Morphisms of Reaction Networks

Luca Cardelli, Microsoft Research & Oxford University

with: Mirco Tribastone, Max Tschaikowski, Andrea Vandin
IMT Institute for Advanced Studies, Lucca

Attila Csikász-Nagy
King’s College London

Neil Dalchau
Microsoft Research Cambridge

CMSB 2016-09-22
Outline

• Computational Methods
  • Comparing Networks
  • Network Bisimulations (and Morphisms)
  • Finding Bisimulations by Theorem Proving

• Systems Biology
  • Morphisms of Antagonistic Networks
  • Network Morphisms as Evolutionary Paths
  • Noise Reduction in Complex Biochemical Switches
Comparing Networks

- High-value activity:
  - 2001 Nobel prize in Physiology for the discovery of “Key regulators of the cell cycle ... they have identified key molecules that regulate the cell cycle in all eukaryotic organisms, including yeast, plants, animals, and human.”
  - These are not the same molecules in all organisms, but it is still “the same network”

- Network differences expose evolution
  - Tracing back ancestral networks from current ones

- Networks are algorithms
  - Algorithms fall in different performance classes (is nature “optimal”?)
  - Different networks for the same function may or may not be in the same class
Morphing networks

• How can we compare different networks?
  • Different number of species
  • Different number of reactions
  • Apparently unrelated connectivity

• So that we can compare their function?
  • Does antagonism (in network structure) guarantee bistability (in function)?

• We do it by *mapping* networks onto one another so that they *emulate* each other (‘s traces)
  • Deterministic version of simulation of reactive systems
Mapping one network into another

- A formal notion was strangely missing from the literature
  - Seen in Biology: single-network analysis (e.g. structure of feedback loops) and network reduction (e.g. while preserving steady states). Study of common or frequent subnetworks.
  - Seen in C.S.: comparing network behaviors (e.g. morphisms of event structures).
  - Nothing much resembling (bi)simulation “on the syntax” (structure) of whole biochemical networks.

- Model reduction is unavoidable and pervasive, but
  - Often criticized/ignored by biologists when it leads to quantities that are “not biologically meaningful”. E.g. a fusion or change a variables in the ODEs where the new variables do not correspond to biological parts. The reduced model should “inform” the original one.

- Science’s ethos
  - The “truth” is the big network, not the small one! If you depart from the truth in any way, you have to explain how you can get back to it.
  - The point is not to reduce the size of the network (although that’s neat), but to understand aspects of the big network by reference to a smaller one.
  - The mapping is more important than either networks.

Norbert Wiener
Pioneer of stochastic processes and inventor of Cybernetics.

“The best material model of a cat is another, or preferably the same, cat”
Chemical Reaction Networks

\[ A + C \rightarrow^{\alpha} C + E \]
\[ B + C \rightarrow^{\alpha} C + E \]
\[ C \rightarrow^{\beta} A \]
\[ D \rightarrow^{\beta} B \]

\[ \dot{V}_A = -\alpha V_A V_C + \beta V_C \]
\[ \dot{V}_B = -\alpha V_B V_C + \beta V_D \]
\[ \dot{V}_C = -\beta V_C \]
\[ \dot{V}_D = -\beta V_D \]
\[ \dot{V}_E = \alpha V_A V_C + \alpha V_B V_C \]

The (autonomous) ODE system \( \dot{V} = F(V) \) underlying a CRN \((S,R)\) is \( F : \mathbb{R}_{\geq 0}^S \rightarrow \mathbb{R}^S \), where each component \( F_X \), with \( X \in S \) is defined as:

\[ F_X(V) := \sum_{\rho \rightarrow X \in R} (\pi(X) - \rho(X)) \cdot \alpha \cdot \prod_{V \in X} V^\rho(V) . \]

This represents the well-known mass-action kinetics, where the reaction rate is proportional to the concentrations of the reactants involved. Since the ODE system of a CRN is given by polynomials, the vector field \( F \) is locally Lipschitz. Hence, the theorem of Picard-Lindelöf ensures that for any \( V(0) \in \mathbb{R}_{\geq 0}^S \) there exists a unique non-continuable solution of \( \dot{V} = F(V) \).
Behavior

directive sample 3.0 100
directive simulation deterministic
directive plot A; B; C; D; E

rate a = 1;
rate b = 2;

init A 1 |
init B 3 |
init C 1 |
init D 3 |
init E 0 |

A + C ->{a} C + E |
B + C ->{a} C + E |
C ->{b} A |
D ->{b} B
Network Bisimulation
A Bisimulation Approach

- For discrete transition systems
  - Nondeterministic: If two systems are in “equivalent” states, and one system can step from one state to another, then the other system can make a similar step and end up in an “equivalent” state. And vice-versa.
  - Stochastic: If two systems are in “equivalent” states, and one system can step from one state to an equivalence class of states (with some collective probability), then the other system can make a similar step and end up again in an “equivalent” equivalence class of states. And vice-versa.
  - Syntactic characterizations (bisimulation is definable over Process Algebras rather than their state spaces).

- For continuous transition systems
  - Continuous: If two systems are in “equivalent” states (e.g. identical states (BB), or up to sum of variables (FB)), and one system takes an infinitesimal step into another state, then the other system can take a similar infinitesimal step and end up in the “equivalent” state. And vice-versa.
  - Defined on traces: no syntactic characterization.

- What we contribute:
  - We define bisimulation (actually two of them) over a syntax for continuous transition systems, where the syntax is that of CRNs.
  - This allows us to both compare and minimize CRNs, via fast algorithms based on partition refinement (Tarjan - CONCUR) or theorem proving (Tarski - POPL).
Forward Bisimulation

- Consider a partition (lumping) of species: \( \{\{A, B\}, \{C\}, \{D\}, \{E\}\} \)
- It may induce a collapse of the CRN:

\[
\begin{align*}
AB + C & \rightarrow^\alpha C + E \\
C & \rightarrow^\beta AB \\
D & \rightarrow^\beta AB
\end{align*}
\]

In the sense that AB represents A+B
Reduction works for that partition

Original CRN, plotting A+B

directive sample 3.0 100
directive simulation deterministic
directive plot sum(A; B); C; D; E

rate a = 1;
rate b = 2;

init A 1 |
init B 3 |
init C 1 |
init D 3 |
init E 0 |

A + C ->(a) C + E |
B + C ->(a) C + E |
C ->(b) A |
D ->(b) B

Reduced CRN with AB₀ = A₀ + B₀

directive sample 3.0 100
directive simulation deterministic
directive plot AB; C; D; E

rate a = 1;
rate b = 2;

init AB 4 |
init C 1 |
init D 3 |
init E 0 |

AB + C ->(a) C + E |
C ->(b) AB |
D ->(b) AB
Because it works on the ODEs

- We can consider $AB = A + B$ and express the system just in terms of $AB$, dropping $A$ and $B$

\[
\begin{align*}
\dot{V}_{AB} &= \dot{V}_A + \dot{V}_B = -\alpha V_{AB} V_C + \beta V_C + \beta V_D \\
\dot{V}_C &= -\beta V_C \\
\dot{V}_D &= -\beta V_D \\
\dot{V}_E &= \alpha V_{AB} V_C
\end{align*}
\]

- And these are the ODEs of the reduced CRN
When does it work, in general?

- A partition $H$ of the ODE (variables) is an (ordinary-) lumping if one can derive an ODE for the partition from the ODE of the original system, in terms of sums of the variables in the partition.

  > Definition 2 (Ordinary fluid lumpability). Let $(S, R)$ be a CRN, $F$ be its vector field, and $H = \{H_1, \ldots, H_n\}$ a partition of $S$. Then, $H$ is ordinary fluid lumpable if for all $H \in H$ there exists a polynomial $\psi_H$ in $|H|$ variables such that $\sum_{X \in H} F_X(V) = \psi_H(\sum_{X \in H_1} V_X, \ldots, \sum_{X \in H_n} V_X)$ for all $V \in \mathbb{R}^{|S|}_+$. 

- Thm: A partition of CRN species that is a Forward Bisimulation is an ordinary lumping of the corresponding ODEs.

- A partition of CRN species is a Forward Bisimulation if the fluxes of the CRN match up in a certain way (checkable just by looking at the CRN, not its ODEs):

  > Definition 7 (Reaction and production rates). Let $(S, R)$ be a CRN, $X, Y \in S$, and $\rho \in MS(S)$. The $p$-reaction rate of $X$, and the $p$-production rate of $Y$-elements by $X$ are defined respectively as

$$err[X, \rho] := (\rho(X) + 1) \sum_{X+rho \rightarrow Y \in R} \alpha, \quad pr(X, \rho, Y) := (\rho(X) + 1) \sum_{X+rho \rightarrow Y \in R} \alpha \cdot \pi(Y)$$

Finally, for $H \subseteq S$ we define $pr[X, \rho, H] := \sum_{Y \in H} pr(X, \rho, Y)$.

  > Definition 8 (Forward CRN Bisimulation). Let $(S, R)$ be a CRN, $\mathcal{R}$ an equivalence relation over $S$ and $H = S/\mathcal{R}$. Then, $\mathcal{R}$ is a forward CRN bisimulation (abbreviated FB) if for all $(X, Y) \in \mathcal{R}$, all $\rho \in MS(S)$, and all $H \in H$ it holds that

$$err[X, \rho] = err[Y, \rho] \quad \text{and} \quad pr[X, \rho, H] = pr[Y, \rho, H]$$

Forward and Backward Bisimulations for Chemical Reaction Networks.
Luca Cardelli, Mirco Tribastone, Max Tschaikowski, Andrea Vandin. [CONCUR’15]

Comparing Chemical Reaction Networks: A Categorical and Algorithmic Perspective.
Luca Cardelli, Mirco Tribastone, Max Tschaikowski, Andrea Vandin. [LICS’16]
Backward Bisimulation

- Consider a partition (lumping) of species: 
  \{\{A, B\}, \{C, D\}, \{E\}\}
- It may induce a collapse of the CRN:

\[
\begin{align*}
AB + CD & \rightarrow_\alpha CD + 2E \\
CD & \rightarrow_\beta AB
\end{align*}
\]

In the sense that AB represents A and B equally
Reduction works for that partition

**Original CRN**, setting \(A_0 = B_0, \ C_0 = D_0\)

| Rate a = 1; | Rate b = 2; |
| init A 1 | init B 1 |
| init C 3 | init D 3 |
| init E 0 |

\[ A + C \rightarrow \{a\} C + E | B + C \rightarrow \{a\} C + E | C \rightarrow \{b\} A | D \rightarrow \{b\} B \]

**Reduced CRN** with \(AB_0 = A_0 = B_0, CD_0 = C_0 = D_0\)

| Rate a = 1; | Rate b = 2; |
| init AB 1 | init CD 3 |
| init E 0 |

\[ AB + CD \rightarrow \{a\} CD + 2 E | CD \rightarrow \{b\} AB \]
Because it works on the ODEs

- If \( V_A(0) = V_B(0) \) and \( V_C(0) = V_D(0) \) then \( V_A(t) = V_B(t) \) and \( V_C(t) = V_D(t) \)

\[
\begin{align*}
\dot{V}_A &= -\alpha V_A V_C + \beta V_C \\
\dot{V}_C &= -\beta V_C \\
\dot{V}_E &= 2\alpha V_A V_C
\end{align*}
\]

- And these are the ODEs of the reduced CRN
When does it work, in general?

- A partition \( H \) of the ODE (variables) is an (exact-) lumping if the derivatives are equal in each partition whenever the concentrations are equal in each partition.

- Thm: A partition of CRN species that is a Backward Bisimulation is an exact lumping of the corresponding ODEs.

- A partition of CRN species is a Backward Bisimulation if the fluxes of the CRN match up in a certain way (checkable just by looking at the CRN, not its ODEs):

\begin{align*}
\text{Definition 4 (Exact fluid lumpability).} \quad & \text{Let } (S, R) \text{ be a CRN, } F \text{ its vector field, and } H \text{ a partition of } S. \text{ We call } V \in \mathbb{R}^S \text{ constant on } H \text{ if } V_{X_i} = V_{X_j} \text{ for all } H \in \mathcal{H}, \text{ and all } X_i, X_j \in H. \text{ Then, } H \text{ is exactly fluid lumpable if } F(V) \text{ is constant on } H \text{ whenever } V \text{ is constant on } H. \\
\text{Definition 13 (Cumulative flux rate).} \quad & \text{Let } (S, R) \text{ be a CRN, } X \in S, \rho \in MS(S), \text{ and } \mathcal{M} \subseteq MS(S). \text{ Then, we define} \\
& \begin{align*}
\text{fr}(X, \rho) & := \sum_{\rho \xrightarrow{\alpha} \rho'' \in H} (\pi(X) - \rho(X)) \cdot \alpha, \\
\text{fr}[X, \mathcal{M}] & := \sum_{\rho \in \mathcal{M}} \text{fr}(X, \rho).
\end{align*}
\end{align*}

We call \( \text{fr}(X, \rho) \) and \( \text{fr}[X, \mathcal{M}] \) \( \rho \)-flux rate and cumulative \( \mathcal{M} \)-flux rate of \( X \), respectively.

\begin{align*}
\text{Definition 14 (Backward CRN bisimulation).} \quad & \text{Let } (S, R) \text{ be a CRN, } \mathcal{R} \text{ an equivalence relation over } S, \mathcal{H} = S/\mathcal{R} \text{ and } \mu \text{ the choice function of } \mathcal{H}. \text{ Then, } \mathcal{R} \text{ is a backward CRN bisimulation (BB) if for any } (X, Y) \in \mathcal{R}, \text{ it holds that} \\
& \begin{align*}
\text{fr}[X, \mathcal{M}] = \text{fr}[Y, \mathcal{M}] & \quad \text{for all } \mathcal{M} \in \{ \rho | \rho \xrightarrow{\alpha} \pi \in R \} / \equiv_{\mathcal{H}}, \quad (2) \\
\text{where any two } \rho, \sigma \in MS(S) \text{ satisfy } \rho \equiv_{\mathcal{H}} \sigma \text{ if } \mu(\rho) = \mu(\sigma).
\end{align*}
\end{align*}
Applications of Bisimulation

- Model Reduction
  - Find reduced networks
  - Compute quotient CRNs
  - Find network symmetries that may be of biological interest

- Morphism Generation
  - Find morphisms between networks (e.g. all the ones for a fixed rate assignment)

---

**Benchmarks from Sneddon et al., Nature Methods, 2011**

<table>
<thead>
<tr>
<th>Model</th>
<th>Reactions</th>
<th>Species</th>
<th>FB</th>
<th>Time (s)</th>
<th>BB</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e9</td>
<td>3538944</td>
<td>262146</td>
<td>222</td>
<td>4.61E+4</td>
<td>222</td>
<td>7.65E+4</td>
</tr>
<tr>
<td>e8</td>
<td>786432</td>
<td>65538</td>
<td>167</td>
<td>1.92E+3</td>
<td>167</td>
<td>3.68E+3</td>
</tr>
<tr>
<td>e7</td>
<td>172032</td>
<td>16386</td>
<td>122</td>
<td>8.15E+1</td>
<td>122</td>
<td>1.77E+2</td>
</tr>
<tr>
<td>e6</td>
<td>36864</td>
<td>4098</td>
<td>86</td>
<td>3.00E+0</td>
<td>86</td>
<td>7.29E+0</td>
</tr>
<tr>
<td>e5</td>
<td>7680</td>
<td>1026</td>
<td>58</td>
<td>1.54E-1</td>
<td>58</td>
<td>4.06E-1</td>
</tr>
<tr>
<td>e4</td>
<td>1536</td>
<td>258</td>
<td>37</td>
<td>9.00E-3</td>
<td>37</td>
<td>1.09E-1</td>
</tr>
<tr>
<td>e3</td>
<td>288</td>
<td>66</td>
<td>22</td>
<td>1.00E-3</td>
<td>22</td>
<td>3.00E-3</td>
</tr>
<tr>
<td>e2</td>
<td>48</td>
<td>18</td>
<td>12</td>
<td>1.00E-3</td>
<td>12</td>
<td>2.00E-3</td>
</tr>
</tbody>
</table>

---

Concur 2015
How does it work?

- **Partition refinement!**
  - Start from the coarsest partition: \{\{A, B, C, D, E\}\}
  - Thm: There is always a coarsest FB or BB partition
  - Find a reason why that partition is *not* an FB or BB (e.g., ask Z3)
  - Split the partition: \{\{A, B, C\}, \{D, E\}\} (this is the clever part)
  - Iterate
  - In the worst case we end up with \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}\}

- **Customizable**
  - If we know that we want to observe A separately, we can start the algorithm e.g. with the partition \{\{A\}, \{B, C, D, E\}\}
Finding Network Bisimulations by Theorem Proving for “general” kinetics
Differential Equations

- Linear ODE systems
  - Control theory
  - Electrical engineering
  - Kolmogorov equation for Continuous Time Markov Chains a.k.a. the Chemical Master Equation for discrete (-molecule count) chemistry

- Nonlinear ODE systems
  - Quantitative models of computing: (continuous) Petri Nets, (mean-field) PEPA, ...
  - Chemical Reaction Networks for continuous (-concentration) chemistry (with Mass Action or with Hill kinetics)
IDOL: Intermediate Drift-Oriented Language

\[ p ::= \varepsilon \mid \dot{x}_i = f, \ p \quad n, m \in \mathbb{Z} \text{ and } m \neq 0 \]

\[ f ::= n \mid x_i \mid f + f \mid f \cdot f \mid f \cdot \frac{1}{m} \]

- Each IDOL program is a list of variable drifts \( \dot{x}_i = f \)
- The semantics is:

\[ [x]^p_c : [0; T] \rightarrow \mathbb{R}^{y_p} \quad [x_i]^p_c(t) = \hat{\sigma}(x_i) + \int_0^t \left[ f_i \right]^p_c([x]^p_c(s)) \, ds \]

- where \( p \) is the full program, \( c = (T, \hat{\sigma}) \) is the time horizon and initial conditions, and \( x \) is the vector of all the \( x_i \).
- Provided there is a unique solution (there are sufficient conditions for that).
We <3 Tarski

- IDOL is within Tarski’s decidable fragment of reals
  - The Law of Mass action has drifts like \( x_1 \cdot x_2 \)
  - Hill kinetics has drifts like \( \frac{x_1^2}{1 + x_1^2} \)
  - PEPA uses drifts like
    \[
    \min(x_1, x_2) := \frac{1}{2}(x_1 + x_2 - |x_1 - x_2|), \quad \text{with } |x| := (x \cdot x)^{\frac{1}{2}}
    \]
    where \( y = x^{\frac{1}{2}} = \exists y(y^2 = x) \)
  - No trigonometry, no exponentials, etc. in our ODEs.

- Bisimulations over CRNs [CONCUR’15]
  - Are also formulas within Tarski’s fragment.
Differential Equivalence Relations

- We encode equivalences over IDOL programs
  - As first-order logic formulas containing IDOL terms.
- Z3 has a solver for them
  

- We use Z3 to minimize ODE (IDOL) systems
  - And, indirectly, to minimize Chemical Reaction Networks
  - On biological networks, Z3 is often faster than specialized polynomial algorithms!

- For Backward Bisimulation in particular:
  - We use a counter-example guided partition refinement algorithm.
- The IDOL solver uses Z3 as a subroutine
  - Possibly iteratively, e.g. for counter-example guided partitioning
### Benchmarks

| Model | $|R|$ | $|S|$ | Red. (s) | Size | Red. (s) | Size |
|-------|-----|-----|--------|------|--------|------|
| M1    | 34  | 70  | 6.54E-1| 745  | 7.85E+3| 105  |
| M2    | 34  | 70  | 2.81E-1| 354  | 3.22E+3| 105  |
| M3    | 1   | 11  | 1.29E-1| 411  | 6.46E+2| 47   |
| M4    | 8   |     | 2.46E-1| 348  | 5.22E+3| 215  |

**Table 1.** FDE reduces more than forward bisimulation (FB).

| Model | $|R|$ | $|S|$ | Red. (s) | Size | BB (s) | BDE (s) | $|S|$ |
|-------|-----|-----|---------|------|--------|---------|-----|
| M5    | 70  |     | 3.68E+3 | 167  | 1.01E+3| 167     |
| M6    | 70  |     | 1.77E+2 | 122  | 3.01E+2| 122     |
| M7    | 70  |     | 2.00E-3 | 12   | 6.00E-2| 12      |
| M8    | 73  |     | 1.32E+3 | 6634 | 3.45E+3| 6634    |
| M9    | 34  | 70  | 2.71E+2 | 5575 | 1.57E+3| 5575    |
| M10   | 22  | 23  | 6.00E-1 | 217  | 3.22E+0| 217     |
| M11   | 53  |     | 6.00E-3 | 56   | 2.71E-1| 56      |
| M12   | 18  |     | 7.00E-3 | 3    | 5.20E-2| 3       |

**Table 2.** BDE has runtimes similar to backward bisimulation (BB).
Automated model reduction for

- **Continuous Time Markov Chains**
  - By their forward Kolmogorov equation
- **Chemical Reaction Networks**
  - By their nonlinear ODE mass action kinetics
- **Stochastic Process Algebra**
  - Including PEPA, which has a min-based interaction law
- **Chemical Master Equation**
  - By the (linear) Kolmogorov equation
- **Linear Control Systems**
  - They are “just” linear ODEs
- **Electronic Circuits**
  - Kirchhoff’s laws...

Just compile them to IDOL

*Symbolic Computation of Differential Equivalences.*
Luca Cardelli, Mirco Tribastone, Max Tschaikowski, Andrea Vandin [POPL’16]
Further improvements

• General theorem proving is very appealing
  • We can leave some model components undefined or underconstrained, and let Z3 “figure them out”.

• Still, specialized algorithms can do better
  • By using a version of Tarjan’s Partition Refinement algorithm, we are getting amazing speedups in the computation of bisimulations for bimolecular CRNs.

Efficient Syntax-Driven Lumping of Differential Equations.
Luca Cardelli, Mirco Tribastone, Max Tschaikowski, Andrea Vandin [TACAS’16]
Morphisms of Antagonistic Networks
Bisimulations (partitions) of 1 network, vs. Morphisms (mappings) between 2 networks

- A morphism between two CRNs that preserves traces can be understood as a (backward) bisimulation over the species of a “union CRN”.
- Conversely, from a (many-to-one, backward) bisimulation we can reconstruct a canonical morphism between two networks.

- Such a bisimulation is called an emulation morphism: one network can exactly reproduce all the traces of the other network.
Antagonistic Networks

1 vs. 1
Mutual Inhibition & Self Activation

1 vs. 1
Mutual Inhibition & Mutual Anti-activation

2 vs. 2
low Notch → high Delta

3 vs. 3
high Delta → low Notch

Cell cycle transitions
Polarity establishment
Gene networks

Septation Initiation

Delta-Notch

The “new” cell cycle switch
Phosphorylation network dynamics in the control of cell cycle transitions
A Consensus Algorithm

- Approximate Majority (AM) Algorithm
  - Uses a third “undecided” population \( b \)
  - Disagreements cause agents to become undecided
  - Undecided agents agree with any non-undecided agent

\[
\begin{align*}
x + y & \rightarrow^r y + b \\
y + x & \rightarrow^r x + b \\
b + x & \rightarrow^r x + x \\
b + y & \rightarrow^r y + y
\end{align*}
\]

A Simple Population Protocol for Fast Robust Approximate Majority

Dana Angluin - James Aspnes - David Eisenstat

chemical reaction network

catalysis

activation inhibition

AM
A Biological Implementation

Approximate Majority (AM)

1) Bistable
   Even when initially $x=y$ (stochastically)

2) Fast (asymptotically optimal)
   $O(\log n)$ convergence time

3) Robust to perturbation
   above a threshold, initial majority wins \textit{whp}

Epigenetic Switch

Figure 1. Basic Ingredients of the Model

Dana Angluin · James Aspnes · David Eisenstat
A Simple Population Protocol for Fast Robust Approximate Majority

2007

2007
Network Emulation  MI emulates AM

• For *any rates and initial conditions* of AM, we can find *some* rates and initial conditions of MI such that the (6) trajectories of MI retrace those (3) of AM:

\[
\begin{align*}
\text{initialize:} \\
z &= x \\
\sim y &= x \\
(y_2 &= x_0 \\
y_1 &= x_1 \\
y_0 &= x_0)
\end{align*}
\]

• How do we find these matching parameters? By a network morphism!
Network Emulation: NCC emulates MI

- For any rates and initial conditions of MI we can find some rates and initial conditions of NCC such that the (18) trajectories of NCC retrace those (6) of MI.

\[ z, r, p \rightarrow z \]
\[ y, q, s \rightarrow y \]

(18 species on 6 trajectories)  
(6 species on 6 trajectories)
The (18) trajectories NCC can always retrace those (3) of AM.

The new cell cycle switch can emulate AM exactly. For any initial conditions of AM. And for any rates of AM.
Emulations are Modular
How to check for emulation

• How do we check a potential emulation morphism for all possible initial conditions of the target?
  • Statically! Check conditions on the joint stoichiometric matrices of the two networks under the mapping.

• How do we check a potential emulation morphism for all possible rates of the target?
  • Can’t; but if one emulation is found, then the rates of the target network can be changed arbitrarily and a related emulation will again exist.
Emulation Zoo

NCC

QI

DN

SI

AMs

CCr'

AM

SCr

SCr'

emulation (transitive)
Biological Corollaries

- By checking only static network and morphism properties we can learn that:
  - All these networks are (at least) bistable
  - (We do not have to reanalyze the steady states of all these dynamical systems)
  - All these networks can perform exactly as fast as AM
  - (We do not have to reprove the complexity bounds for all these networks)
Network Emulation Morphism FAQ

- **What guarantees emulation?**
  - Reactant morphism + stoichiomorphism: static, state-independent (structural) conditions

- **How do you find them?**
  - Emulation Theorem => they do not depend on initial conditions
  - Change of Rates Theorem => can look for rate-1 morphisms
  - E.g. test all possible rate-1 homomorphism between two networks to see if they are stoichiomorphisms

- **How common are they?**
  - Likely relatively rare, but still many useful ones => richness of networks space

- **How useful are they?**
  - Establish structural, algorithmic, (non-accidental) reasons for kinetic similarity
  - Explain simple behavior “facets” of complicated networks
  - Investigate evolutionary paths (maybe)

- **How brittle are they?**
  - Will a perturbed trajectory of the source network converge to a trajectory of the target network?
  - What about other reaction kinetics?

- **What about stochastic?**
  - Is there a CME Emulation Theorem?
Network Morphisms as Evolutionary Paths
Network Evolution

Across species: *Ortholog genes*

Within species: *Paralog genes*

“same function”

“new function”
Walks in Network Space

( Homomorphism and stoichiomorphism (transitive))
Walks in Network Space
Walks in Network Space

Neutral paths in network space

Side jumps
Noise Reduction in Biochemical Switches
Basic Switches (deterministic)

(A) Influence network diagrams
(B) Chemical reaction network diagrams and feedback loops
(C) Numerical solutions of the deterministic kinetics of the networks:
   - Horizontal axis is time
   - Vertical axis is species concentration

First some arbitrary initial conditions are chosen for AM. Then the initial conditions of the other networks are chosen in such a way that each trace of each of the other networks retraces exactly one trace of AM. This can be done for any initial conditions chosen for AM, and indicates the potential of each of the other networks to operate as a simpler switch.

Noise Reduction in Complex Biological Switches

Luca Cardelli\textsuperscript{1,5,7}, Attila Csikász-Nagy\textsuperscript{3,4,5}, Neil Dalchau\textsuperscript{1,5}, Mirco Tribastone\textsuperscript{5,7}, Max Tschikowski\textsuperscript{5,7}

(To appear.)
Basic Switches (stochastic)

Horizontal axes is time
Vertical axes is number of molecules.

(A) Influence networks.
(B) Chemical Master Equation solution: probability distribution, with color (in 10 bands from light = 0 to dark = 1) indicating the probability that at time t there are y molecules of the single indicated species.
(C) Chemical Master Equation solution: mean (solid lines) and standard deviation (color bands) for the species in the network.
(D) Central Limit Approximation solution: mean (solid lines) and standard deviation (color bands) for the species in the network.

Disentangle the contribution of complexity to stochasticity

Compare network noise on the baseline of deterministic emulation, across networks of different size and structure.
More Complex Switches

Horizontal axes are time, vertical axes are number of molecules.

(A) Influence networks.
(B) ODE solutions for comparison
(C) Chemical Master Equation solution: mean (solid lines) and standard deviation (color bands) for the species in the network.
(D) Central Limit Approximation solution: mean (black lines) and standard deviation (color bands) for the species in the network.
Intrinsic Noise

Complexity improves overall performance of the cell cycle switch. The performance of different networks was evaluated by calculating the standard deviation of the main molecular states over time. Standard deviations are calculated via numerical integration of the chemical master equation (CME) using the Visual GEC software, and via numerical integration of the central limit approximation (CLA) in Matlab.
Extrinsic Noise

Extrinsic noise is introduced by randomly perturbing all the reaction rates (separately but from the same distribution) of each model. (So the total variation in more complex models is actually higher.) Variations in network behaviour is assessed in comparison to the default parameters, in which all reaction rates are set equal to 1. Network variation is quantified using the summed Wasserstein metric over the whole probability distribution over time.

MI and SI have the same number of species and reactions.

Complexity can confer robustness to extrinsic noise.

MI and SI have the same number of species and reactions.
Noise vs. Complexity

• With corresponding initial conditions, all studied networks show the same mean behavior.
• CCr emulating AM is the simplest explanation of the core cell cycle switching function.
• Many other biological switches can be so reduced to an algorithm with well-understood properties.
• On the basis of kinetic similarity of mean behavior, we show variations in noise behavior (both intrinsic and extrinsic).
• Noise tends to decrease with complexity, but this also depends on network structure and not directly on total molecular counts.
Conclusions
Computational Methods

• Comparing Networks
  • Explanation of network structure (how functionality is achieved)

• Network Bisimulations (and Morphisms)
  • Feasible for large networks by partition refinement algorithms

• Finding Bisimulations by Theorem Proving
  • Also feasible for large networks by “magical” theorem proving
  • Supports kinetics other than mass action
Systems Biology

- Morphisms of Antagonistic Networks
  - Entail deep properties of complex networks (bistability, optimality)

- Network Morphisms as Evolutionary Paths
  - Neutral paths in network space

- Noise Reduction in Complex Biochemical Switches
  - Deterministic morphisms as a baseline for making stochastic comparisons between networks of different sizes