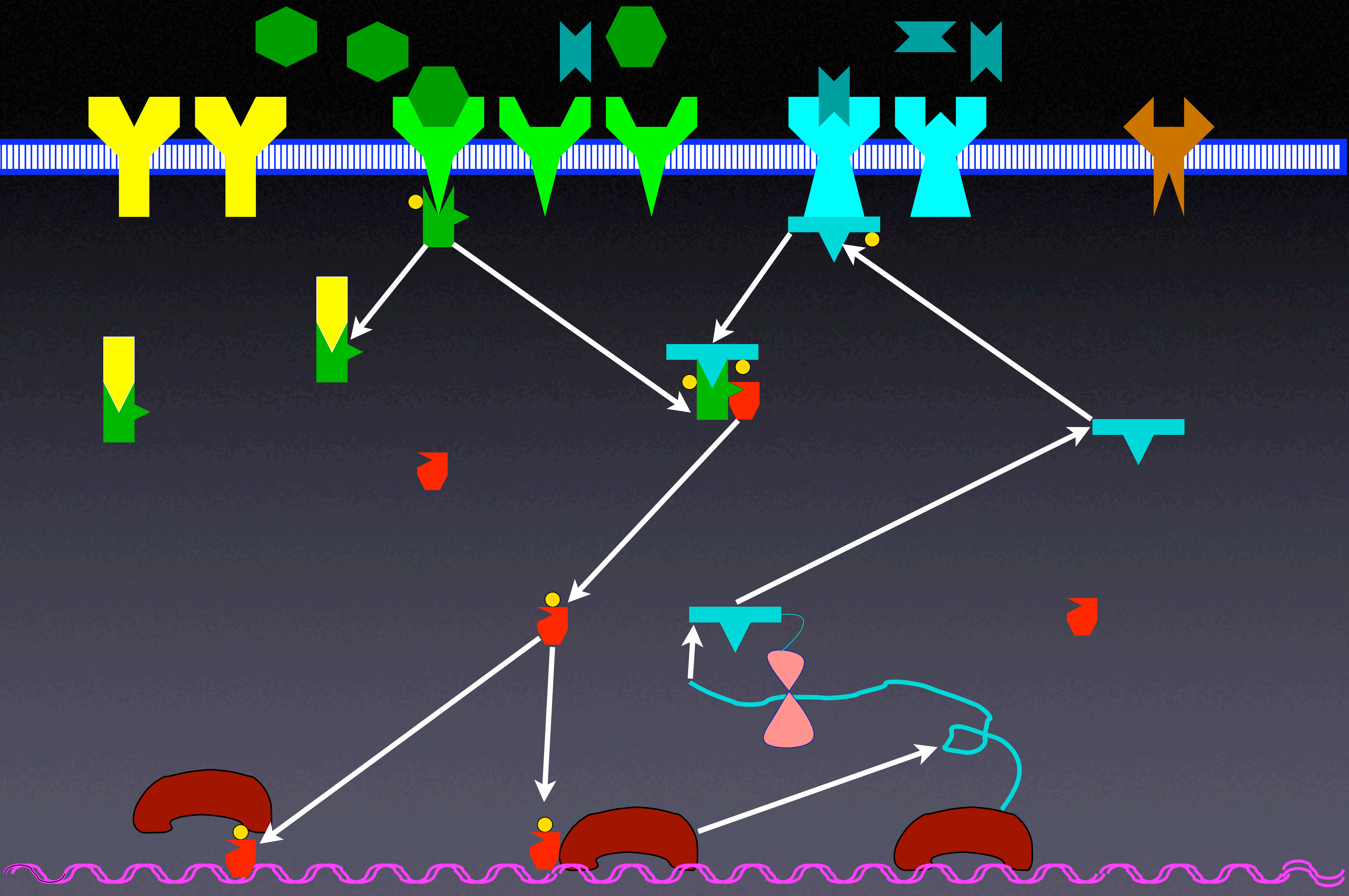
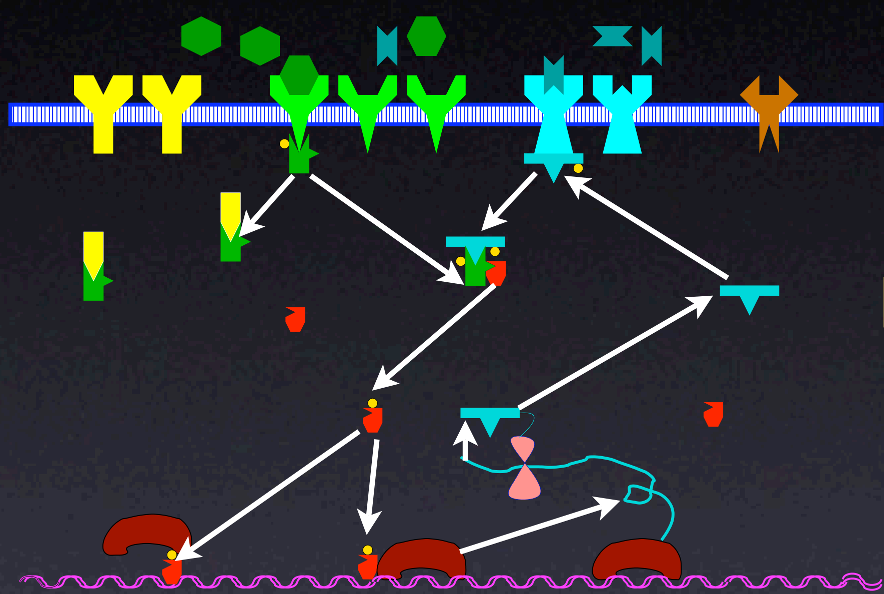
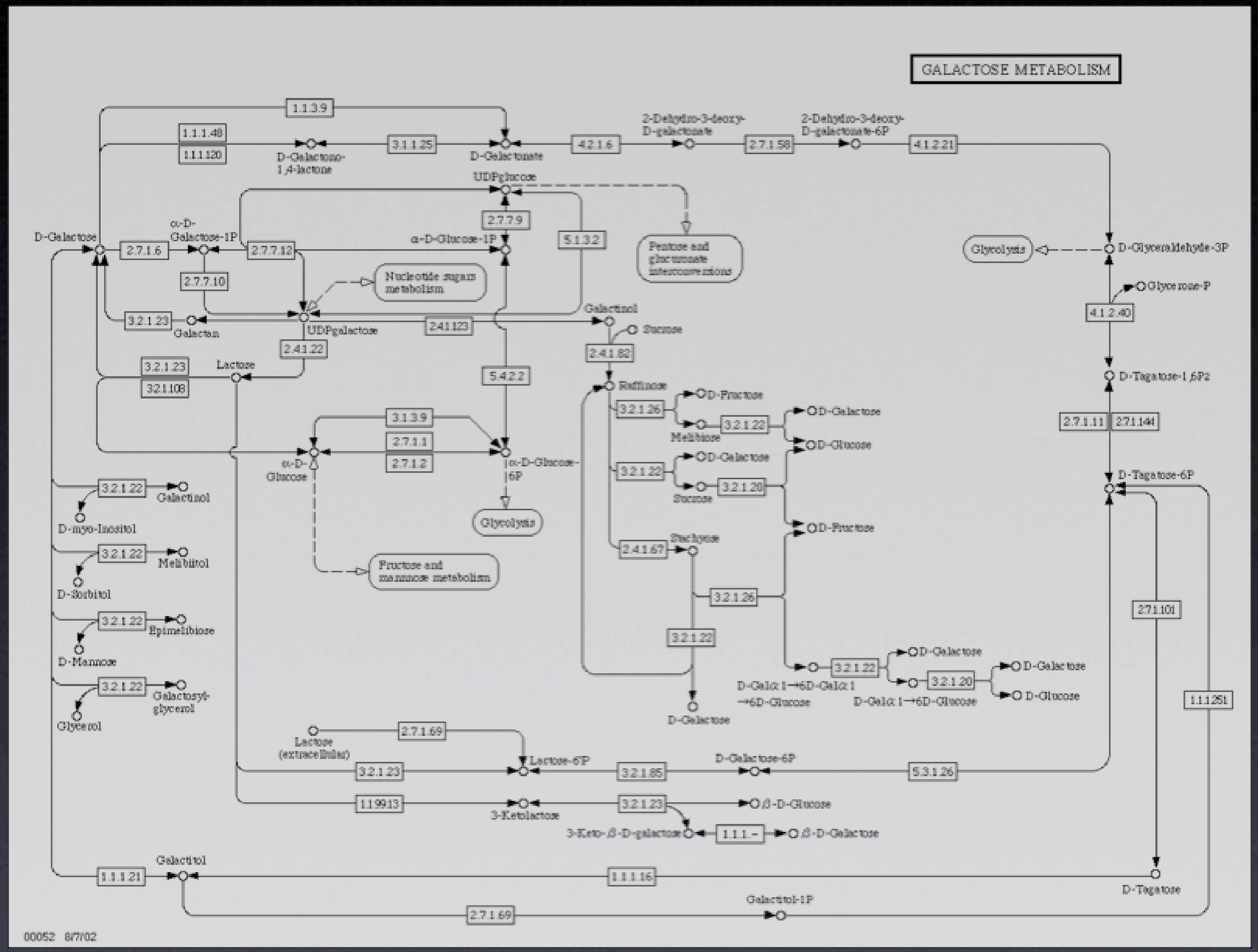
Biomolecular reaction networks: gene regulation & the Repressilator

Phys 682/ CIS 629: Computational Methods for Nonlinear Systems



Processing information to coordinate activity



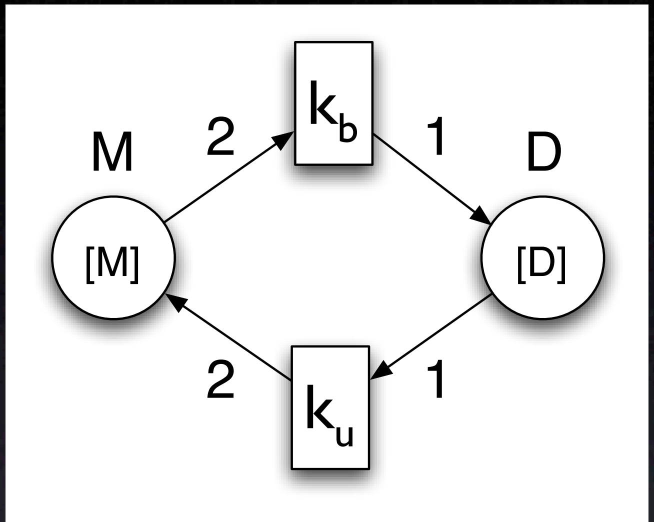


regulatory & signaling networks as information processing systems

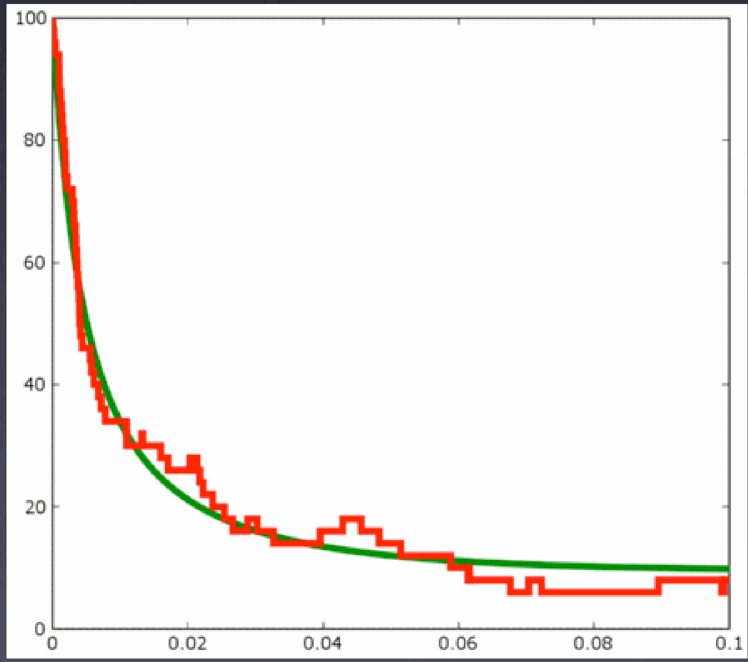
coordinating the processing of matter and energy

Stochastic cells: simple dimerization reaction

- simple dimerization reaction
  - homodimerization: M+M ↔ D
  - (as distinct from heterodimerization: A+B ↔ AB)
  - introduce Petri net representation
    - places (circles): molecular species
    - transitions (rectangles): chemical reactions, parameterized by rate constants
    - arcs (directed segments): stoichiometric weights
- compare stochastic and deterministic simulations
  - deterministic
    - dy/dt = f(y; kb, ku); y = (M, D)
  - stochastic
    - Gillespie algorithm



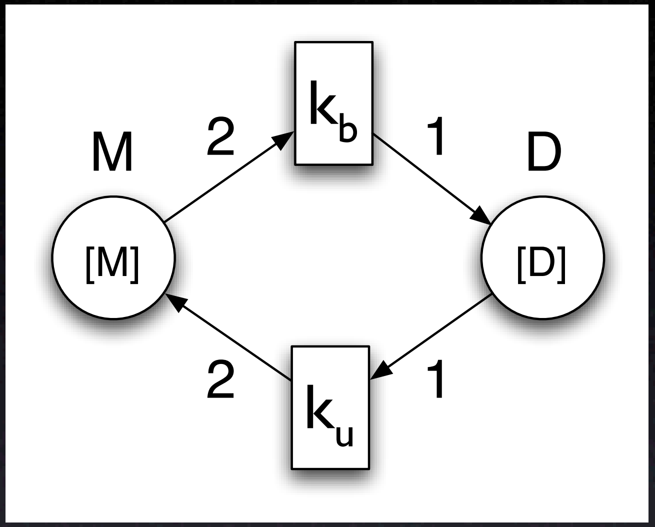
Petri net for M+M ↔ D



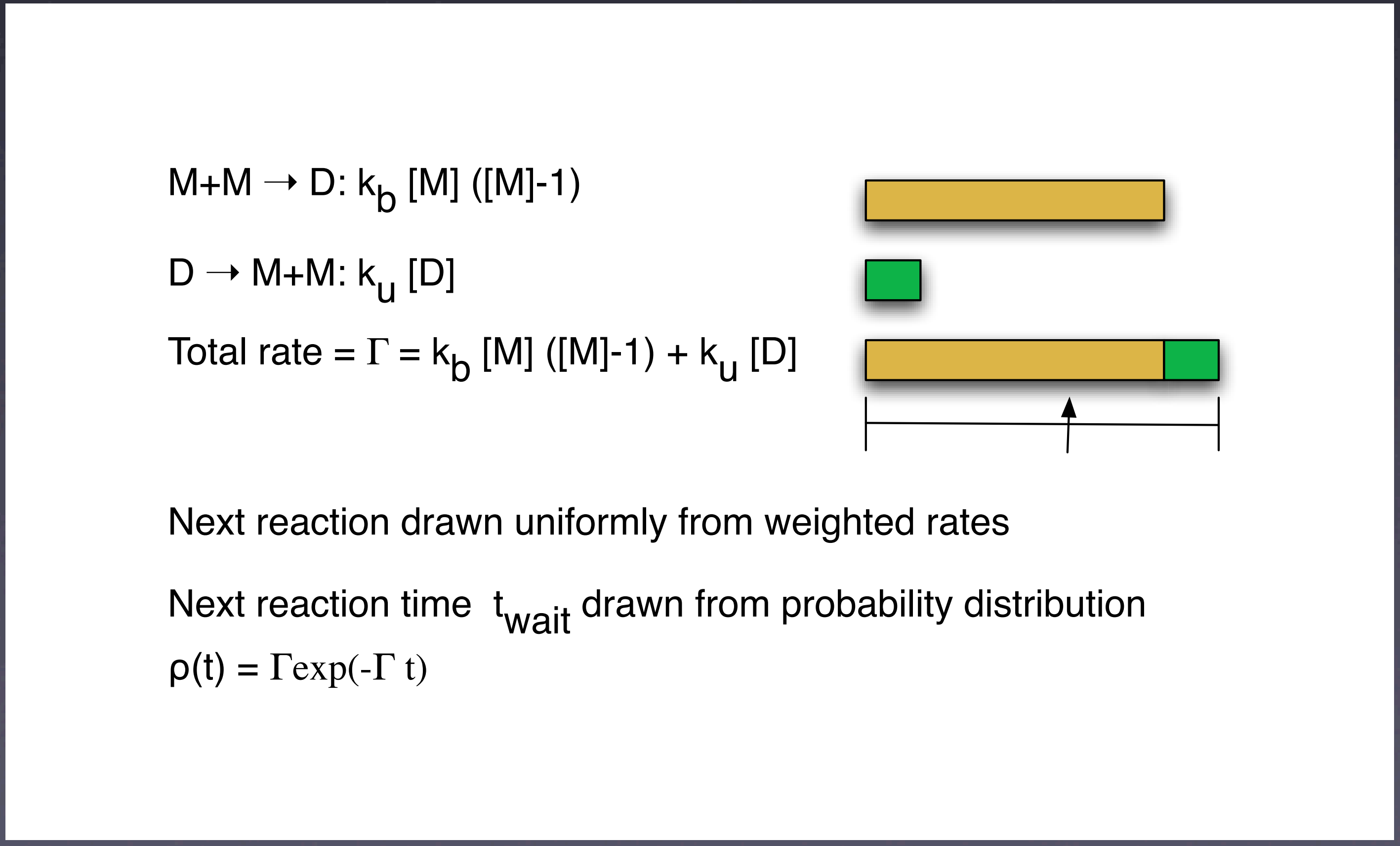
Stochastic vs. deterministic simulation

Gillespie algorithm

- Gillespie's "Direct Method", a.k.a. continuous time Monte Carlo, or the Bortz-Kalos-Lebowitz algorithm
- a stochastic method for simulating reaction dynamics
  - pick at random a reaction to occur next, and a time at which it will occur (consistent with reaction rates)

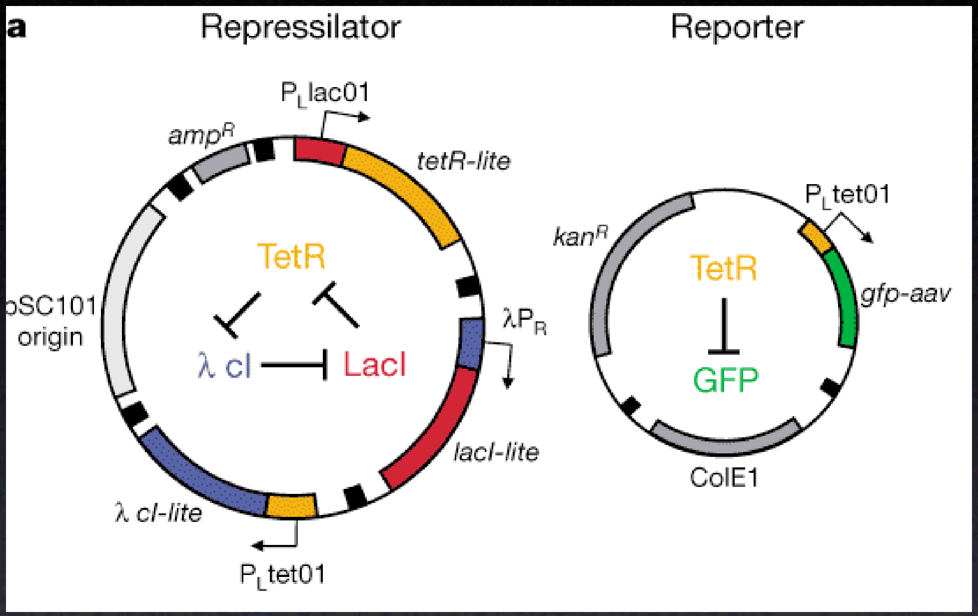


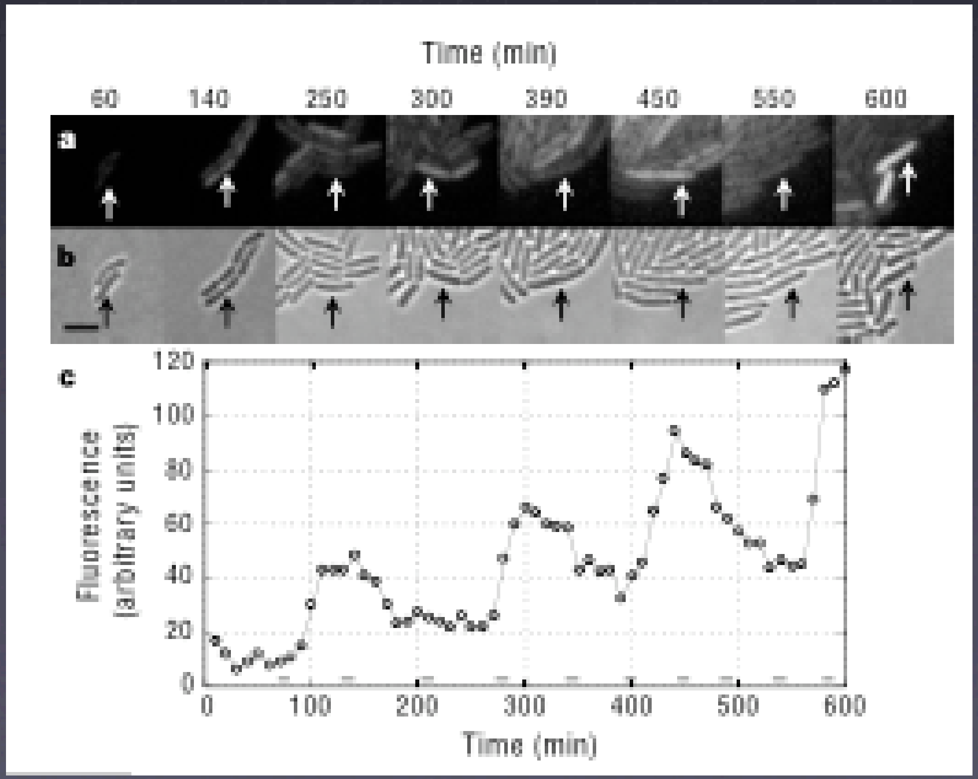
Petri net for M+M ↔ D



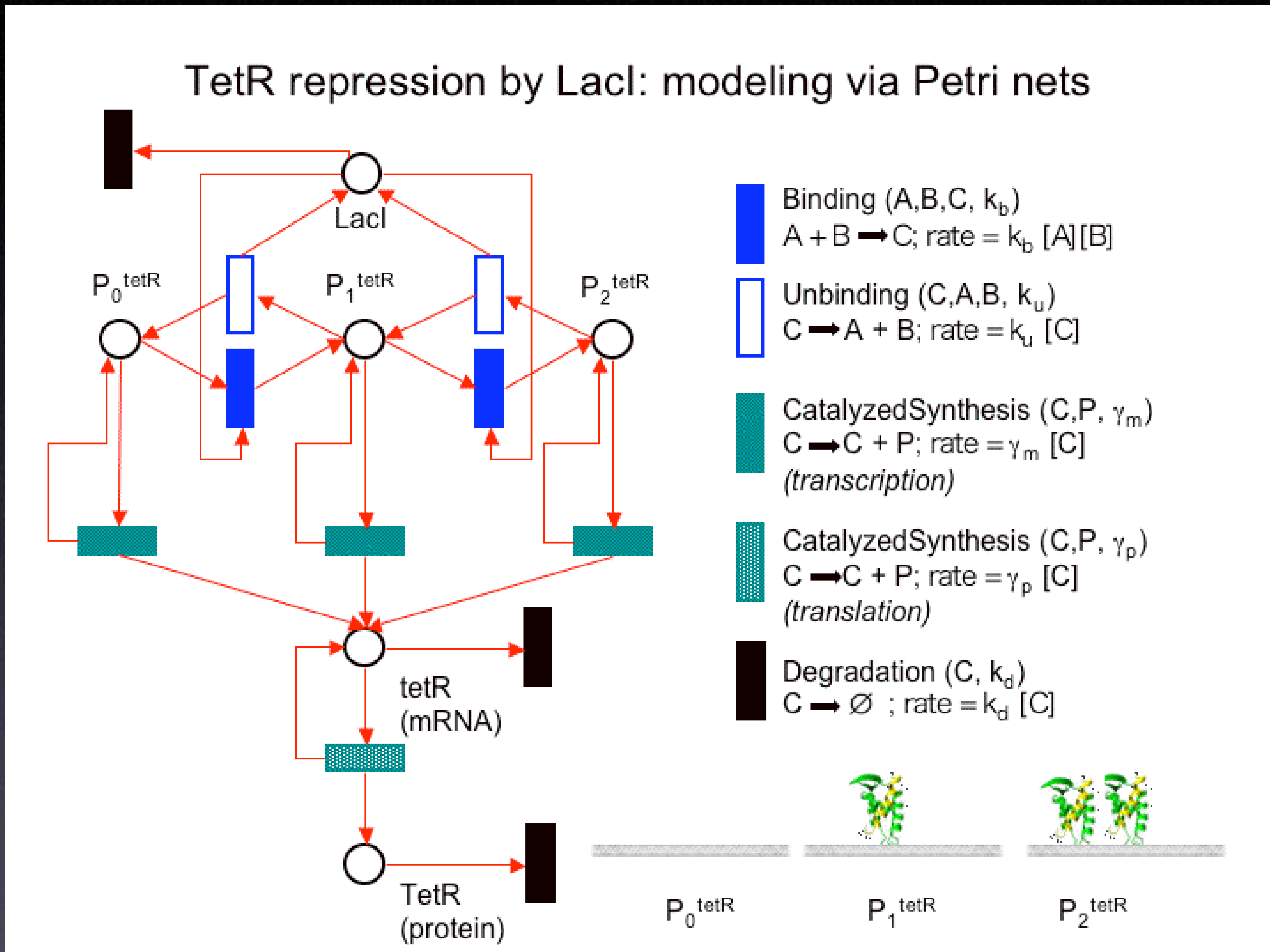
The Repressilator

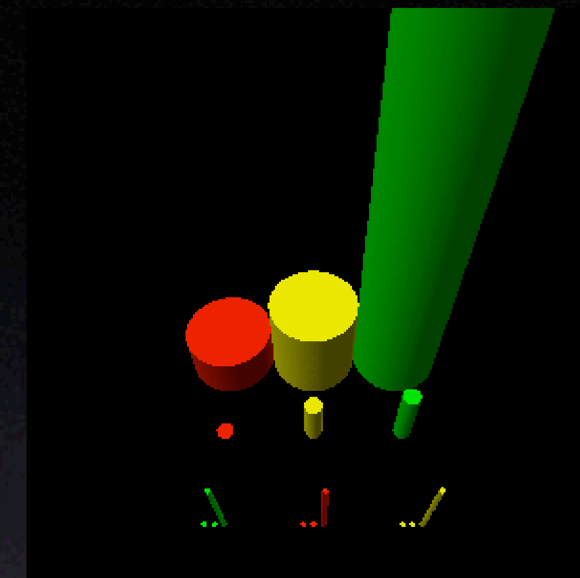
- Repressilator
  - Elowitz & Leibler, Nature 403, 335-338 (2000)
  - Repressor Oscillator
    - engineered synthetic system encoded on a plasmid (introduced into *E. coli*)
    - oscillatory mRNA/protein dynamics from mutually repressing proteins
    - TetR inhibits λ c1 inhibits LacI inhibits TetR (rock-paper-scissors)
  - paper describes both experimental system and mathematical models
    - ODE-based model
    - stochastic, reaction-based model

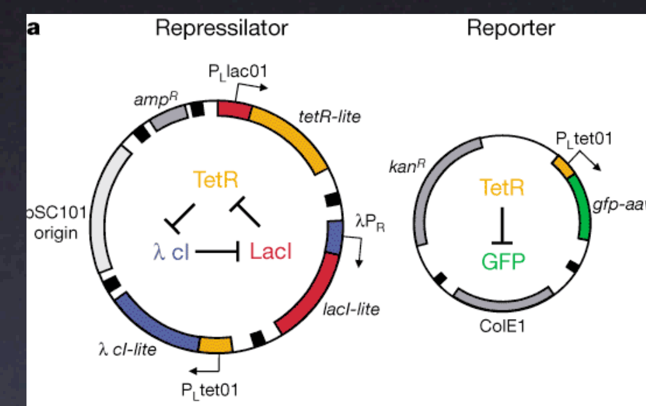


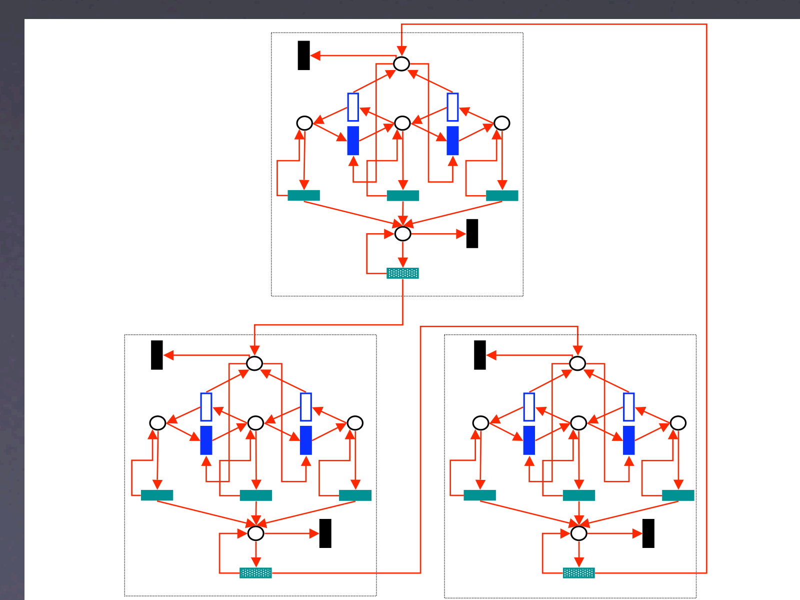


The Repressilator reaction network









$$\frac{dm_i}{dt} = -m_i + \frac{\alpha}{(1 + p_j^n)} + \alpha_0$$

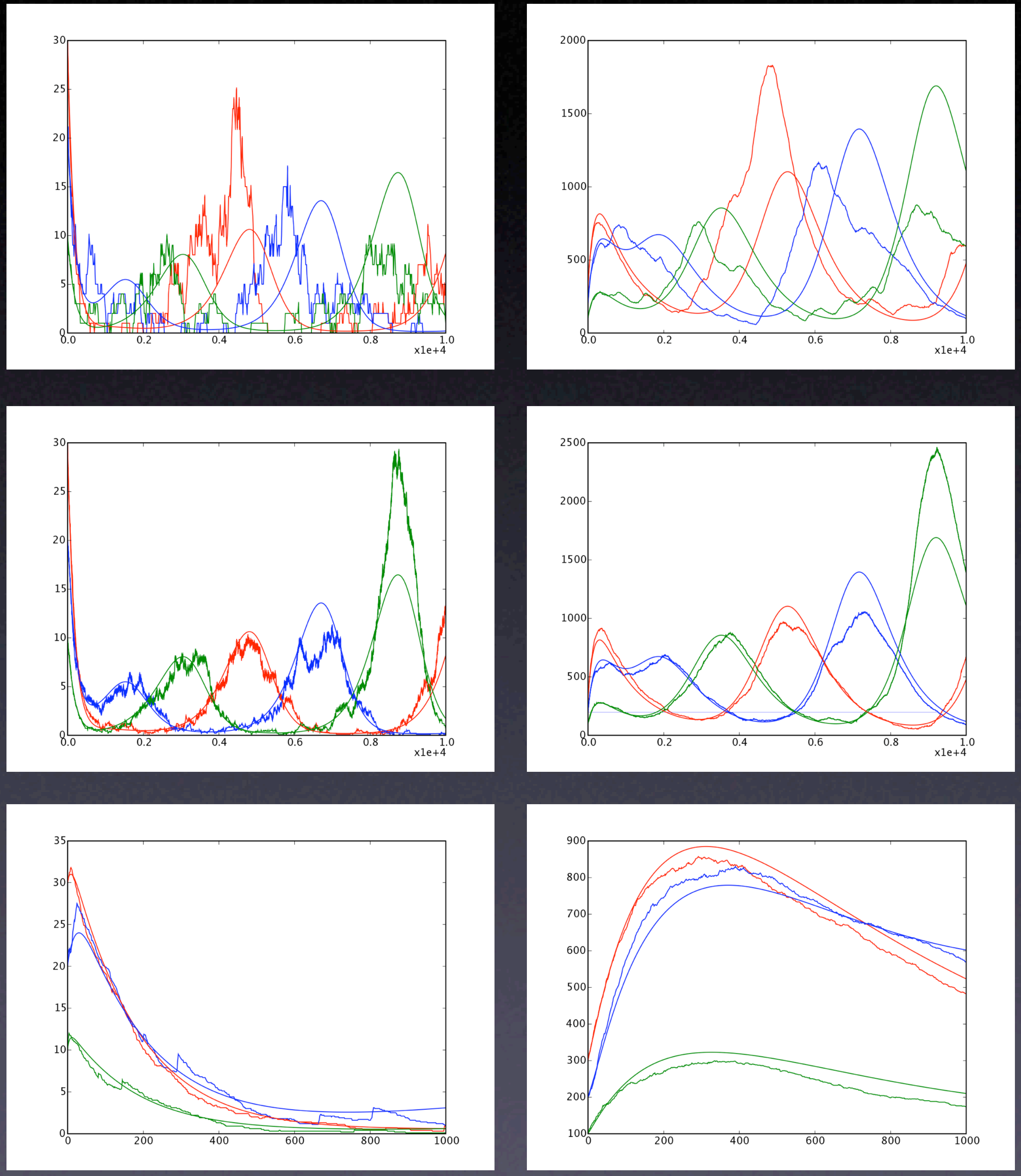
$$\frac{dp_i}{dt} = -\beta(p_i - m_i)$$

$$\begin{pmatrix} i = lacI, tetR, cI \\ j = cI, lacI, tetR \end{pmatrix}$$

coarse-grained
continuum model

Noise in the Repressilator

- *shot noise*
  - fluctuations due to fact that chemical numbers are discrete and potentially small
- *telegraph noise*
  - fluctuations due to fact that some states (e.g., promoter bound by protein) are either *on* or *off*
- can scale parameters in model to accentuate or diminish different types of noise



mRNA

protein

Stochastic/Deterministic Repressilators: the use of inheritance

- Inheritance allows for the definition of families of related classes, distinguished from one another by degrees of specialization
  - *base class* / *superclass*: more generic; *derived class* / *subclass*: more specialized
- Inheritance also allows for code reuse (common behavior can be defined in the superclass)

```
class Repressilator:
    # code to define chemicals & reactions
    def __init__(self, ...):          # initialize a Repressilator
    def AddChemical(self, chemical):        # add a chemical
    def AddReaction(self, reaction):        # add a reaction

class StochasticRepressilator (Repressilator):
    def ComputeReactionRates(self):
        # compute instantaneous rates for Gillespie alg.
    def Step(self, dtmax):
        # implement Gillespie alg. for time up to dtmax
    def Run(self, tmax, delta_t):
        # run Gillespie steps for time up to tmax

class DeterministicRepressilator (Repressilator):
    def dcdt(self, c, t):
        # return right-hand-side for ODE integration
    def Run(self, tmax, dt):
        # integrate ODE for time up to tmax
```