Microsoft Research

The Cell Cycle Switch Computes Approximate Majority

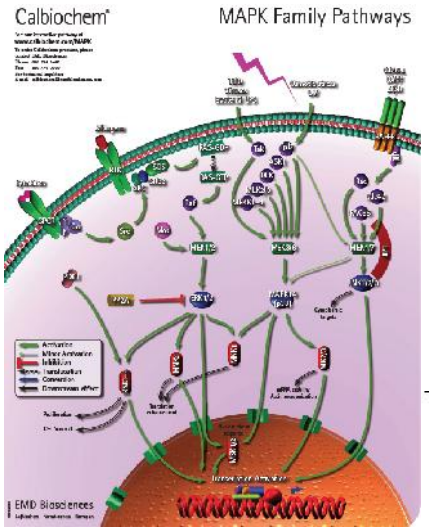
Luca Cardelli, Microsoft Research

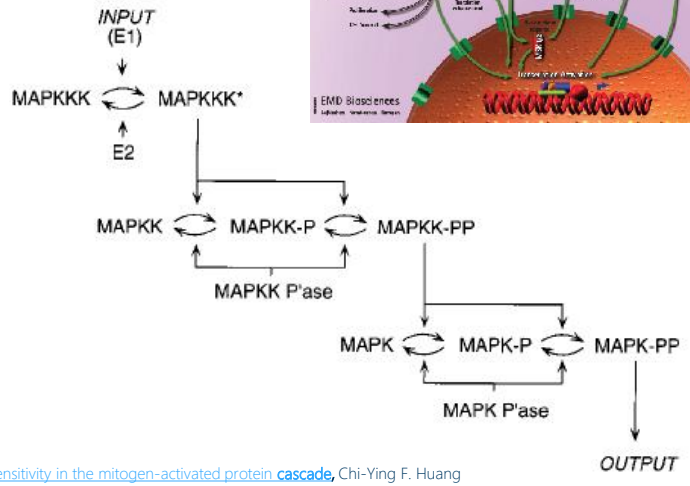
Joint work with Attila Csikász-Nagy, CoSBi & King's College London

Natural Algorithms Workshop, Princeton, 2013-05-21

Cells Compute

- No survival without computation!
  - Finding food
  - Avoiding predators
- How do they compute?
  - Clearly doing "information processing"
  - Based on complex, higher-order interactions
    - MAPKKK = MAP Kinase Kinase Kinase = *that which operates on that which operates on that which operates on protein*.
  - How 'sophisticated' are natural algorithms?





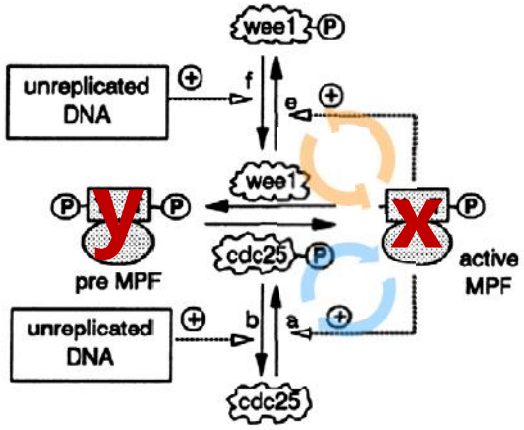
Ultrasensitivity in the mitogen-activated protein cascade, Chi-Ying F. Huang and James E. Ferrell, Jr., 1996, *Proc. Natl. Acad. Sci. USA*, 93, 10078-10083.

Outline

- Analyzing biomolecular networks
  - Try do understand the function of a network
  - But also try to understand its *structure*, and what determines it
- The Cell-Cycle Switches
  - Some of the best studied molecular networks
  - Important because of their fundamental function (cell division) and the stability of the network across evolution
- We ask:
  - What does the cell cycles switch compute?
  - How does it compute it?

The Cell Cycle Switch

- This network is universal in all Eukaryotes [P. Nurse]
  - I.e., the *network* at the core of cell division is *the same* from yeast to us
  - *Not the components* of the network, *nor the rates*



Journal of Cell Science 106, 1153-1168 (1993)
Printed in Great Britain © The Company of Biologists Limited 1993

Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos

Bela Novak* and John J. Tyson†
Department of Biology, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24060-0406, USA
*Permanent address: Department of Agricultural Chemical Technology, Technical University of Budapest, 1521 Budapest Gellert Ter 4, Hungary
†Author for correspondence

Double positive feedback on x
Double negative feedback on x
No feedback on y
What on earth … ???

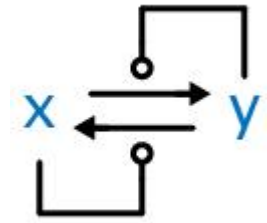
- The function is very well-studied. But why this structure?
- I.e., why this algorithm?

How to Build a Good Switch

- What is a "good" switch?
  - We need first a bistable system: one that has two *distinct* and *stable* states. I.e., given any initial state the system must settle into one of two states
  - The settling must be fast (not get stuck in the middle for too long) and robust (must not spontaneously switch back)
  - Finally, we need to be able to flip the switch by external inputs
- "Population" Switches
  - Populations of identical agents (molecules) with the whole population switching from one state to another as a whole
  - Highly concurrent (stochastic)

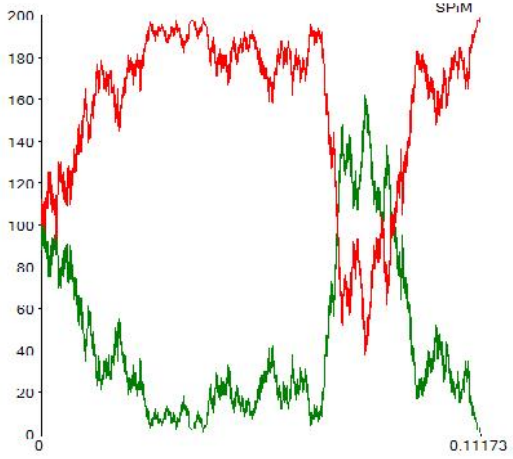
A Bad Algorithm

- Direct Competition
  - x catalyzes the transformation of y into x
  - y catalyzes the transformation of x into y
  - when all-x or all-y, it stops
- This system has two end states, but
  - Convergence to an end state is slow (a random walk)
  - Any perturbation of an end state can start a random walk to the other end state (hence not really *bistable*)



$y + x \rightarrow x + x$

$x + y \rightarrow y + y$



A Very Good Algorithm

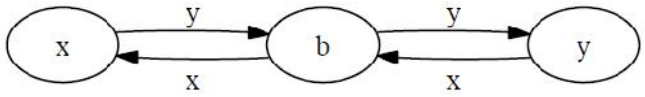
- Approximate Majority (AM)
  - Decide which of two populations is in majority
- A fundamental 'population protocol'
  - Agents in a population start in state x or state y
  - A pair of agents is chosen randomly at each step, they interact ('collide') and change state
  - The whole population must eventually agree on a majority value (all-x or all-y) with probability 1

Dana Angluin · James Aspnes · David Eisenstat

A Simple Population Protocol for Fast Robust Approximate Majority

We analyze the behavior of the following population protocol with states $Q = \{b, x, y\}$. The state b is the **blank** state. Row labels give the initiator's state and column labels the responder's state.

	x	b	y
x	(x,x)	(x,x)	(x,b)
b	(b,x)	(b,b)	(b,y)
y	(y,b)	(y,y)	(y,y)



Third 'undecided' state

1) Disagreements cause agents to become undecided
2) Undecided agents believe any non-undecided agent they meet

Properties

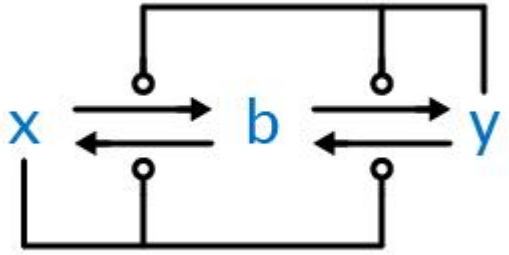
[Angluin et al., http://www.cs.yale.edu/homes/aspnes/papers/disc2007-eisenstat-slides.pdf]

- With high probability, for *n* agents
  - The total number of interactions before converging is $O(n \log n)$
    $\Rightarrow$ fast
  - The final outcome is correct if the initial disparity is $\omega(\sqrt{n} \log n)$
    $\Rightarrow$ solution states are robust to perturbations
- Logarithmic time bound in parallel time
  - *Parallel time* is the number of steps divided by the number of agents
  - In parallel time the algorithm converges with high probability in $O(\log n)$

Chemical Implementation

Chemistry as a programming language for population algorithms!

$x + y \rightarrow y + b$
$y + x \rightarrow x + b$
$b + x \rightarrow x + x$
$b + y \rightarrow y + y$

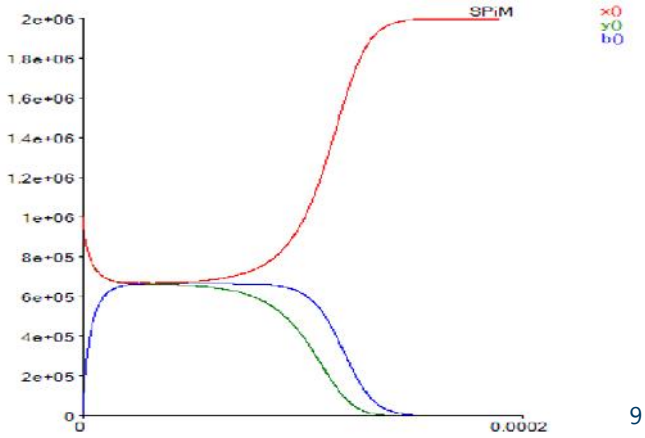


Bistable
Even when x=y! (stochastically)

Fast
O(*log n*) convergence time

Robust to perturbation
above a threshold, initial majority wins *whp*

Worse-case scenario example, starting with x=y, b=0:

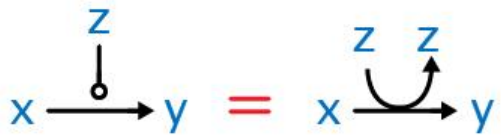


Back to the Cell Cycle

- The AM algorithm has ideal properties for settling a population into one of two states
- But that is not what the cell cycle uses
- Or is it?

Influence Network Notation

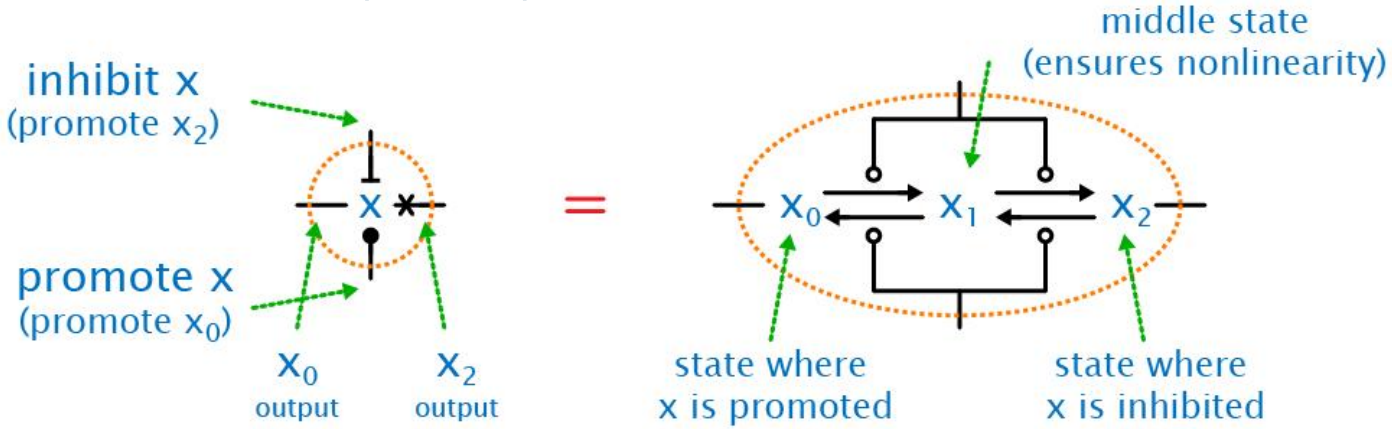
- Catalytic reaction



z is the catalyst

$x + z \rightarrow z + y$

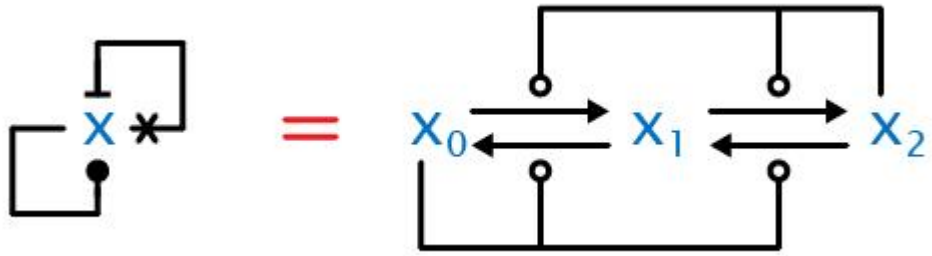
- 'Double kinase-phosphatase' motif



influence node

catalytic node

Step 1: the AM Network



(x_2 promotes x_2)

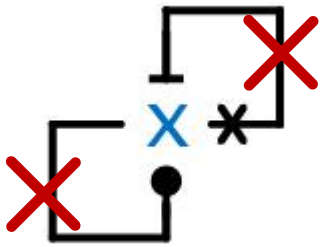
(x_0 promotes x_0)

- … not biochemically plausible

Natural Constraint #1

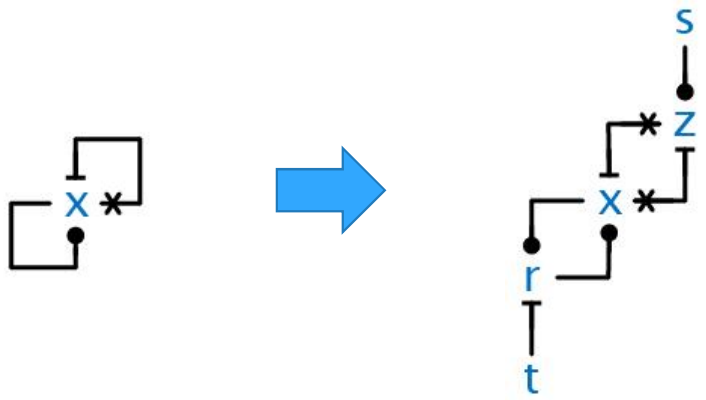
- Direct autocatalysis is not commonly seen in nature

$$x_1 + x_0 \rightarrow x_0 + x_0$$
$$x_1 + x_2 \rightarrow x_2 + x_2$$



Step 2: remove auto-catalysis

- Replace autocatalysis
  - By *mutual* (simple) catalysis, introducing intermediate species z and r
  - z and r need to 'relax back' when they are not being promoted: s and t provide the back pressure for such relaxation



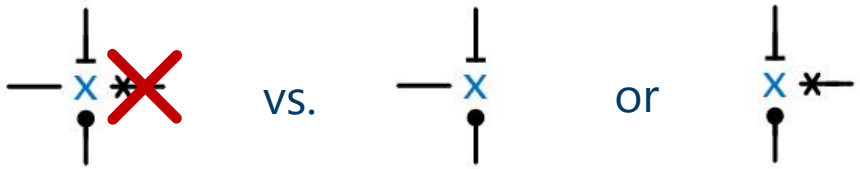
(x_2 promotes z_2, promotes x_2)

(x_0 promotes r_0, promotes x_0)

- … still not biochemically plausible.

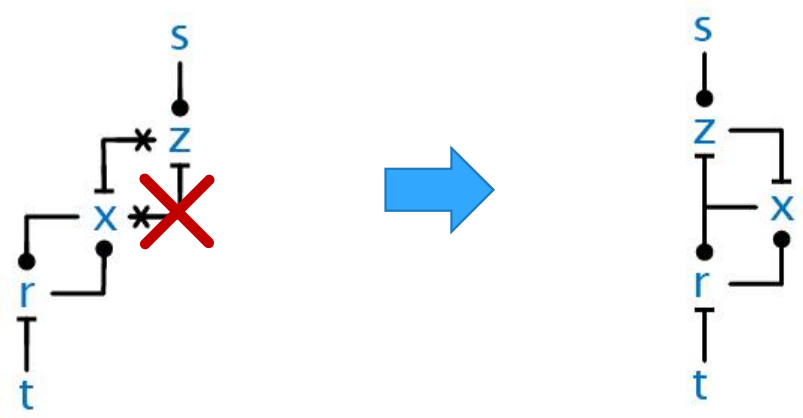
Natural Constraint #2

- x_0 and x_2 (usually two states of the same molecule) are both active catalysts in that network
- That is not commonly seen in nature



Step 3: only one active state per species

- Remove the catalytic activity of x_2
  - By "flipping the z feedback to the other side"



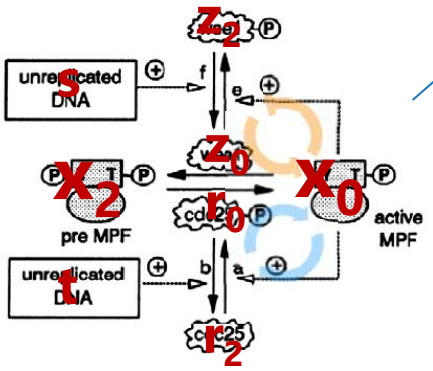
(x_2 promotes z_0 via s bias,
z_0 promotes x_2 via inhibiting x_0)

(x_0 promotes r_0, promotes x_0)

- All species now have one active (x_0,z_0,r_0) and one inactive (x_2,z_2,r_2) form
- This is 'biochmically plausible'

Network Structure

- … and that is the cell-cycle switch!



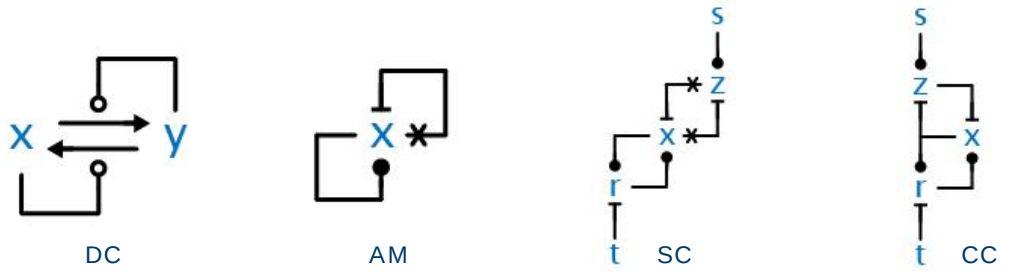
Nobel-prize winning network

Variation on a distributed algorithm

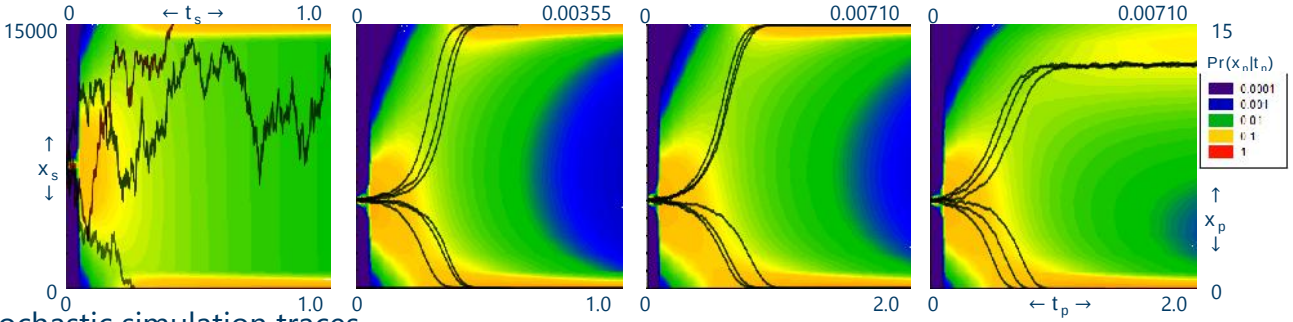
- But did we preserve the AM function through our network transformations?
- Ideally: prove either that the networks are ‘contextually equivalent’ or that the transformations are ‘correct’
- Practically: compare their ‘typical’ behavior

Convergence Analysis

- Switches as computational systems



Start symmetrical
(x_0=x_1=x_2 etc.)

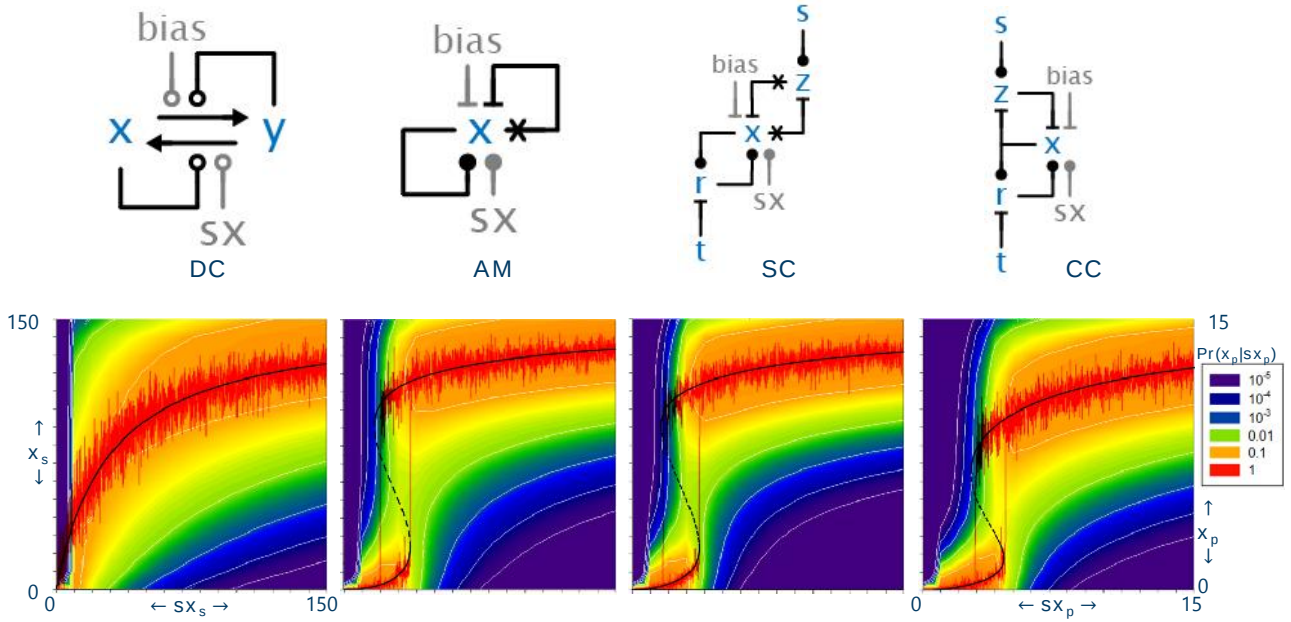


Black lines: several stochastic simulation traces
Color: full probability distribution of small-size system

NEW!
CC appears to converge in log time

Steady State Analysis

- Switches as dynamical systems

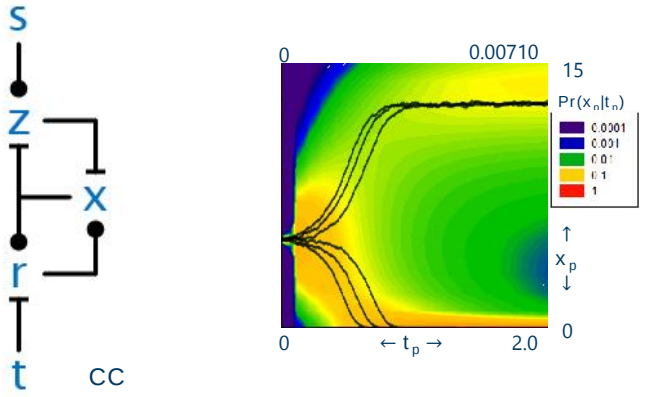


Black lines: deterministic ODE bifurcation diagrams
Red lines: noisy stochastic simulations
Color: full probability distribution of small-size system

NEW!
AM shows hysteresis

Evidence that CC is 'similar' to AM

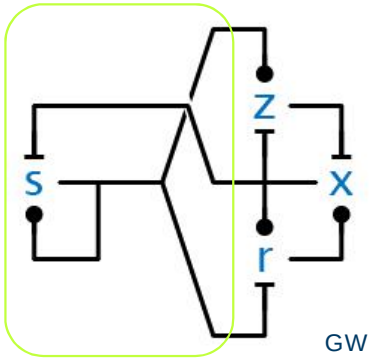
- But there was a difference
  - The output of CC does not go 'fully on' like AM:

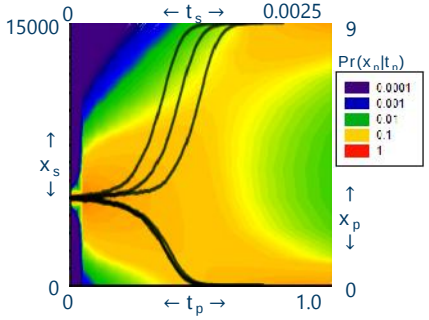


  - Because s continuously inhibits x through z, so that x cannot fully express
  - Q: Why didn't nature do better than that?

Nature fixed it!

- There is another known feedback loop
  - By which x suppresses s “in retaliation” via the so-called Greatwall loop
  - Also, s and t happen to be the same molecule





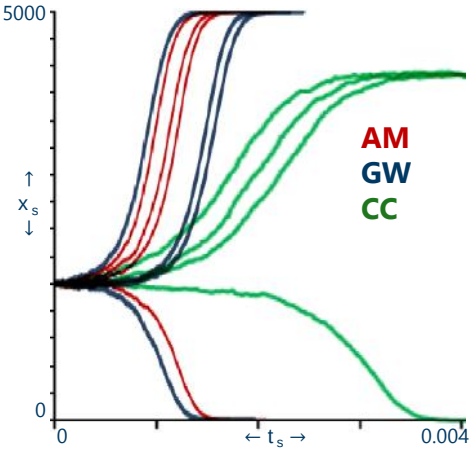
Full activation!

  - (As usual, there are many more details in real biological networks; this is one of the many details people knew about without fully understanding its function)

More surprisingly

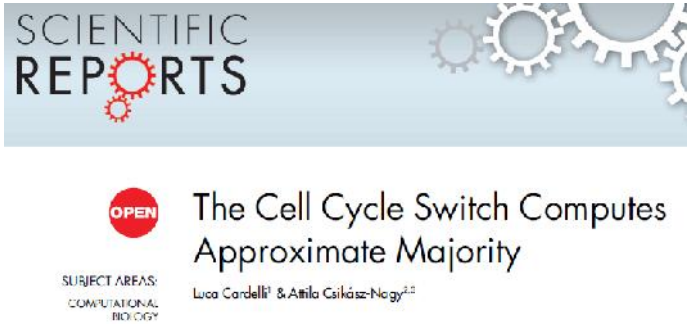
- Made it faster too!
  - The extra feedback also speeds up the decision time of the switch, making it about as good as the 'optimal' AM switch:

Conclusion (in our published paper):
Nature is trying as hard as it can to implement an AM-class algorithm!



The Greatwall Kinase

- Our paper appeared:
  - Suggesting GW is a better switch than CC, also in the context of oscillators



- Another paper the same week:
  - Showing experimentally that the Greatwall loop is a necessary component of the switch, i.e. the not-as-good-as-AM network has been 'refuted'

But what about network equivalence?

- Our evidence is empirical
  - Although quantitative and covering both kinetic and steady state behavior
  - Also, contextual equivalence holds in the context of oscillators (see paper)
- Analytical evidence is harder to obtain
  - The proof techniques for the AM algorithm are hard and do not generalize easily to more complex networks
  - Quantitative theories of behavioral equivalence and behavioral approximation, e.g. in process algebra, are still lacking (although rich qualitative theories exist)

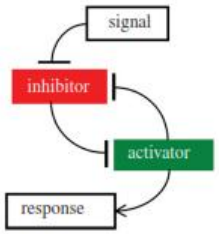
Mutual Inhibition

- A new paper suggests that all cellular switches in all phases of the cell cycle follow (abstractly) a mutual inhibition pattern:

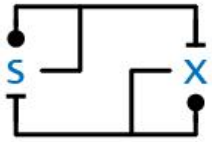
Molecular mechanisms creating bistable switches at cell cycle transitions

Anael Verdugo, P. K. Vinod, John J. Tyson and Bela Novak

Open Biol. 2013 3, 120179, published 13 March 2013

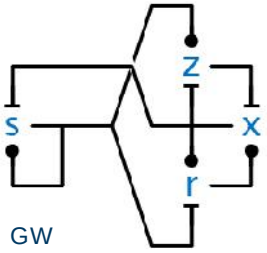


In our notation:



MI

cf.:



GW

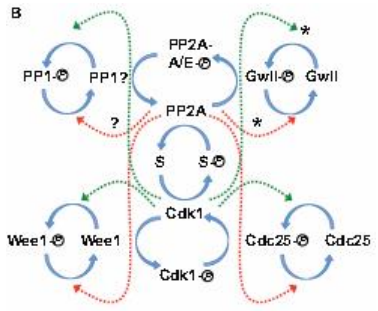
New Cell Cycle Network

- A new paper presents a more complete view of the cell cycle switch
- N.B. "phosphorylation network dynamics" is the same as our x_0-x_1-x_2 motif

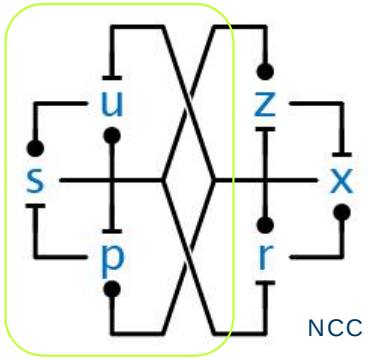
Phosphorylation network dynamics in the control of cell cycle transitions

Daniel Fisher[1,*], Liliana Krasinska[1,‡], Damien Coudreuse[2,‡] and Béla Novák[3,‡]
[1]Institut de Génétique Moléculaire de Montpellier, IGMM, CNRS UMR 5535, Université Montpellier I and II, 34293 Montpellier, France
[2]Institute of Genetics and Development of Rennes, CNRS UMR 6290, 35043 Rennes, France
[3]Oxford Centre for Integrative Systems Biology, Department of Biochemistry, University of Oxford, South Parks Road, Oxford OX1 3QU, UK
*Author for correspondence (daniel.fisher@igmm.cnrs.fr)
‡These authors contributed equally to this work

Journal of Cell Science 126, 4753–4761
© 2013. Published by The Company of Biologists Ltd
doi: 10.1242/jcs.138501

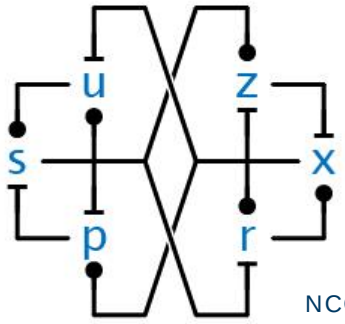


In our notation:



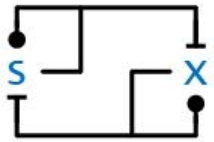
Network Emulation

- For chosen (uniform) initial conditions, the ODEs (and hence trajectories) of **NCC** collapse to those of **MI** (thanks to David Soloveichik):

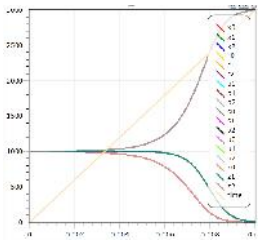


NCC

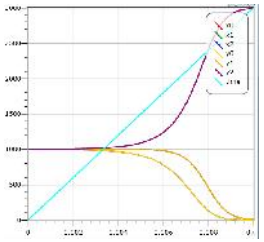
x,r,p ⇢ x
s,u,z ⇢ s



MI



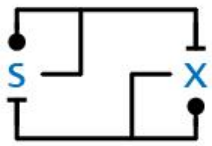
(18 species on 3 trajectories)



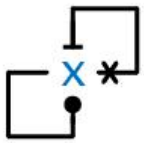
(6 species on 3 trajectories)

Network Emulation

- For chosen (uniform) initial conditions, the ODEs (and hence trajectories) of **MI** collapse to those of **AM**:

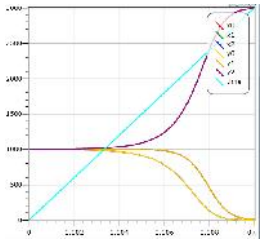


S,X⇢ X

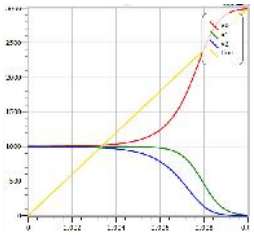


MI

AM



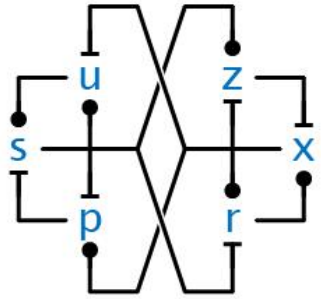
(6 species on 3 trajectories)



(3 species on 3 trajectories)

Conclusions

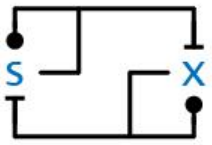
- The cell cycle switch *can* exactly emulate AM



NCC

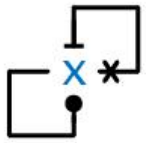
(New) cell cycle switch

emulates:



MI

emulates:



AM

Approximate majority algorithm

- Nature likes a good algorithm!


Microsoft

In separate work…

- We have a chemical implementation of AM using DNA gates
- I.e., a 'synthetic reimplementation' of the central cell-cycle switch.

Programmable chemical controllers made from DNA

Yuan-Jyue Chen[1], Neil Dalchau[2], Niranjan Srinivas[3], Andrew Phillips[2], Luca Cardelli[2], David Soloveichik[4], and Georg Seelig[1,5]

[1] Department of Electrical Engineering, University of Washington, Seattle
[2] Microsoft Research, Cambridge (UK)
[3] Computation and Neural Systems, California Institute of Technology, Pasadena
[4] Center for Systems and Synthetic Biology, University of California, San Francisco
[5] Department of Computer Science & Engineering, University of Washington, Seattle