CS 4491/CS 7990
SPECIAL TOPICS IN BIOINFORMATICS

* Some contents are adapted from Dr. Rainer Breitling, University of Manchester, UK

Mingon Kang, Ph.D.
Computer Science, Kennesaw State University
Outline

- Part 1: Why modelling?
- Part 2: The statistical physics of modelling: \[ A \rightarrow B \]
  (where do differential equations come from?)
- Part 3: Translating biology to mathematics
  (finding the right differential equations)
Biology = Concentrations
Humans think small-scale...

(the “7 items” rule)

- phone number length (memory constraint)
- optimal team size (manipulation constraint)
- maximum complexity for rational decision making

...but biological systems contain (at least) dozens of relevant interacting components!
Humans think linear...

...but biological systems contain:

• non-linear interaction between components
• positive and negative feedback loops
• complex cross-talk phenomena
The simplest chemical reaction

\[ A \rightarrow B \]

- irreversible, one-molecule reaction
- examples: all sorts of decay processes, e.g. radioactive, fluorescence, activated receptor returning to inactive state
- any metabolic pathway can be described by a combination of processes of this type (including reversible reactions and, in some respects, multi-molecule reactions)
The simplest chemical reaction

\[ A \rightarrow B \]

various levels of description:

- homogeneous system, large numbers of molecules = ordinary differential equations, **kinetics**
- small numbers of molecules = probabilistic equations, **stochastics**
- spatial heterogeneity = partial differential equations, **diffusion**
Kinetics Description

- Imagine a box containing N molecules. How many will decay during time t? k*N
- Imagine two boxes containing N/2 molecules each. How many decay? k*N
- Imagine two boxes containing N molecules each. How many decay? 2k*N
- In general:

\[ \frac{dn(t)}{dt} = \lambda \cdot n(t) \quad \Leftrightarrow \quad n(t) = N_0 e^{-\lambda t} \]

differential equation (ordinary, linear, first-order) 

exact solution (in more complex cases replaced by a numerical approximation)
Kinetics Description

If you know the concentration at one time, you can calculate it for any other time! (and this really works)
- Consider only decay
- Concentration of Molecule $A = [A]$, usually in units mol/litre (molar)
- Rate constant $= k$, with indices indicating constants for various reactions ($k_1, k_2...$)

Therefore:

$$A \rightarrow B$$

$$\frac{d[A]}{dt} = -\frac{d[B]}{dt} = -k_1[A]$$
1. Simple Decay Reaction

decay <- function(t, y, parms)
{
  k = 1
  list(c(-k * y[1], k * y[1]))
}

out <- ode(y = c(y = c(5, 1)), times = c(1:10), func = decay, parms = NULL)
plot(out)
Decay Reaction in R

**y1**

![Graph y1](image)

**y2**

![Graph y2](image)
Reversible, Single-Molecule Reaction

A ⇌ B, or A → B || B → A, or

Differential equations:

\[
\frac{d[A]}{dt} = -k_1[A] + k_2[B] \\
\frac{d[B]}{dt} = k_1[A] - k_2[B]
\]

Main principle: Partial reactions are independent!
Reversible, single-molecule reaction

Differential Equation:

\[ \frac{d[A]}{dt} = -k_1[A] + k_2[B] \]

\[ \frac{d[B]}{dt} = k_1[A] - k_2[B] \]

Equilibrium (steady-state):

\[ \frac{d[A]_{equi}}{dt} = \frac{d[B]_{equi}}{dt} = 0 \]

\[ -k_1[A]_{equi} + k_2[B]_{equi} = 0 \]

\[ \frac{[A]_{equi}}{[B]_{equi}} = \frac{k_2}{k_1} = K_{equi} \]
isomerisation <- function(t, y, parms)
{
    k1 = 1
    k2 = 0.5
    list(c(-k1 * y[1] + k2 * y[2],
           k1 * y[1] - k2 * y[2]))
}

out <- ode(y = c(y = c(5, 1)), times = c(1:10), func = isomerisation, parms = NULL)
plot(out)
Isomerization Reaction in R

![Graphs of y1 and y2 over time](image)
Isomerization Reaction

If you know the concentration at one time, you can calculate it for any other time... so what would be the algorithm for that?
Irreversible, two-molecule reaction

\[ \text{A} + \text{B} \rightarrow \text{C} \]

Differential equations:

\[
\frac{d[A]}{dt} = \frac{d[B]}{dt} = -\frac{d[C]}{dt}
\]

\[
\frac{d[A]}{dt} = -k[A][B]
\]

Non-linear!

Underlying idea: Reaction probability = Combined probability that both \([A]\) and \([B]\) are in a “reactive mood”:

\[
p(AB) = p(A)p(B) = k_1^*[A]k_2^*[B] = k[A][B]
\]
A simple metabolic pathway

\[ \text{A} \rightarrow \text{B} \leftarrow \text{C} + \text{D} \]

Differential equations:

<table>
<thead>
<tr>
<th>d/dt</th>
<th>decay</th>
<th>forward</th>
<th>reverse</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A]=</td>
<td>−k1 [A]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[B]=</td>
<td>+k1 [A]</td>
<td>−k2 [B]</td>
<td>+k3 [C] [D]</td>
</tr>
<tr>
<td>[C]=</td>
<td></td>
<td>+k2 [B]</td>
<td>−k3 [C] [D]</td>
</tr>
<tr>
<td>[D]=</td>
<td></td>
<td>+k2 [B]</td>
<td>−k3 [C] [D]</td>
</tr>
</tbody>
</table>
Metabolic Networks as Bigraphs

A → B ← C + D

\[
\begin{array}{c|ccc}
 & k1 & k2 & k3 \\
\hline
A & -1 & 0 & 0 \\
B & 1 & -1 & 1 \\
C & 0 & 1 & -1 \\
D & 0 & 1 & -1 \\
\end{array}
\]

\[
\begin{array}{c|cccc}
 & \text{d/dt} & \text{decay} & \text{forward} & \text{reverse} \\
\hline
[A] & -k1 [A] & & & \\
[B] & +k1 [A] & -k2 [B] & +k3 [C] [D] & \\
[C] & & +k2 [B] & -k3 [C] [D] & \\
[D] & & +k2 [B] & -k3 [C] [D] & \\
\end{array}
\]
Biological description ➔ bigraph ➔

differential equations
Biological description → bigraph → differential equations

substance A  ➔  EC 1.1.1.2  ➔  substance B

A ➔ k1 ➔ B
Biological description → bigraph → differential equations
Biological description $\rightarrow$ bigraph $\rightarrow$ differential equations

Fig. courtesy of W. Kolch
Biological description ➔ bigraph ➔ differential equations

Fig. courtesy of W. Kolch
Biological description ➔ bigraph ➔ differential equations

Fig. courtesy of W. Kolch
The Raf-1/RKIP/ERK pathway

Can you model it?
(11x11 table, 34 entries)
Description in MATLAB:
3. The RKIP/ERK pathway

function dydt = erk_pathway_wolkenhauer(t, y)
% from Kwang-Hyun Cho et al., Mathematical Modeling...
    k1 = 0.53;
k2 = 0.0072;
k3 = 0.625;
k4 = 0.00245;
k5 = 0.0315;
k6 = 0.8;
k7 = 0.0075;
k8 = 0.071;
k9 = 0.92;
k10 = 0.00122;
k11 = 0.87;
% continued on next slide...
3. The RKIP/ERK pathway

dydt = [ 
-k1*y(1)*y(2) + k2*y(3) + k5*y(4) 
-k1*y(1)*y(2) + k2*y(3) + k11*y(11) 
k1*y(1)*y(2) - k2*y(3) - k3*y(3)*y(9) + k4*y(4) 
k3*y(3)*y(9) - k4*y(4) - k5*y(4) 
k5*y(4) - k6*y(5)*y(7) + k7*y(8) 
k5*y(4) - k9*y(6)*y(10) + k10*y(11) 
-k6*y(5)*y(7) + k7*y(8) + k8*y(8) 
k6*y(5)*y(7) - k7*y(8) - k8*y(8) 
-k3*y(3)*y(9) + k4*y(4) + k8*y(8) 
-k9*y(6)*y(10) + k10*y(11) + k11*y(11) 
k9*y(6)*y(10) - k10*y(11) - k11*y(11) 
];
Description in MATLAB:

3. The RKIP/ERK pathway

Analysis of the model:

```matlab
>> [t y] = ode45(@erk_pathway_wolkenhauer, [0 10], [2.5 2.5 0 0 0 0 2.5 0 2.5 3 0]); % (initial values!)
>> plot (t, y);
>> legend ('[Raf1*]', '[RKIP]', '[Raf1/RKIP]', '[RAF/RKIP/ERK]', '[ERK]', '[RKIP-P]', '[MEK-PP]', '[MEK-PP/ERK]', '[ERK-PP]', '[RP]', '[RKIP-P/RP] ');
```
The RKIP/ERK pathway in MATLAB