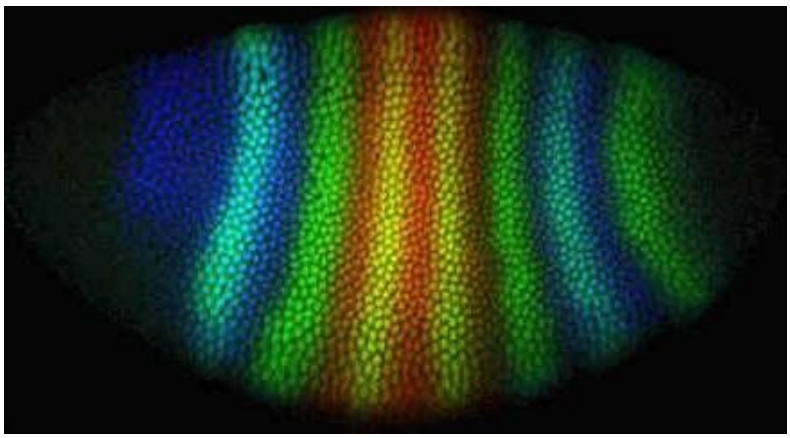
$$\frac{\partial^2}{\partial l^2} x_i(0, t) = 0 \text{ and } \frac{\partial^2}{\partial l^2} x_i(\lambda, t) = 0. \tag{18}$$

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 substances 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:

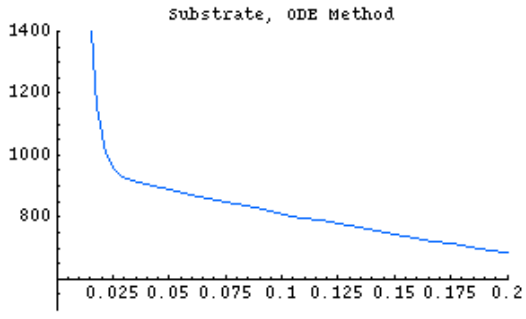
species, etc. The time evolution of the function $p(\boldsymbol{X}, t)$ can now be specified as follows:

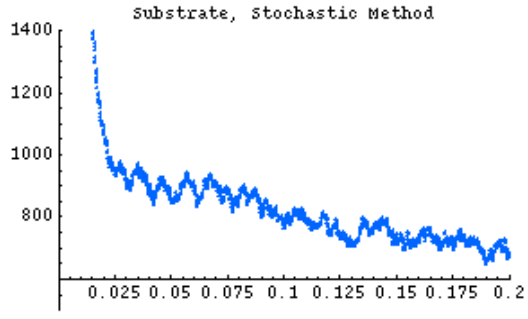
$$p(\boldsymbol{X}, t + \Delta t) = p(\boldsymbol{X}, t)\left(1 - \sum_{j=1}^{m} \alpha_j \Delta t\right) + \sum_{j=1}^{m} \beta_j \Delta t, \tag{21}$$

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 $\boldsymbol{X}$ at t, and $\beta_k \Delta t$ the probability that reaction j will bring the system in state $\boldsymbol{X}$ from another state in $[t, t + \Delta t]$ (Gillespie, 1977, 1992). Rearranging (21), and taking the limit as $\Delta t \to 0$, gives the *master equation* (van Kampen, 1997):

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

These equations are very difficult to solve and simulate!





ODE vs. Stochastic solutions

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.