Artificial Biochemistry
Biological Systems as Reactive Systems

Luca Cardelli
Microsoft Research

Computability in Europe, Swansea
2006-07-05
Structural Architecture

Eukaryotic Cell
(10~100 trillion in human body)
Membranes everywhere

Nuclear membrane
Mitochondria
Golgi
Vesicles
E.R.
Plasma membrane (<10% of all membranes)

H. Lodish et al., Molecular Cell Biology, fourth edition p.1
Stochastic Collectives
Stochastic Collectives

• “Collective“:
  - A large set of interacting finite state automata:
    • Not quite language automata (“large set“)
    • Not quite cellular automata (“interacting” but not on a grid)
    • Not quite process algebra (“finite state” and “collective“)
    • Cf. “multi-agent systems” and “swarm intelligence”

• “Stochastic“:
  - Interactions have rates
    • Not quite discrete (hundreds or thousands of components)
    • Not quite continuous (non-trivial stochastic effects)
    • Not quite hybrid (no “switching” between regimes)

• Very much like biochemistry
  - Which is a large set of stochastically interacting molecules/proteins
  - Are proteins finite state and subject to automata-like transitions?
    • Let’s say they are, at least because:
    • Much of the knowledge being accumulated in Systems Biology
      is described as state transition diagrams [Kitano].
State Transitions
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.expasy.ch/cgi-bin/show_thumbnails.pl">http://www.expasy.ch/cgi-bin/show_thumbnails.pl</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Interacting Automata

Communicating automata: a graphical FSA-like notation for “finite state restriction-free π-calculus processes”. Interacting automata do not even exchange values on communication.

The stochastic version has rates on communications, and delays.

“Finite state” means: no composition or restriction inside recursion. Analyzable by standard Markovian techniques, by first computing the “product automaton” to obtain the underlying finite Markov transition system. [Buchholz]
Q: What kind of mass behavior can this produce?
(We need to understand that if want to understand biochemical systems.)
Groupies and Celebrities

**Groupie** (wants to be like somebody different)

- `!a @ A`
- `?b @ B`
- `!b @ A`

**Celebrity** (does not want to be like somebody else)

- `!a @ A`
- `?b @ B`
- `?a @ A`

A stochastic collective of celebrities:

Stable because as soon as a A finds itself in the majority, it is more likely to find somebody in the same state, and hence change, so the majority is weakened.

A stochastic collective of groupies:

Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B’s have plenty of A’s to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

Luca Cardelli
Both Together

A way to break the deadlocks: Groupies with just a few Celebrities

Many Groupies

\( \text{Ga} \) \( \text{Gb} \)
\( !a \) \( ?a \) \( !b \) \( ?b \)

A few (2) Celebrities

\( \text{Ca} \) \( \text{Cb} \)
\( !a \) \( !b \) \( ?a \) \( ?b \)

directive sample 10.0 1000
directive plot Ga(); Gb(); Ca(); Cb()
new a@1.0:chan()
new b@1.0:chan()
let Ca() = do !a; Ca() or ?a; Cb() and Cb() = do !b; Cb() or ?b; Ca()
let Ga() = do !a; Ga() or ?b; Gb() and Gb() = do !b; Gb() or ?a; Ga()
run 1 of (Ca() | Cb())
run 100 of (Ga() | Gb())

A tiny bit of "noise" can make a huge difference

never deadlock
Regularity can arise not far from chaos

Hysteric Groupies

We can get more regular behavior from groupies if they "need more convincing", or "hysteresis" (history-dependence), to switch states.

```
directive sample 10.0 1000
directive plot Ga(); Gb()
new a@1.0:chan()
new b@1.0:chan()
let Ga() = do !a; Ga() or ?b; ?b; ?b;
Gb()
and Gb() = do !b; Gb() or ?a; ?a; ?a;
Ga()
let Da() = !a; Da() and Db() = !b; Db() run 100 of (Ga() | Gb()) run 1 of (Da() | Db())
```

N.B.: It will not oscillate without doping (noise)

"regular" oscillation

A "solid threshold" to observe switching

(With doping to break deadlocks)

A

B

!a

?a

?b

?b

?a

!b

?a

?b

?b

?b

!b
Hysteric 3-Way Groupies

N.B.: It will not oscillate without doping (noise)

directive sample 3.0 1000
directive plot A(); B(); C()
new a@1.0:chan()
new b@1.0:chan()
new c@1.0:chan()

let A() = do !a; A() or ?c; ?c; C() and B() = do !b; B() or ?a; ?a; A()
and C() = do !c; C() or ?b; ?b; B()

let Da() = !a; Da()
and Db() = !b; Db()
and Dc() = !c; Dc()

run 100 of (A() | B() | C())
run 1 of (Da() | Db() | Dc())

1 sample orbit A vs. B,C
Oscillation as Emergence

Just 2 of the hysteric groupies do not oscillate regularly at all!

Without changing the components, interesting properties emerge with a critical size of the population.

Nor 16...

Dotted lines indicate cross sections where one may look for evidence of alternation.

Pretty good with 64...

new a@1.0:chan()
new b@1.0:chan()

let A() = do !a; A() or ?b; ?b; ?b; B() and B() = do !b; B() or ?a; ?a; ?a; A()

let As() = !a; As() and Bs() = !b; Bs()

run 64 of (A() | B())
run 1 of (As() | Bs())
Distributions can be Programmed

Exercise (hard):
Build a small automaton where one state has an occupation distribution like this:

Or, more specifically, build a 3-state, A-B-C, automaton such that:

\[ [B]^* = [B][A]-[C] \]
Semantics of Collective Behavior
“Micromodels”: Continuous Time Markov Chains

- The underlying semantics of stochastic π-calculus (and stochastic interacting automata). Well established in many ways.
  - Automata with rates on transitions.

- “The” correct semantics for chemistry, executable.
  - Gillespie stochastic simulation algorithm

- Lots of advantages
  - Compositional, compact, mechanistic, etc.

- But do not give a good sense of “collective” properties.
  - Yes one can do simulation.
  - Yes one can do program analysis.
  - Yes one can do modelchecking.
  - But somewhat lacking in “analytical properties” and “predictive power”.
They always ask:
- “Yes, but how does you automata model relate to the 75 ODE models in the literature?”

Going from processes/automata to ODEs directly:
- In principle: just write down the Rate Equation: [Calder, Hillston]
  - Determine the set of all possible states $S$ of each process.
  - Determine the rates of the transitions between such states.
  - Let $[S]$ be the “number of processes in state $S$” as a function of time.
  - Define for each state $S$:
    
    $[S]^* = \text{rate of change of the number of processes in state } S$
    Cumulative rate of transitions from any state $S'$ to state $S$, times $[S']$,
    minus cumulative rate of transitions from $S$ to any state $S''$, times $[S]$.

- Intuitive (rate = inflow minus outflow), but often clumsy to write down precisely.

But why go to the trouble?
- If we first convert processes to chemical reactions, then we can convert to ODEs by standard means!
Macromodel of Interaction

Law of Mass Interaction

The speed of interaction is proportional to the number of possible interactions.

Decay

\[ [D]^* = -\lambda [D] \]

\[ [E]^* = \lambda [D] \]

Mass interaction

Interaction Law generalizes Decay Law

\[ [A]^* = -\lambda [A] [B] \]

\[ [B]^* = -\lambda [A] [B] \]

\[ [AB]^* = \lambda [A] [B] \]

Chemical Law of Mass Action

http://en.wikipedia.org/wiki/Chemical_kinetics

The speed of a chemical reaction is proportional to the activity of the reacting substances.

Activity = concentration, for well-stirred aqueous medium

Concentration = number of moles per liter of solution

Mole = 6.022141×10^{23} particles

\([A]_0 = 1000\]

\(\lambda = 1,2,4,8\)

† speed of interaction (formally definable)

= number of interactions over time

not proportional to the number of interacting processes!

[P] is the number of processes P (this is informal; it is only meaningful for a set of processes offering a given action, but a set of such processes can be counted and plotted)
From Chemistry to ODEs
Chemical Reactions

\[ A \rightarrow_r B_1 + \ldots + B_n \]  Degradation  \[ [A]^* = -r[A] \]  Exponential Decay
\[ A_1 + A_2 \rightarrow_r B_1 + \ldots + B_n \]  Asymmetric Collision  \[ [A_i]^* = -r[A_1][A_2] \]  Mass Action Law
\[ A + A \rightarrow_r B_1 + \ldots + B_n \]  Symmetric Collision  \[ [A]^* = -r[A][A-1] \]  Mass Action Law (assuming \( A \neq B_i \neq A_j \) for all \( i,j \))

No other reactions!

Chapter IV: Chemical Kinetics
[David A. Reckhow, CEE 572 Course]
... reactions may be either elementary or non-elementary. Elementary reactions are those reactions that occur exactly as they are written, without any intermediate steps. These reactions almost always involve just one or two reactants. Non-elementary reactions involve a series of two or more elementary reactions. Many complex environmental reactions are non-elementary. In general, reactions with an overall reaction order greater than two, or reactions with some non-integer reaction order are non-elementary.

Enzymatic reactions:
\[ S \underset{E}{\overset{r}{\rightleftharpoons}} P \]
the "r" is given by Michaelis-Menten (approximated steady-state) laws:
\[ E + S \rightleftharpoons ES \]
\[ ES \rightarrow P + E \]

Trimolecular reactions:
\[ A + B + C \rightarrow_r D \]
the measured "r" is an (imperfect) aggregate of e.g.:
\[ A + B \leftrightarrow AB \]
\[ AB + C \rightarrow D \]
From Reactions to ODEs

Caveat: A deterministic approximation of a stochastic system (i.e. possibly misleading)

\[ [X]^* = N \cdot I \]

\[ [A]^* = -l_1 - l_2 \]
\[ [B]^* = -l_1 + l_4 \]
\[ [C]^* = 2l_1 - l_2 - l_3 \]
\[ [D]^* = l_2 \]
\[ [E]^* = l_3 \]
\[ [F]^* = l_3 - 2l_4 \]

Set a rate law for each reaction
(Degradation/Asymmetric/Symmetric)

E.g. \[ [A]^* = -k_1[A][B] - k_2[A][C] \]

Write the coefficients by columns

Read the concentration changes from the rows

<table>
<thead>
<tr>
<th>reactions</th>
<th>( \mathbf{X} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{N} )</td>
<td>( v_1 )</td>
</tr>
<tr>
<td>( A )</td>
<td>-1</td>
</tr>
<tr>
<td>( B )</td>
<td>-1</td>
</tr>
<tr>
<td>( C )</td>
<td>2</td>
</tr>
<tr>
<td>( D )</td>
<td>1</td>
</tr>
<tr>
<td>( E )</td>
<td>1</td>
</tr>
<tr>
<td>( F )</td>
<td>1</td>
</tr>
</tbody>
</table>

Quantity changes

Stoichiometric matrix

Rate laws

Stoichiometric Matrix

Species

Reactions

\( \mathbf{X} \)

\( \mathbf{N} \)

\( \mathbf{A} \)

\( \mathbf{B} \)

\( \mathbf{C} \)

\( \mathbf{D} \)

\( \mathbf{E} \)

\( \mathbf{F} \)

\( A \)

\( B \)

\( C \)

\( D \)

\( E \)

\( F \)

\( k_1 \)

\( k_2 \)

\( k_3 \)

\( k_4 \)

\( A \rightarrow k_1 C+C \)

\( A+C \rightarrow k_2 D \)

\( C \rightarrow k_3 E+F \)

\( F+F \rightarrow k_4 B \)
From Processes to Chemistry
Chemical Ground Form (CGF)

\[ E ::= X_1=M_1, \ldots, X_n=M_n \]
\[ M ::= \pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n \]
\[ P ::= X_1 \mid \ldots \mid X_n \]
\[ \pi ::= \tau_r \ ?n(r) \ !n(r) \]
\[ CGF ::= E,P \]

(To translate chemistry back to processes we need a bit more than simple automata: we may have “+” on the right of \( \rightarrow \), that is we may need “\mid” after \( \pi \).

\( \oplus \) is stochastic choice (vs. + for chemical reactions) 
0 is the null solution (\( P|0 = 0|P = P \))
and null molecule (\( M\oplus0 = 0\oplus M = M \) (\( \tau_0;P = 0 \))
\( X_i \) are distinct in \( E \)
Each name \( n \) is assigned a fixed rate \( r \): \( n(r) \)

Ex: interacting automata
(which are CGFs using “\mid” only in initial conditions):

\[ A = !a;A \oplus ?b;B \]
\[ B = !b;B \oplus ?a;A \]
\[ A\mid A\mid B\mid B \]

Automaton in state A
Automaton in state B
Initial conditions: 2A and 2B
**CGF Semantics**

**Reduction**

\[
\begin{align*}
E, (X_1 | P) & \rightarrow_r E, (P_1 | P) \quad \text{if} \quad E \equiv X_1 = \tau; P_1 \oplus M_1, E' \\
E, (X_1 | X_2 | P) & \rightarrow_r E, (P_1 | P_2 | P) \quad \text{if} \quad E \equiv X_1 = ?n(r); P_1 \oplus M_1, E_1 = X_2 = !n(r); P_2 \oplus M_2, E_2 \\
E, P & \rightarrow_r E'', P'' \quad \text{if} \quad E, P = E', P_1 \quad \text{and} \quad E', P_1 \rightarrow_r E', P_2 \quad \text{and} \quad E', P_2 = E'', P'' \\
\end{align*}
\]

**Structural Congruence**

- \( \equiv \) is an equivalence relation

\[
\begin{align*}
E, E' & \equiv E', E \\
M \oplus M' & \equiv M' \oplus M \\
P | P' & \equiv P' | P \\
E & \equiv E' \land P = P' \Rightarrow E, P = E', P' \\
E & \equiv E' \land M = M' \Rightarrow X = M, E \equiv X = M', E' \\
M & \equiv M' \land P = P' \Rightarrow p; P \oplus M = p; P' \oplus M' \\
P & \equiv P' \Rightarrow X | P = X | P' \\
\end{align*}
\]

\[
E = (A = !a; A \oplus ?b; B \\
B = !b; B \oplus ?a; A)
\]

\[
E, (A|B|B) \rightarrow^{r(a)} E, (A|A|B) \rightarrow^{r(b)} E, (A|B|B) \rightarrow^{r(b)} E, (B|B|B)
\]
Automata to Chemistry

\[
\begin{align*}
A + B &\rightarrow B + B \\
B + A &\rightarrow A + A
\end{align*}
\]

\[
\begin{align*}
A + B_d &\rightarrow B + B_d \\
B + A_d &\rightarrow A + A_d
\end{align*}
\]

\[
\begin{align*}
A + C &\rightarrow C + C \\
C + B &\rightarrow B + B \\
B + A &\rightarrow A + A
\end{align*}
\]

\[
\begin{align*}
A + C_d &\rightarrow C + C_d \\
C + B_d &\rightarrow B + B_d \\
B + A_d &\rightarrow A + A_d
\end{align*}
\]
Three Main Cases

Unary reactions. These are not finite state systems, but finite species systems are ok!

E:\[ X = r; (X | X) \]

C(E):\[ X \rightarrow r X + X \]

That is:\[ A + C \rightarrow 2 \rho \ B + D \]

Binary reactions.
The same interaction can occur multiple times and must be taken into account:

E:\[ A = n; B \oplus n; B \]
\[ C = n; D \]

C(E):\[ A + C \rightarrow \rho(n) B + D \]
\[ A + C \rightarrow \rho(n) B + D \]

That is:\[ A + C \rightarrow 2\rho(n) B + D \]

Symmetric reactions:

E:\[ X = a; 0 \oplus a; Y \]

C(E):\[ X + X \rightarrow 2\rho(a) Y \]

The rate of a was pre-halved and must be restored.
Chemical reactions for $E$:  

(N.B.: $\{\ldots\}^m$ is a multiset, and $P$ is $P$ with all the $|$ changed to $+$)

$Ch_G(E) := \{(X \rightarrow^r P) \text{ s.t. } (X \equiv \tau_r; P \oplus \ldots) \in E\}^m$

$\cup^m \{(X + Y \rightarrow^r P + Q) \text{ s.t. } X \neq Y, (X \equiv ?n_{(r)}; P \oplus \ldots), (Y \equiv !n_{(r)}; Q \oplus \ldots) \in E^2\}^m$

$\cup^m \{(X + X \rightarrow^{2r} P + Q) \text{ s.t. } (X \equiv ?n_{(r)}; P \oplus \ldots \equiv !n_{(r)}; Q \oplus \ldots) \in E\}^m$

Initial conditions for $P$:  

$Ch_G(P) := P$
From Processes to ODEs
Nonlinear Transitions
Basic Nonlinear Transition

\[ A = ?c(s) ; B \]
\[ B = !c(s) ; B \]

\[ A+B \rightarrow^s B+B \]

\[ [A]^* = -s[A][B] \]
\[ [B]^* = s[A][B] \]

N.B.: needs at least 1 B to “get started”.

Matlab

\[
\begin{align*}
\text{Matlab} & \\
\text{interval/step} & [0.0001:0.0] \\
(A) & \text{ode}1 (+ x)= -x^2 \times 2 1000.0 \\
(B) & \text{ode}2 (+ x)= x^2 \times 2 1.0
\end{align*}
\]
Bell Exercise

Build a \textit{small} network where one node has a distribution like \( B() \):

\[ [B]^* = [B][A]-[C] \]

\[ A = ?b(1);B \]
\[ B = !b(1);B \oplus ?c(1);C \]
\[ C = !c(1);C \]

\[ A+B \rightarrow B+B \]
\[ B+C \rightarrow C+C \]

\( [A]^* = -[A][B] \)
\( [B]^* = [A][B]-[B][C] \)
\( [C]^* = [B][C] \)

\begin{align*}
\text{directives} & \text{ sample } 0.0025 \ 1000 \\
\text{directives} & \text{ plot } B(): \ A(); \ C() \\
\text{new} & \text{ b}@1.0: \text{chan} \ \text{new} \ c}@1.0: \text{chan} \\
\text{let} \ A() &= \ ?b; \ B() \\
\text{and} \ B() &= \text{do } !b; \ B() \text{ or } ?c; \ C() \\
\text{and} \ C() &= \ !c; C() \\
\text{run } ((10000 \ \text{of} \ A()) | B() | C())
\end{align*}
Oscillator

```plaintext
directive sample 0.1 1000
directive plot A1(); A2(); A3()
val r=1.0 val s=1.0
new a1@s:chan new a2@s:chan new a3@s:chan
let A1() = do !a1;A1() or delay@r;A2() or ?a2;
       ?a2; A2()
and A2() = do !a2;A2() or delay@r;A3() or ?a3;
       ?a3; A3()
and A3() = do !a3;A3() or delay@r;A1() or ?a1;
       ?a1; A1()
run 1000 of A1()

N.B. this does not deadlock!
```

A = !a(s); A ⊕ r; B ⊕ ?b(s); A'
A' = ?b(s); B
B = !b(s); B ⊕ r; C ⊕ ?c(s); B'
B' = ?c(s); C
C = !c(s); C ⊕ r; A ⊕ a(s); C'
C' = ?a(s); A

[A'] = -s[A'][B] + s[A][B]
[B'] = -s[B'][C] + s[B][C]
[C'] = -s[C'][A] + s[C][A]

Robust Stochastic Oscillation

Sustained Deterministic Oscillation

Matlab

continuous_sys_generator
Epidemics


http://mathworld.wolfram.com/Kermack-MckendrickModel.html
Epidemics

Developing the Use of Process Algebra in the Derivation and Analysis of Mathematical Models of Infectious Disease

R. Norman and C. Shankland
Department of Computing Science and Mathematics, University of Stirling, UK.
{ces,ran}@cs.stir.ac.uk

Abstract. We introduce a series of descriptions of disease spread using the process algebra WSCCS and compare the derived mean field equations with the traditional ordinary differential equation model. Even the preliminary work presented here brings to light interesting theoretical questions about the “best” way to defined the model.
\[ S = ?i_{(t)}; I \]
\[ I = !i_{(t)}; I \oplus ?i_{(t)}; I \oplus \tau_r; R \]
\[ R = ?i_{(t)}; R \]

\[ S + I \rightarrow^{+} I + I \]
\[ I + I \rightarrow^{+} I + I \]
\[ I \rightarrow^{\tau} R \]
\[ R + I \rightarrow^{+} R + I \]

\[ [S]^* = -t[S][I] \]
\[ [I]^* = t[S][I] - r[I] \]
\[ [R]^* = r[I] \]

Automata match the standard ODE model!

(the Kermack-McKendrick, or SIR model)
Simplified Model

\[ S = \Delta i(t) ; I \]
\[ I = \Delta i(t) ; I \oplus \tau ; R \]
\[ R = 0 \]

\[ S + I \rightarrow^+ I + I \]
\[ I \rightarrow^r R \]

\[ [S]^* = -t[S][I] \]
\[ [I]^* = t[S][I] - r[I] \]
\[ [R]^* = r[I] \]

Not totally obvious that one could have simplified the automata model.

Same ODE, hence equivalent automata models.

---

directive sample 500.0 1000
directive plot Recovered(): Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03

let Recovered() = ()

and Susceptible() = ?infect: Infected()

and Infected() = do !infect; Infected() or delay@recover; Recovered()

run (200 of Susceptible() | 2 of Infected())
Lotka-Volterra
Predator-Prey

Since predator and prey drive each other to extinction (stochastically), we restart the populations periodically.

(This is a case where the continuous system oscillates and the stochastic one does not! We have seen examples of the opposite situation.)
ODE

\[ H = \tau_b; (H|H) \oplus ?c_{(p)};0 \]
\[ C = \tau_m;0 \oplus !c_{(p)};(C|C) \]

\[ H \rightarrow^b H + H \]
\[ C \rightarrow^m 0 \]
\[ H + C \rightarrow^p C + C \]

\[ [H]^* = b[H]-p[H][C] \]
\[ [C]^* = -m[C]+p[H][C] \]

Lotka-Volterra Equations

The Lotka-Volterra equations describe an ecological predator-prey (or parasite-host) model which assumes that, for a set of fixed positive constants \(A\) (the growth rate of prey), \(B\) (the rate at which predators destroy prey), \(C\) (the death rate of predators), and \(D\) (the rate at which predators increase by consuming prey), the following conditions hold:

1. A prey population \(x\) increases at a rate \(dx = Ax \, dt\) (proportional to the number of prey) but simultaneously destroyed by predators at a rate \(dx = -Bxy \, dt\) (proportional to the product of the numbers of prey and predators).

2. A predator population \(y\) decreases at a rate \(dy = -Cy \, dt\) (proportional to the number of predators), but increases at a rate \(dy = Dxy \, dt\) (again proportional to the product of the numbers of prey and predators).

This gives the coupled differential equations:

\[
\frac{dx}{dt} = Ax - Bxy \tag{1}
\]
\[
\frac{dy}{dt} = -Cy + Dxy \tag{2}
\]

Automata match the Lotka-Volterra model (with \(B=D\)
Laws by ODEs
Choice Law by ODEs

\[ \tau_\Lambda;B \oplus \tau_\mu;B = \tau_{\Lambda+\mu};B \]

\[ A = \tau_\Lambda;B \oplus \tau_\mu;B \]

\[ A \overset{\Lambda}{\rightarrow} B \]

\[ A \overset{\mu}{\rightarrow} B \]

\[ [A]^* = -\Lambda[A] - \mu[A] \]

\[ [B]^* = \Lambda[A] + \mu[A] \]

\[ A = \tau_{\Lambda+\mu};B \]

\[ A \overset{\Lambda+\mu}{\rightarrow} B \]

\[ [A]^* = -(\Lambda+\mu)[A] \]

\[ [B]^* = (\Lambda+\mu)[A] \]
Idle Delay Law by ODEs

\[ A = \tau_\lambda; A \oplus \tau_\mu; B = A = \tau_\mu; B \]

\[ A = \tau_\lambda; A \oplus \tau_\mu; B \]

\[ A = \tau_\mu; B \]

\[ A \rightarrow^\lambda A \]

\[ A \rightarrow^\mu B \]

\[ [A]^* = -\mu[A] \]

\[ [B]^* = \mu[A] \]
Idle Interaction Law by ODEs

It may seem like \( A \) should decrease half as fast, but NO! Two ways to explain:
- State \( A \) is *memoryless* of any past idling.
- Activity on \( c \) is double

\[
A = ?c;B \\
C = !c;C
\]

\[
A + C \rightarrow_r B + C
\]

\[
[A]^* = -r[A][C] \\
[B]^* = r[A][C] \\
[C]^* = 0
\]

\[
A = ?c;A \oplus ?c;B \\
C = !c;C
\]

\[
A + C \rightarrow_r A + C \\
A + C \rightarrow_r B + C
\]

\[
[A]^* = -r[A][C] \\
[B]^* = r[A][C] \\
[C]^* = 0
\]
Asynchronous Interleaving

\[ \tau_{\lambda}; B \mid \tau_{\mu}; D = \tau_{\lambda}; (B \mid \tau_{\mu}; D) + \tau_{\mu}; (\tau_{\lambda}; B \mid D) \]

Amazingly, the B’s and the D’s from the two branches sum up to exponential distributions
Asynchronous Interleaving Law by ODEs

\[ \tau_{\lambda}; B \mid \tau_{\mu}; D = \tau_{\lambda}; (B \mid \tau_{\mu}; D) + \tau_{\mu}; (\tau_{\lambda}; B \mid D) \]

Want to show that B and D on both sides have the “same behavior” (equal quantities of B and D produced at all times)

\[ [B] = \lambda[A_1] \]
\[ [D] = \mu[C_1] \]

Want to show that B and D on both sides have the “same behavior” (equal quantities of B and D produced at all times)

\[ [B] = \lambda[A_1] \]
\[ [D] = \mu[C_1] \]

So, for example, if we run a stochastic simulation of the left hand side with 1000*A1 and 1000*C1, we obtain the same curves for B and D than a stochastic simulation of the right hand side with 1000*Y.
Parametric Form
Chemical Parametric Form (CPF)

\[ E ::= X_1(p_1)=M_1, \ldots, X_n(p_n)=M_n \]

Definitions \hspace{1cm} (n \geq 0)

\[ M ::= \pi_1; \top_1 \oplus \ldots \oplus \pi_n; \top_n \]

Molecules \hspace{1cm} (n \geq 0)

\[ P ::= X_1(p_1) \mid \ldots \mid X_n(p_n) \]

Solutions \hspace{1cm} (n \geq 0)

\[ \pi ::= \tau_r ?n(p) !n(p) \]

Interactions

\[ \text{CPF} ::= E, P \]

with initial conditions

⊕ is stochastic choice (vs. + for chemical reactions)

0 is the null solution (P|0 = 0|P = P)

and null molecule (M⊕0 = 0⊕M = M) (τ_0; P = 0)

\(X_i\) are distinct in \(E\), \(p\) are vectors of names

\(p\) are vectors of distinct names when in binding position

Each free name \(n\) in \(E\) is assigned a fixed rate \(r\):

written either \(n(r)\), or \(\rho_{\text{CPF}}(n)=r\).

A translation from CPF to CGF exists

(expanding all possible instantiation of parameters from the initial conditions)

An incremental translation algorithm exists

(expanding on demand from initial conditions)
Repressilator ODEs

Neg(a,b) = ?a; Inh(a,b) ⊕ τε: (Tr(b) | Neg(a,b))
Inh(a,b) = τη; Neg(a,b)
Tr(b) = !b; Tr(b) ⊕ τ; 0
Neg(x(r),y(r)) | Neg(y(r),z(r)) | Neg(z(r),x(r))

simplifying (N is the quantity of each of the 3 gates)

[Neg/x,y]* = ηN - (η+r[Tr/x])[Neg/x,y]
[Neg/y,z]* = ηN - (η+r[Tr/y])[Neg/y,z]
[Neg/z,x]* = ηN - (η+r[Tr/z])[Neg/z,x]
[Tr/x]* = ε[Neg/z,x] - γ[Tr/x]
[Tr/y]* = ε[Neg/x,y] - γ[Tr/y]
[Tr/z]* = ε[Neg/y,z] - γ[Tr/z]

No sustained oscillations (with SPiM parameters). But see Elowitz&Leibler.

Matlab
continuous_sys_generator

SPiM
No sustained oscillations (with SPiM parameters). But see Elowitz&Leibler.
Groupies ODE
Doped Groupies ODE

Q: What does this do?

\[ A = !a(r); A \oplus ?b(r); B \]
\[ B = !b(r); B \oplus ?a(r); A \]
\[ A_d = !a_r; A_d \]
\[ B_d = !b_r; B_d \]

\[ A + B \rightarrow^r A + A \]
\[ B + A \rightarrow^r B + B \]
\[ A + B_d \rightarrow^r B + B_d \]
\[ B + A_d \rightarrow^r A + A_d \]


\[ [A_d]^* = 0 \]
\[ [B_d]^* = 0 \]

\[ [A_d], [B_d] \] are constant; assume them both = k

ODE predicts converging stable equilibrium at \([A]=[B]\) instead of the total chaos observed in the stochastic system!

For \(k=0\) (no dope), predicts deadlock \([A]^*=[B]^*=0\) but at any value of \([A]\), which is definitely not true in the stochastic system.
Conclusions
Conclusions

- **Stochastic Collectives**
  - Complex global behavior from simple components
  - Emergence of collective functionality from “non-functional” components
  - (Cf. “swarm intelligence”: simple global behavior from complex components)

- **Artificial Biochemistry**
  - Stochastic collectives with Law of Mass Interaction kinetics
  - Connections to classical Markov theory, chemical Master Equation, and Rate Equation

- **Properties of collective behavior**
  - Simulation
  - Systematic translation to ODEs from parametric process “libraries”
  - Correspondence (or not) between stochastic and deterministic behavior

- **Interdisciplinary connections**
  - Process descriptions vs. chemical descriptions
  - Process descriptions vs. ODE descriptions