The Cell Cycle Switch Computes Approximate Majority

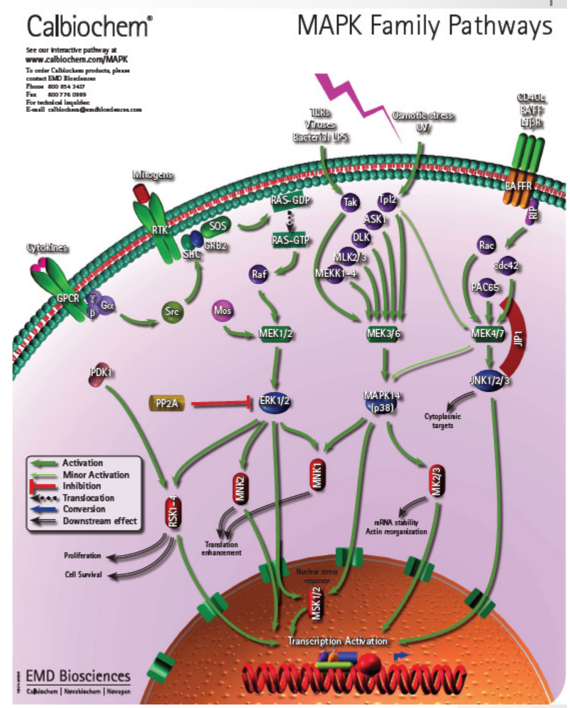
Luca Cardelli
Microsoft Research

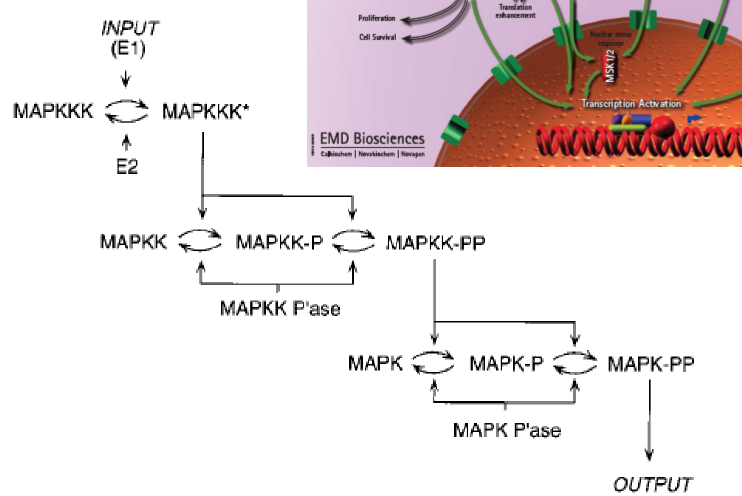
Joint work with Attila Csikász-Nagy
CoSBi & King's College London

Edinburgh, 2012-11-06
http://lucacardelli.name

Cells Compute

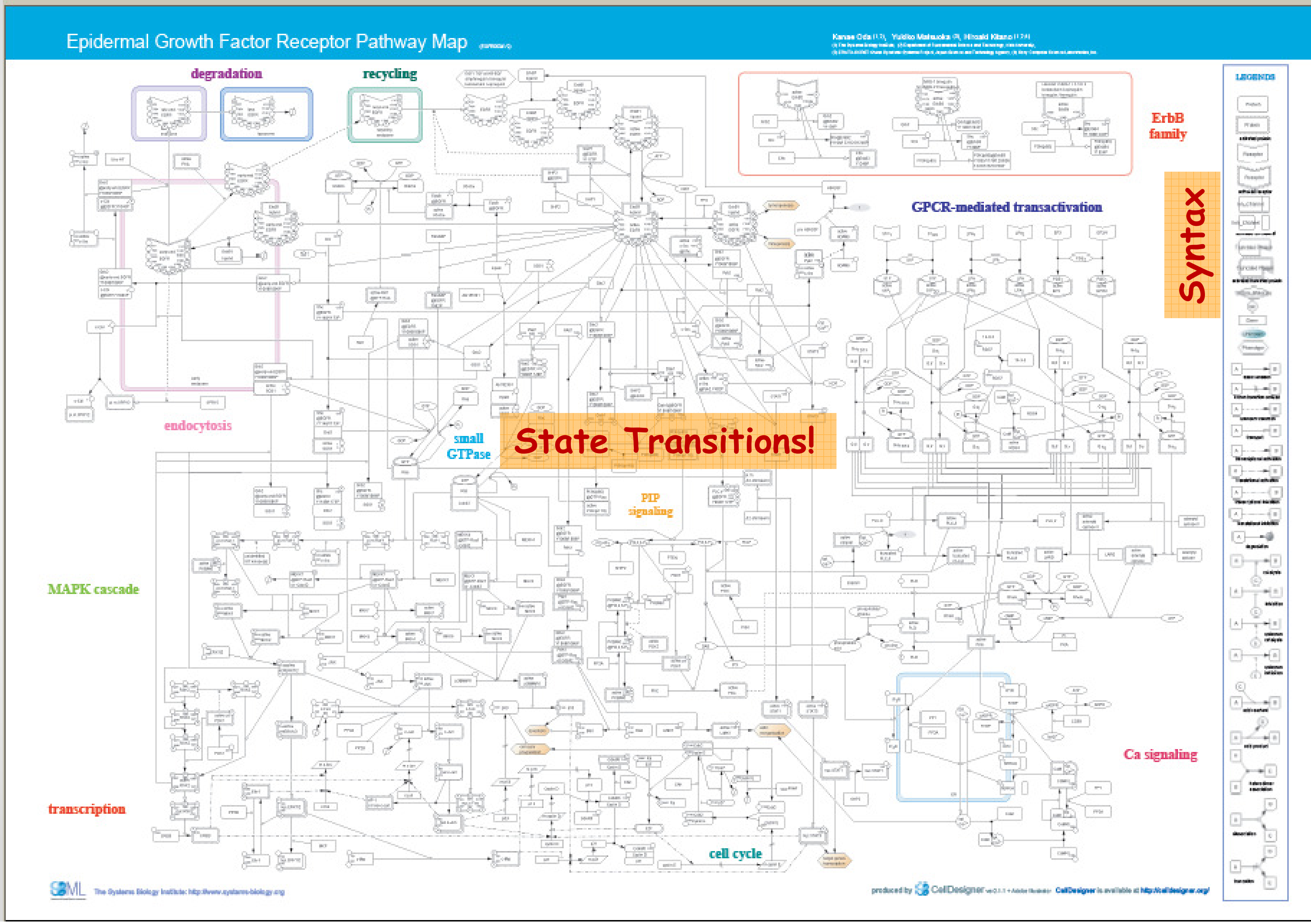
- No survival without computation!
  - Finding food
  - Avoiding predators
- How do they compute?
  - Unusual computational paradigms.
  - Proteins: do they work like electronic circuits?
  - Genes: what kind of software is that?
- Signaling networks
  - Clearly "information processing"
  - They are "just chemistry": molecule interactions
  - But what are their principles and algorithms?
- Complex, higher-order interactions
  - MAPKKK = MAP Kinase Kinase Kinase:
    that which operates on that which operates on
    that which operates on protein.





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.

Molecular Interaction Maps (Kohn/Kitano)



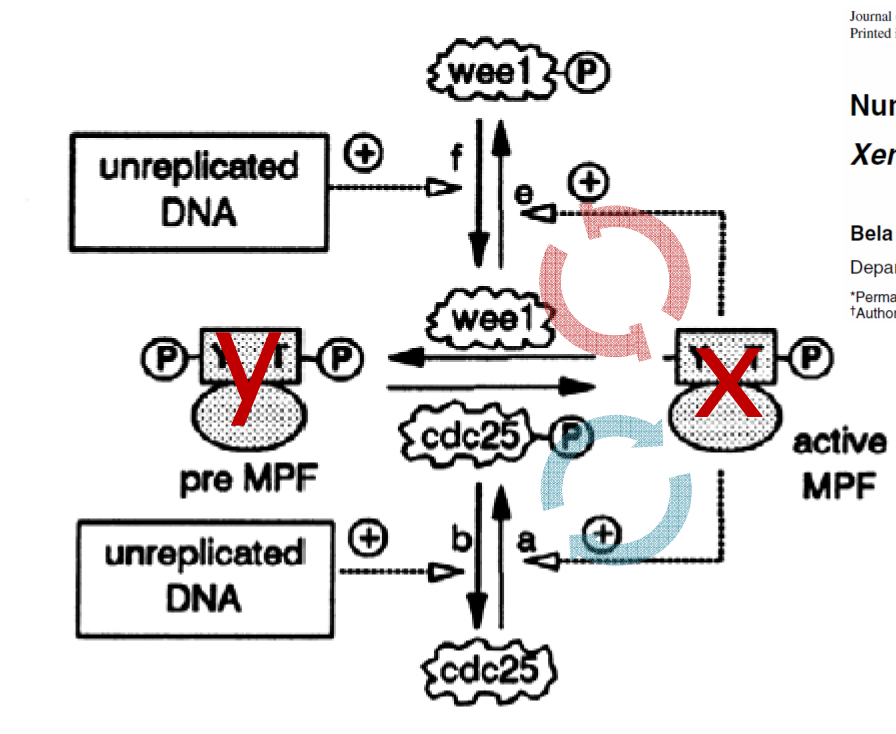
Outline

- Analyzing biomolecular networks
  - Various biochemical/bioinformatic techniques can tell us something about network structures.
  - We try do discover the function of the network, or to verify hypotheses about its function.
  - We try to understand how the structure is dictated by the function and other natural constraints.

- The Cell-Cycle Switches and Oscillators
  - Some of the best studied molecular networks.
  - Important because of their fundamental function (cell division) and preservation across evolution.

The Cell Cycle Switch

- At the core of the cell-cycled oscillator.
  - This network is universal in all Eukaryotes [P. Nurse].



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 … ???

  - Well studied. But *why this structure?*

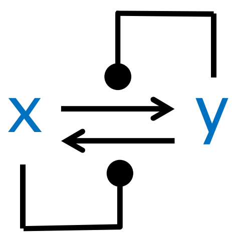
How to Build a Switch

- 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: drive the transitions by external inputs.

- "Population" Switches
  - Populations of identical agents (molecules) that switch from one state to another *as a whole*.
  - Highly concurrent (stochastic).

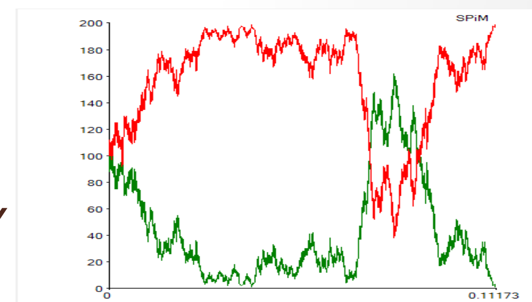
A Bad Algorithm

- Direct x–y competition
  - x catalyzes the transformation of y into x
  - y catalyzes the transformation of x into y



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

- This system is ~~bistable~~, but
  - Convergence to a stable state is *slow* (a random walk).
  - *Any* perturbation of a stable state can initiate a random walk to the other stable state.

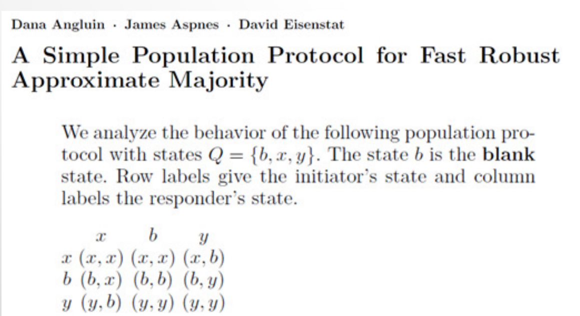



```
directive sample 0.0002
1000
directive plot x(); y(); b()

val rt = 10.0
new xcat@rt:chan
new ycat@rt:chan

let x() =
  do !xcat; x()
  or ?ycat; y()

and y() =
  do !ycat; y()
  or ?xcat; x()

run 100 of x()
run 100 of y()
```

A Very Good Algorithm

- Approximate Majority
  - Decide which of two populations is in majority

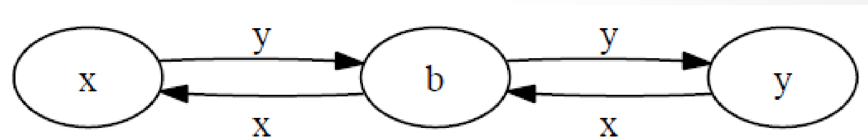
- A fundamental ‘population protocol’
  - Agents in a population start in state x or state y.
  - A pair of agents is chosen randomly at each step, they interact ("collide") and change state.
  - The whole population must eventually agree on a majority value (all x or all y) with probability 1.


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.

Properties

- With high probability, for n agents

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

  - The number of state changes before converging is O(n log n)
  - The total number of <u>interactions</u> before converging is O(n log n)
  - The final outcome is correct if the initial disparity is ω(sqrt(n) log n)

- The algorithm is the fastest possible
  - Must wait Ω(n log n) steps in expectation for all agents to interact

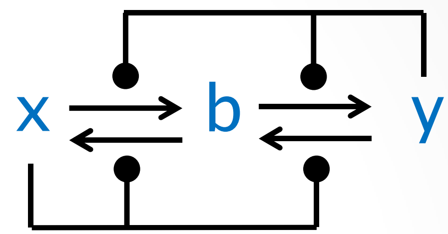
- Logarithmic time bound
  - Parallel time is the number of steps divided by the number of agents.
  - In parallel time the algorithm converges with high probability in O(log n).
  - That is true for any initial conditions, even x=y!

"Although we have described the population protocol model in a sequential light, in which each step is a single pairwise interaction, interactions between pairs involving different agents are independent and may be thought of as occurring in parallel. In measuring the speed of population protocols, then, we define 1 unit of parallel time to be jV j steps. The rationale is that in expectation, each agent initiates 1 interaction per parallel time unit; this corresponds to the chemists' idealized assumption of a well-mixed solution."
Distributed Computing 21(2):87-102.

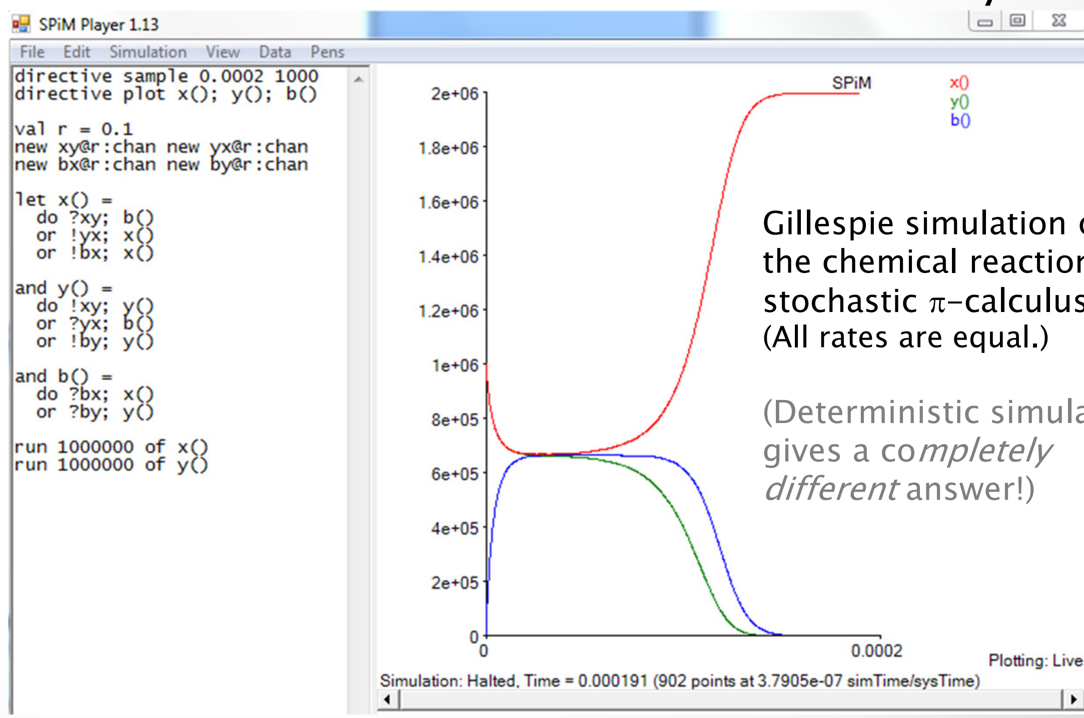
Chemical Implementation

A programming language for population algorithms!

$$x + y \to y + b$$
$$y + x \to x + b$$
$$b + x \to x + x$$
$$b + y \to y + y$$



Worse case test: start with x=y.



Gillespie simulation of the chemical reactions in stochastic π-calculus. (All rates are equal.)

(Deterministic simulation gives a *completely different* answer!)

Bistable
- Even when x=y! (stochastically)

Fast
- O(log n) convergence time

Robust
- ω(√n log n) majority wins whp

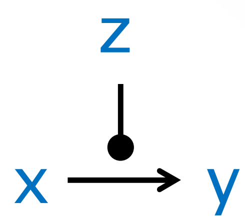
Back to the Cell Cycle

- The AM algorithm has great properties for settling a population into one of two states.
- But that is not what the cell cycle uses to switch its populations of molecules.
- Or is it?

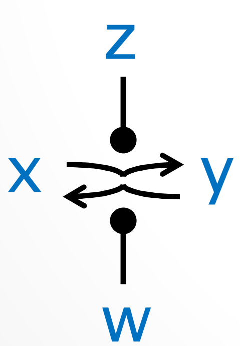
Some Notation

- Catalytic reaction

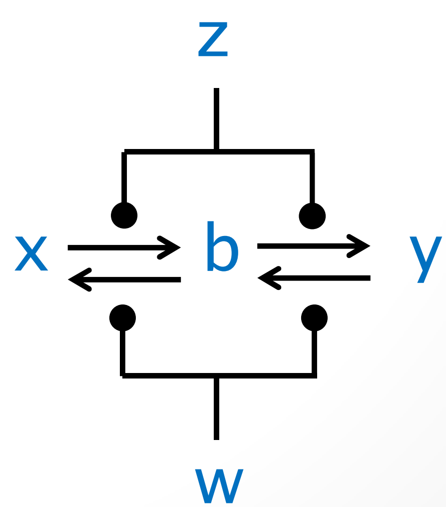
$x + z \rightarrow z + y$ =



- Double 'kinase-phosphatase' reactions

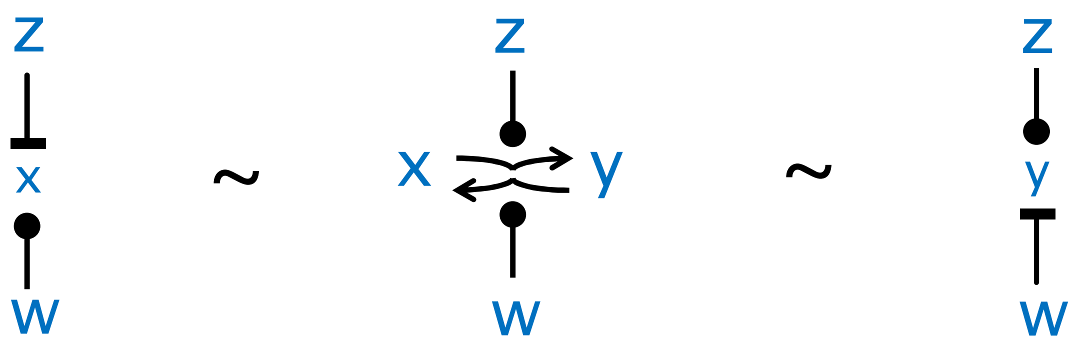


=



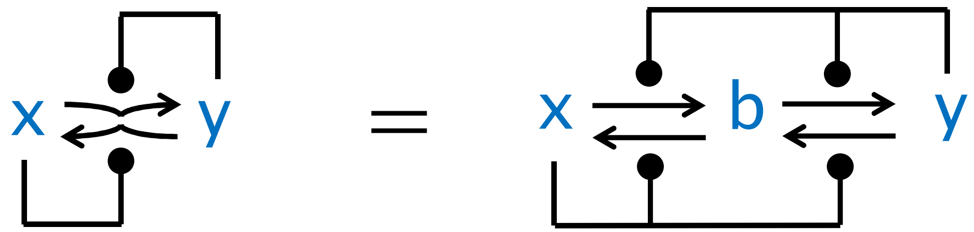
Stimulation/Inhibition

- A possible (mass-action) non-linear mechanism for stimulation/inhibition influence.



Step 1: the AM Network

Abbreviated notation:

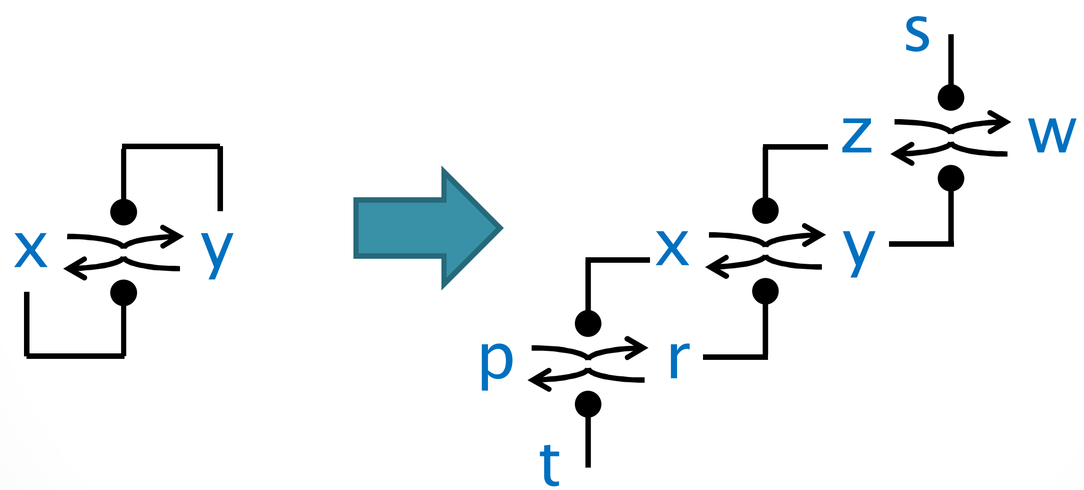


- Not biochmically plausible.
- CONSTRAINT: Autocatalysis, and especially intricate autocatalysis, is not commonly seen in nature.

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

Step 2: remove auto-catalysis

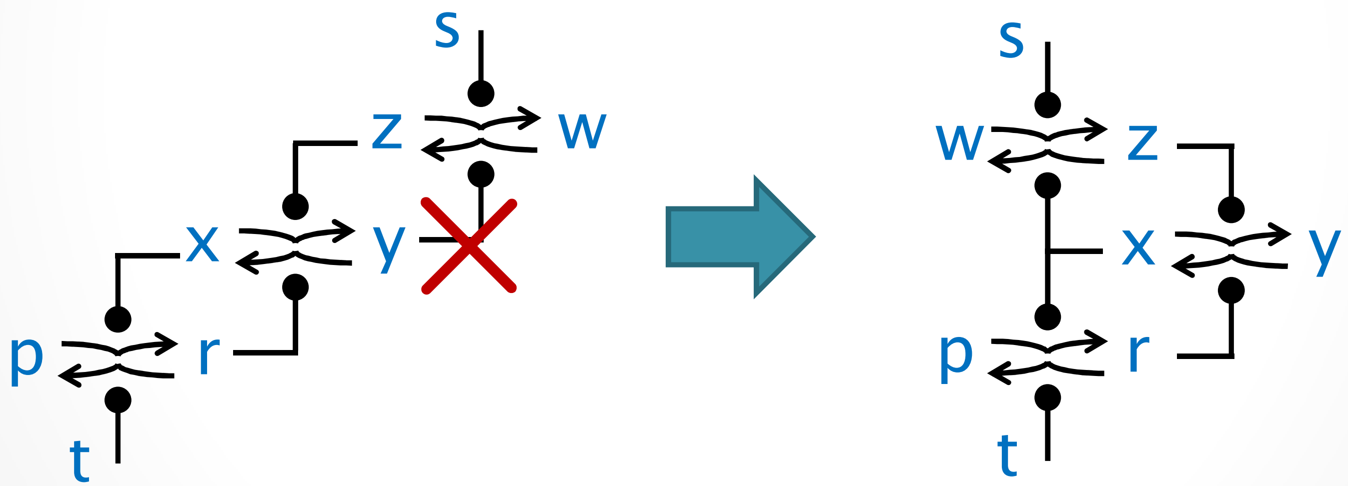
- Replace autocatalysis by mutual (simple) catalysis, introducing intermediate species z, r.
  - Here z breaks the y auto-catalysis, and r breaks the x auto-catalysis, while preserving the feedbacks.
  - z and r need to 'relax back' (to w and p) when they are not catalyzed: s and t provide the back pressure.



- Still not biochmically plausible.
- CONSTRAINT: x and y (two states of the same molecule) are distinct active catalysts: that is not common in nature.

Step 3: only one active state

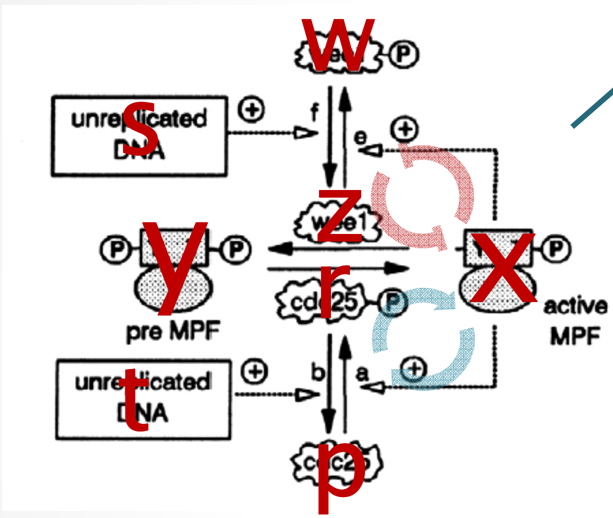
- Remove the catalytic activity of y.
  - Instead of y activating itself through z, we are left with z activating y, which remains passive.
  - We still need z to (sometimes) activate y.
  - Hence, to fully deactivate y we now need to deactivate z.
  - Since x 'wants' to deactivate y, we make x deactivate z.



- All species now have one active (x,z,r) and one inactive (y,w,p) form. This is 'biochmically plausible'.

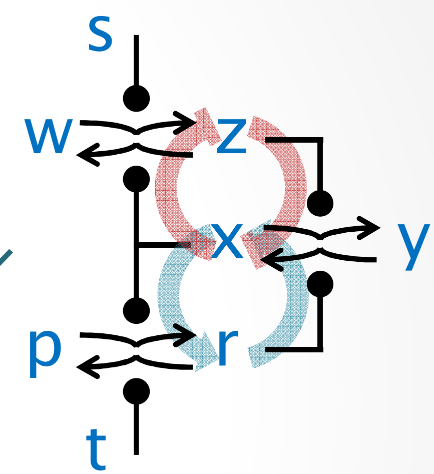
Network Structure

- … and that *is* the cell-cycle switch!



Nobel-prize winning network

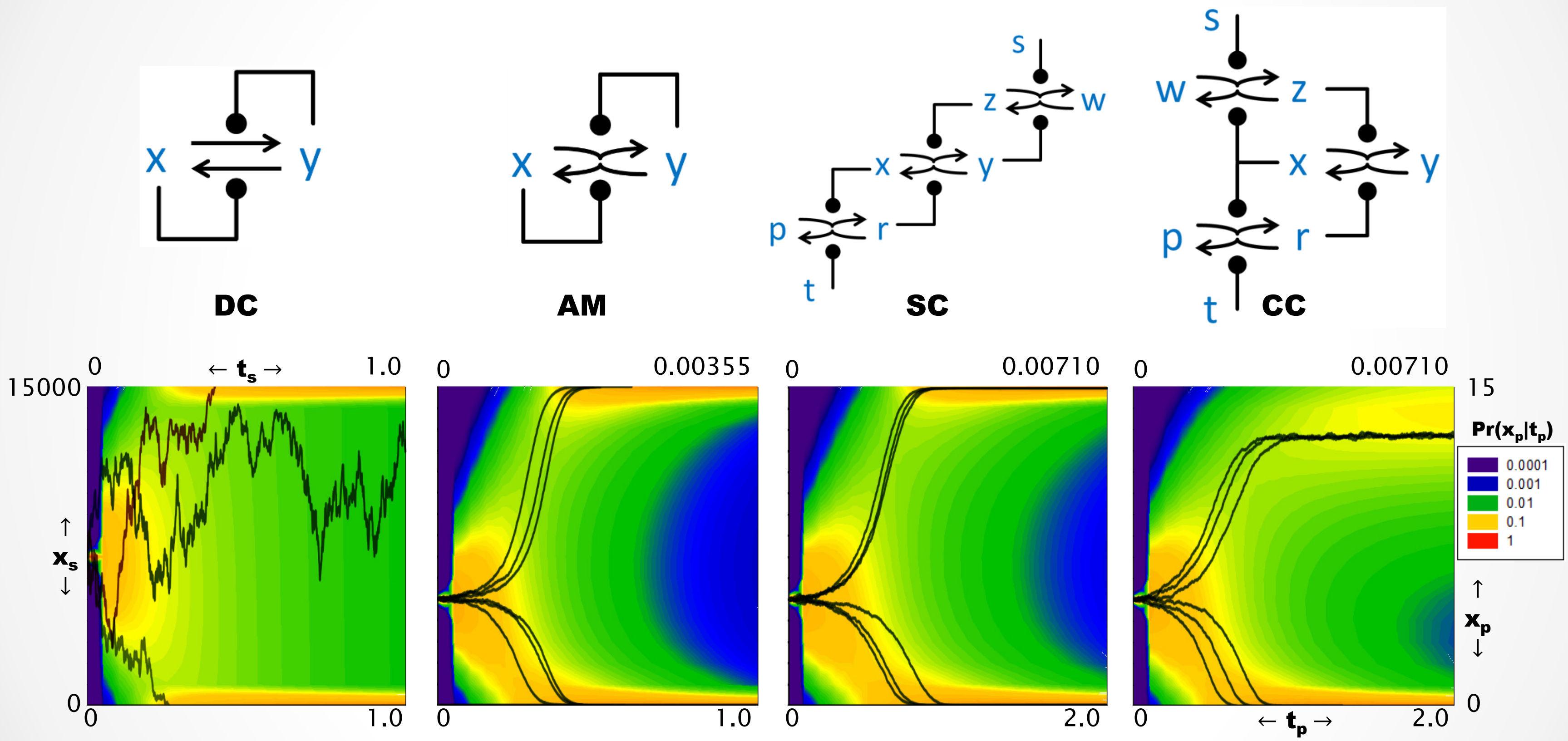
Variation on a distributed algorithm



- The question is: 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

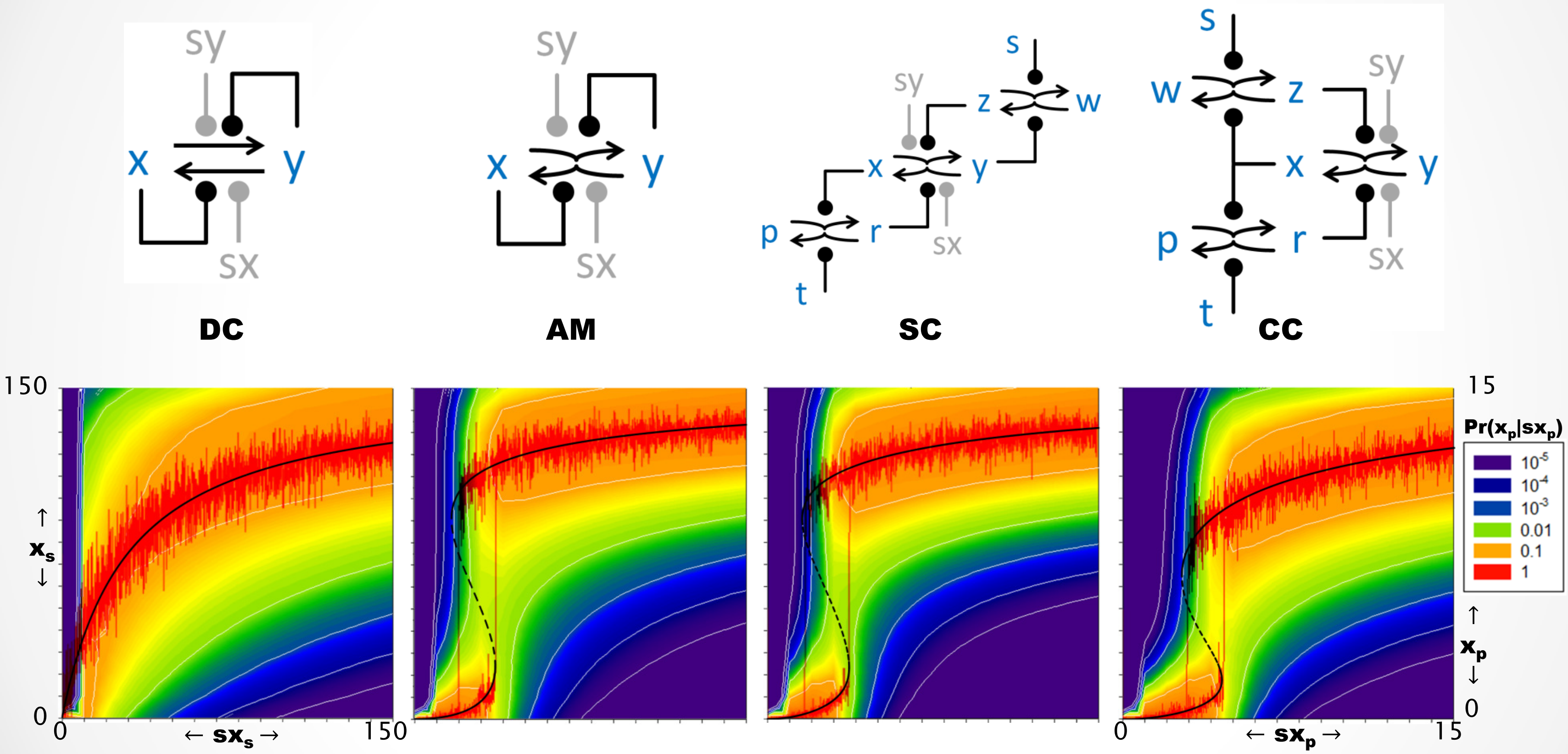
Switches as Computational Systems – Convergence



NEW!
CC
converges
in log time

Steady State Analysis

Switches as Dynamical Systems – Steady State Response



NEW!
AM shows
hysteresis

The Argument So Far

- Relating dynamical and computational systems in isolation (as *closed systems*)
  - The AM algorithm (network) implements an input-driven switching function (in addition to the known majority function).
  - The CC algorithm implements a input-less majority function (in addition to the known switching function).
  - The structures of AM and CC are related, and an intermediate network shares some properties of both.

- But what about the context?
  - Will AM and CC behave similarly in any context (as *open systems*)?
  - That's a hard question, so we look at their intended context: implementing oscillators.
  - Also, oscillators are almost the 'worst case' contexts: very sensitive to component behavior.

Oscillators

- Basic in Physics, studied by simple *phenomenological* (not structural) ODE models.

- Non-trivial in Chemistry: it was only discovered in the 20's (Lotka) that chemical systems can (theoretically) oscillate: before, oscillation was thought impossible. Shown experimentally only in the 50's.

- Mechanics (since antiquity) and modern Electronics (as well as Chemistry) must engineer the *network structure* of oscillators.

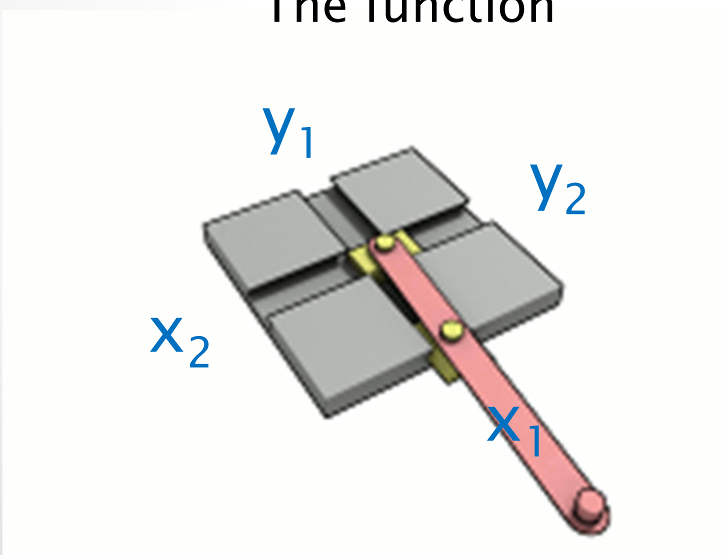
- Biology: all natural cycles are oscillators. Here we must reverse engineer their network structure.

- Computing: how can populations of agents (read: molecules) interact (network) to achieve oscillations?

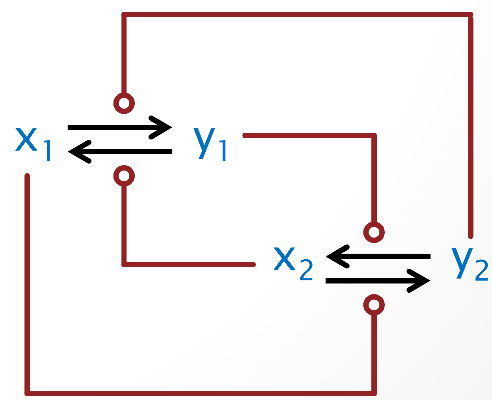
The Trammel of Archimedes

- A device to draw ellipses
  - Two interconnected **switches**.
  - When one switch is on (off) it flips the other switch on (off). When the other switch is on (off) it flips the first switch off (on).
  - The amplitude is kept constant by mechanical constraints.

The function



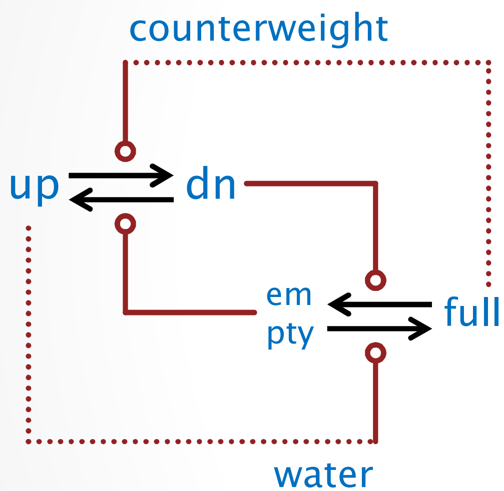
The network

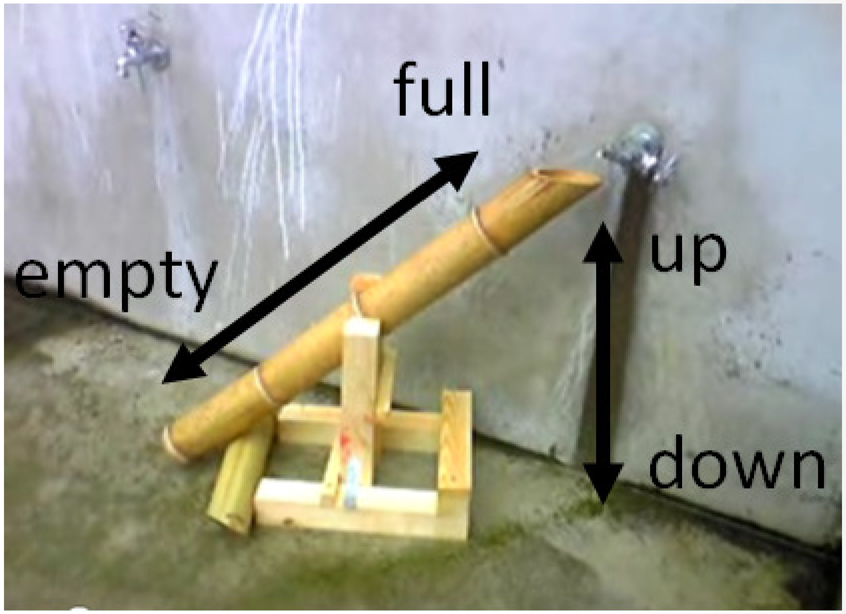


en.wikipedia.org/wiki/Trammel_of_Archimedes

The Shishi Odoshi

- A Japanese scarecrow (*lit.* scare-deer)
  - Used by Bela Novak to illustrate the cell cycle switch.





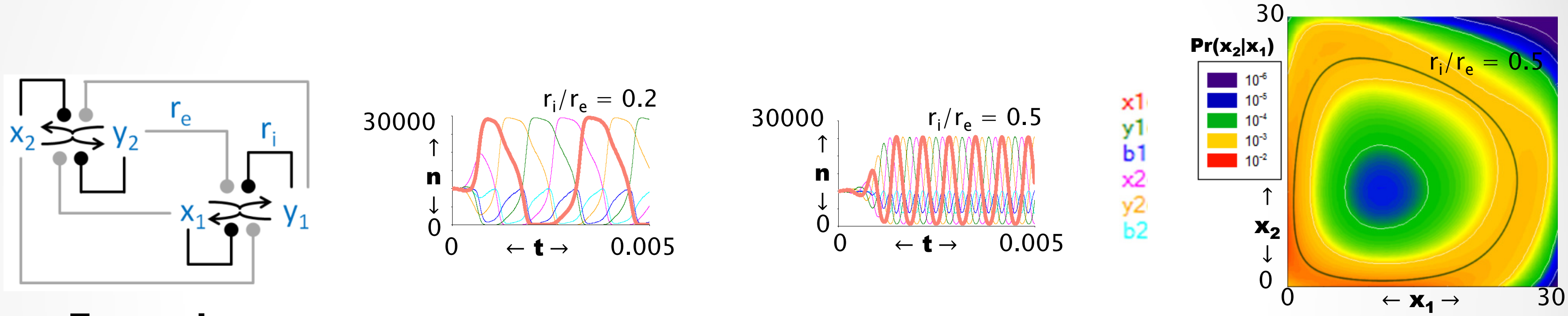
http://www.youtube.com/watch?v=VbvecTIftcE&NR=1&feature=fvwp

empty + up → up + full
up + full → full + dn
full + dn → dn + empty
dn + empty → empty + up

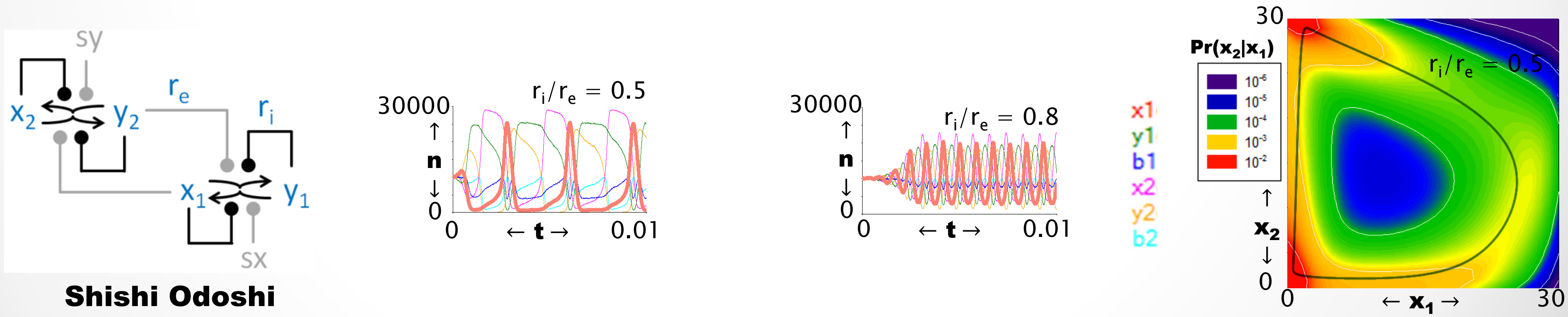
Outer switched connections replaced by constant influxes: tap water and gravity.

Contextual Analysis

AM switches in the context of larger networks (oscillators).



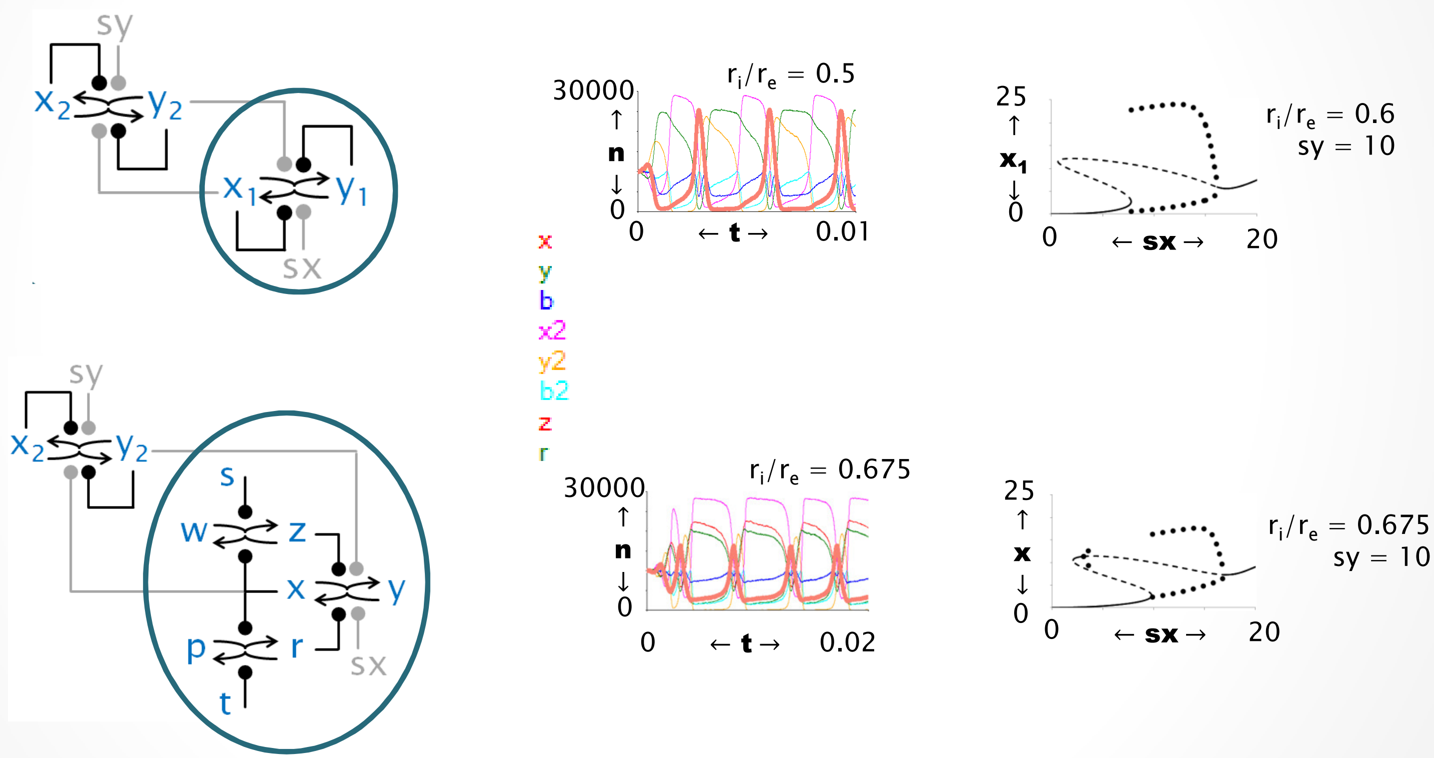
Trammel



Shishi Odoshi

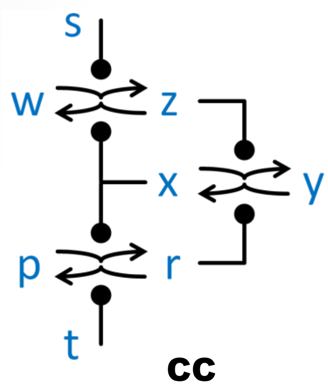
Modularity Analysis

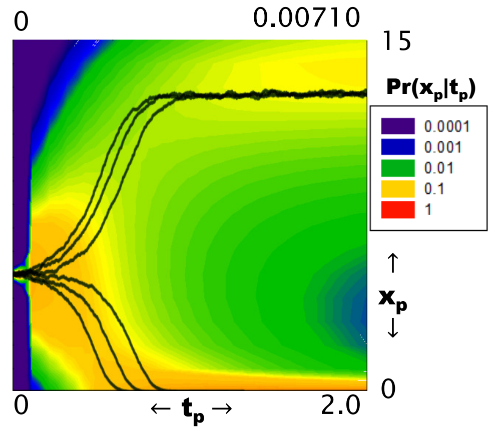
CC can be swapped in for AM.



But there was a difference

We have seen that the output of CC does not go ‘fully on’ like AM:





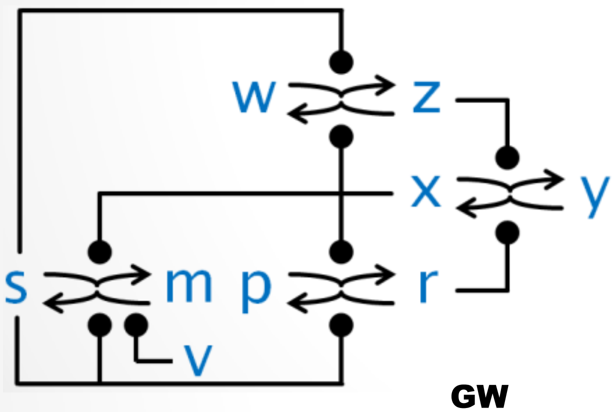
(And similarly the CC oscillator does not go ‘full on’.)

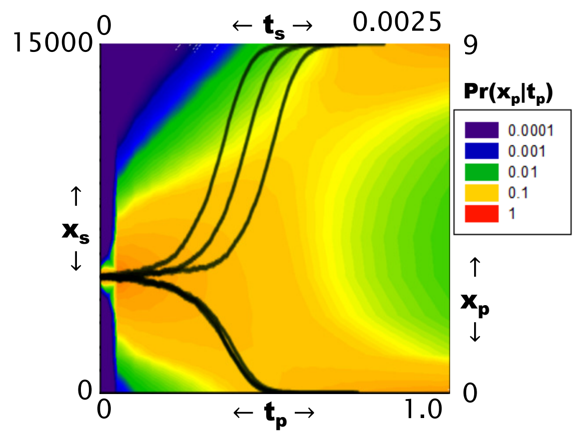
Because s continuously inhibits x through z, so that x cannot fully express. This could be solved if x would inhibit s in retaliation.

Q: How would *you* fix this Nobel-prize winning network?

Nature fixed it!

There is another known feedback loop in real cell cycle switches by which x suppresses s:



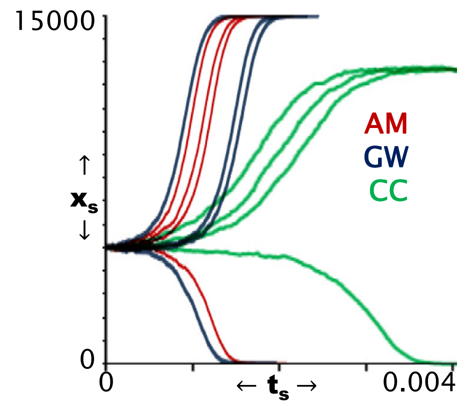


Full activation!

(Also, s and t happen to be the same molecule)

And made it fast too!

More surprising: the extra feedback also speeds up the decision time of the switch, making it about as good as the ‘optimal’ AM switch:



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

The Greatwall Kinase

- Another paper appeared the same week as ours:

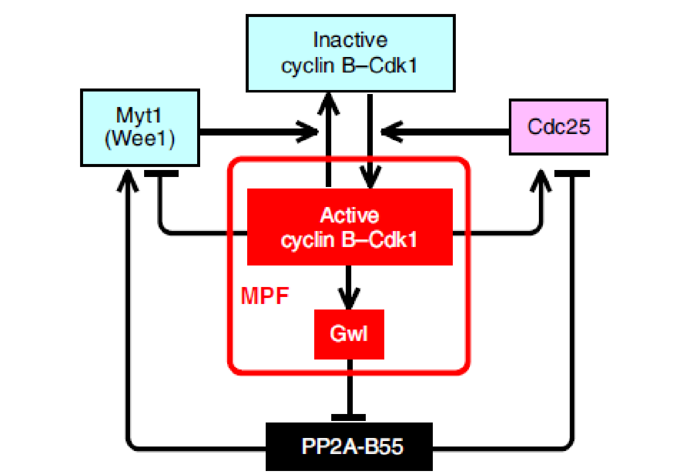


Figure 7 | MPF as a core component in the autoregulatory loop for cyclin B-Cdk1 activation. Cyclin B-Cdk1 is by itself very inefficient in triggering the autoregulatory loop in recipient oocytes, but MPF, consisting of both cyclin B–Cdk1 and Gwl, can efficiently initiate the activation loop, leading to full activation of cyclin B-Cdk1 in recipients.

- Showing experimentally that the (known) Greatwall/PP2A loop is a <u>necessary</u> component of the switch.

SCIENTIFIC REPORTS

OPEN

SUBJECT AREAS:
COMPUTATIONAL BIOLOGY

The Cell Cycle Switch Computes Approximate Majority

Luca Cardelli[1] & Attila Csikász-Nagy[2,3]

nature COMMUNICATIONS

OPEN

ARTICLE

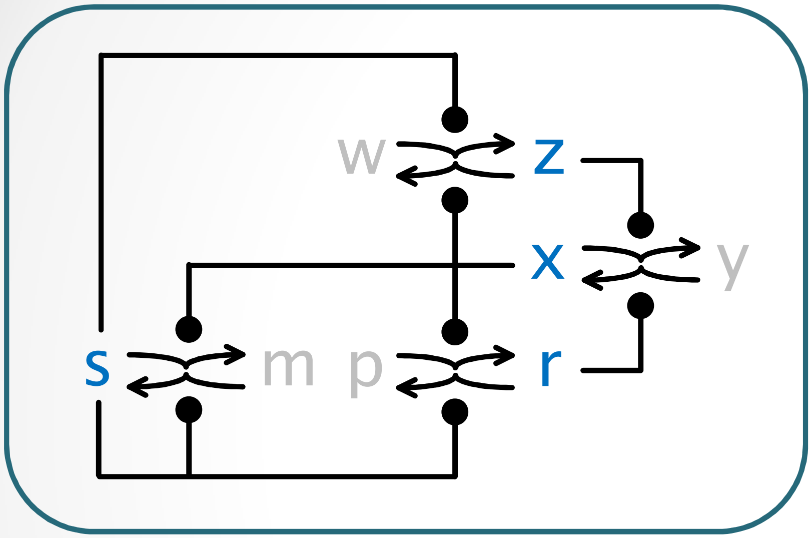
Received 6 Jul 2012 | Accepted 14 Aug 2012 | Published 11 Sep 2012
DOI: 10.1038/ncomms2062

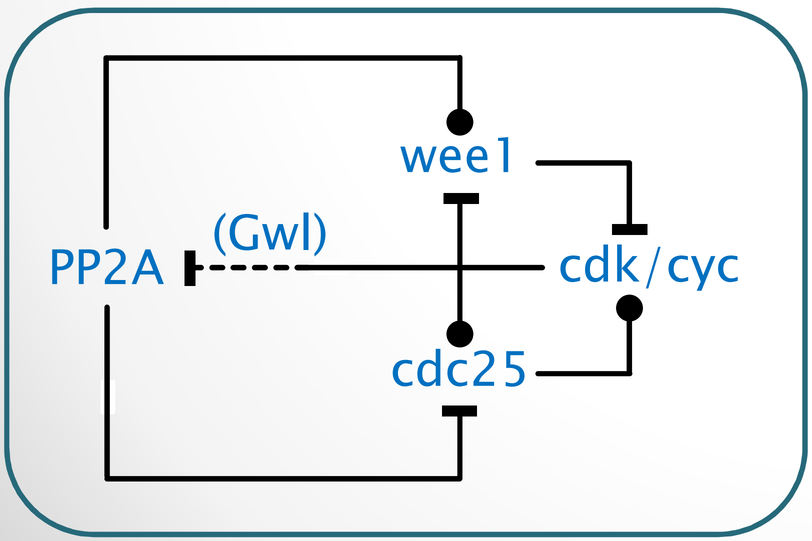
Greatwall kinase and cyclin B-Cdk1 are both critical constituents of M-phase-promoting factor

Masatoshi Hara[1,†], Yusuke Abe[1,†], Toshiaki Tanaka[2], Takayoshi Yamamoto[1,†], Eiichi Okumura[1] & Takeo Kishimoto[1]

Same as ours

Our Network





Their Network

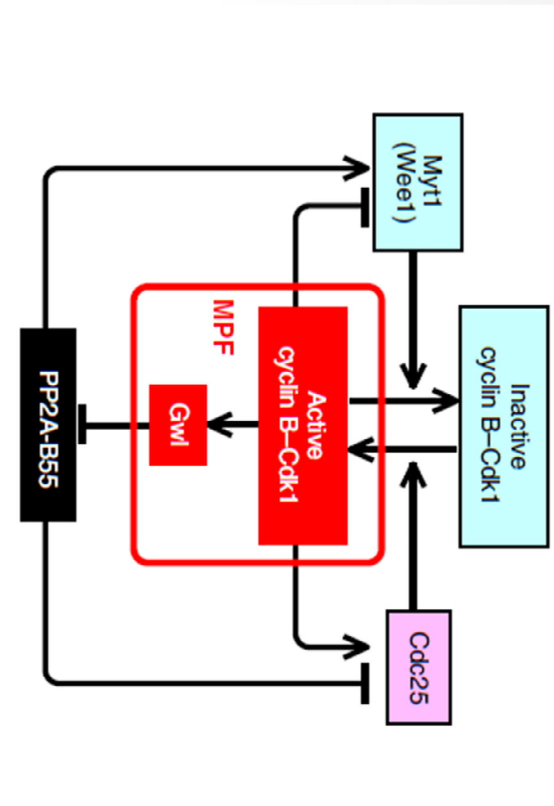
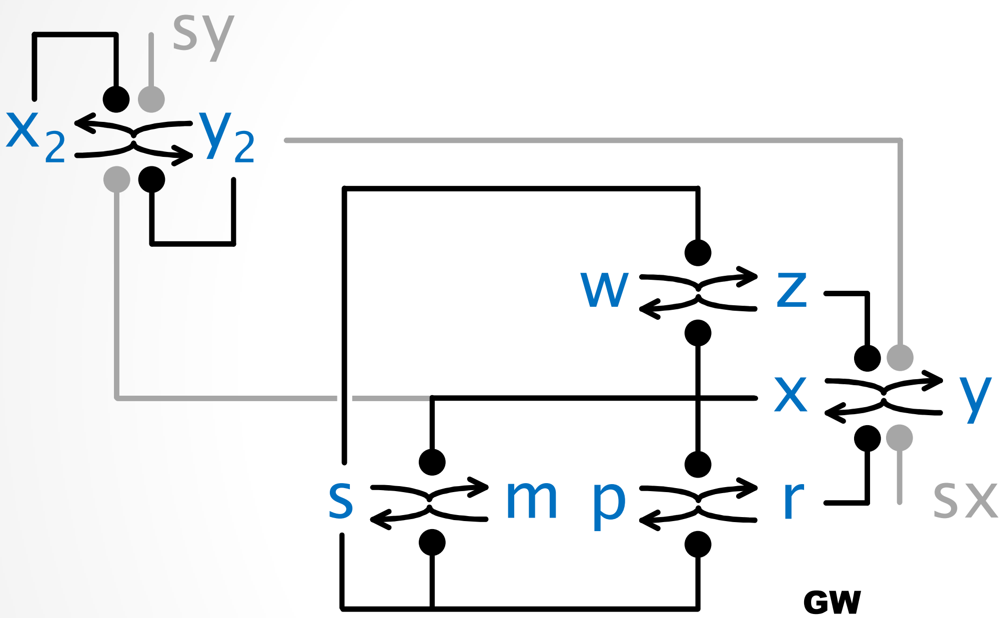


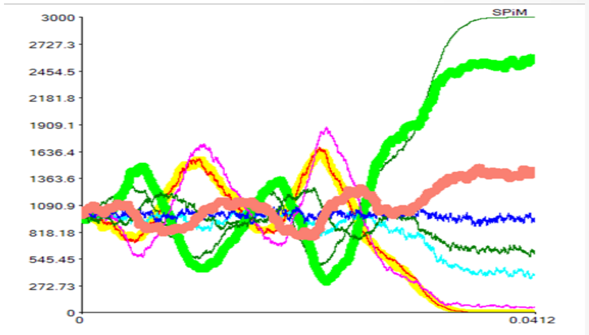
Figure 7 | MPF as a core component in the autoregulatory loop for cyclin B-Cdk1 activation. Cyclin B-Cdk1 is by itself very inefficient in triggering the autoregulatory loop in recipient oocytes, but MPF, consisting of both cyclin B-Cdk1 and Gwl, can efficiently initiate the activation loop, leading to full activation of cyclin B-Cdk1 in recipients.

Basically an experimental validation that the real CC is really a good AM.

A new switch candidate: GW

- Will it work in the oscillator?

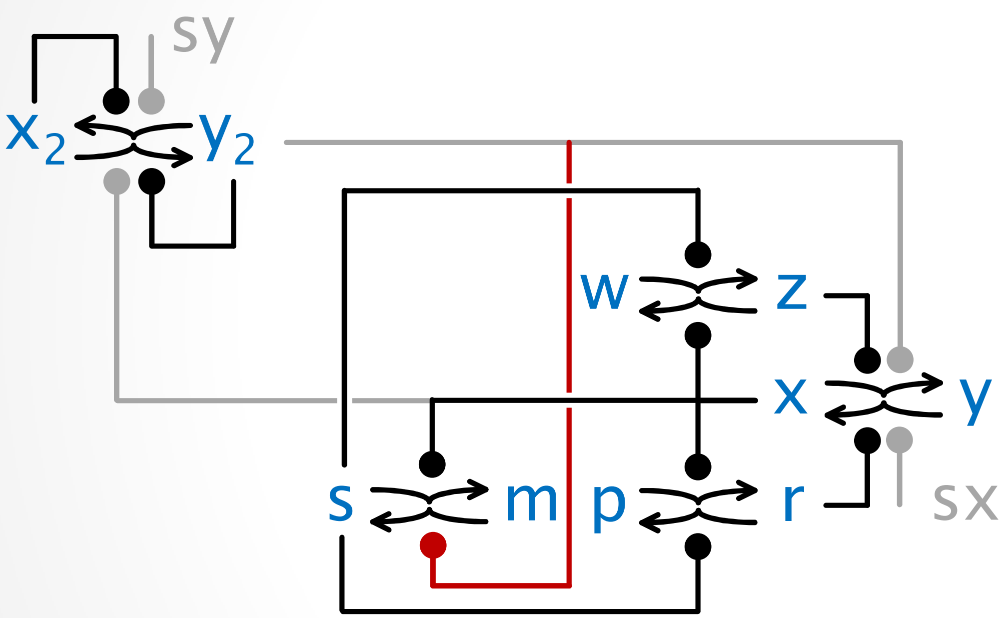


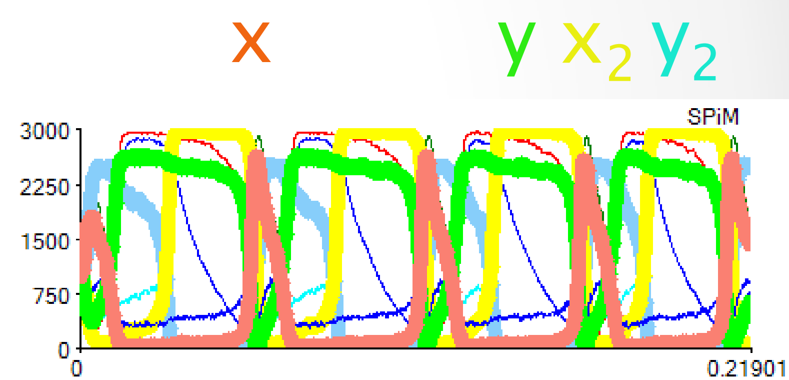


- Absolutely not! ☹
    - The x stable state is just too strong: a high x will shut down s completely; which means that r will be fully on, and it in turn will reinforce x fully. And y_2 can never be strong enough to push down x when x-r are in such a strong mutual feedback. No amount of fiddling seems to give enough control on that situation.

However this will

- Neatly closing up all the loose ends:
  - Put s under control of y_2, so y_2 can succesfully undermine x.

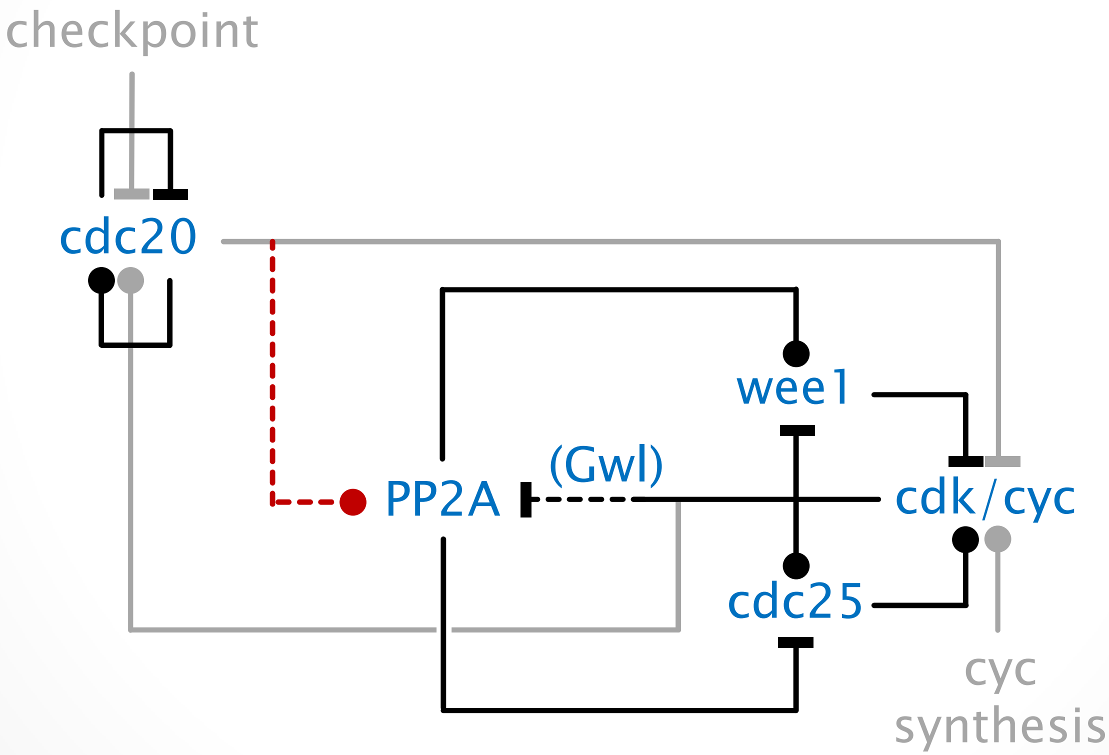




- Beautifully spiky and full-on oscillations.
  - On the first try, with all default parameters:
    all black rates 1.0, all gray (&red) rates 0.5, all initial amounts equal.

A new scientific hypothesis

- Hence, a condition for robust oscillations:
  - Either Gwl or PP2A or something along that path must be under control of cdc20.



  - There are some hints in the literature that this may be the case, but no direct experimental validation.

Summary

- The structure of AM implements an input-driven switching function (in addition to the known majority function).
- The structure of CC/GW implements a input-less majority function (in addition to the known switching function).
- The structures of AM and CC/GW are related, and an intermediate network shares the properties of both.
- The behaviors of AM and CC/GW in isolation are related.
- The behaviors of AM and CC/GW in oscillator contexts are related.
- A refinement (GW) of the core CC network, known to occur in nature, improves switching performance and brings it in line with AM performance.

Computational Outlook

Computational viewpoint

- Cells are computational engines
  - Their *primary* function is information processing
    - Which controls feeding, escape, and reproduction.
    - Without properly processing information cells soon die (by starvation or predation).
    - Hence a strong pressure to process information better.
  - That *happens* to be implemented by chemistry
    - Fundamental is not the 'hardware' (proteins etc.) which easily varies between organisms but the 'software' the runs on the hardware.
- So, what algorithms do they run?

Reverse Engineering

- Q (traditional): What kind of dynamical system is the cell-cycle switch?
- A (traditional): Bistability – ultrasensitivity – hysteresis … Focused on how unstructured sub-populations change over time.

- Q: What kind of algorithmic system is the cell-cylce switch?
- A: Interaction – complexity – convergence ... Focused on individual molecules as programmable, structured, algorithmic entities.

- Leading to a better understanding of not just the *function* but also the *network* (algorithm).

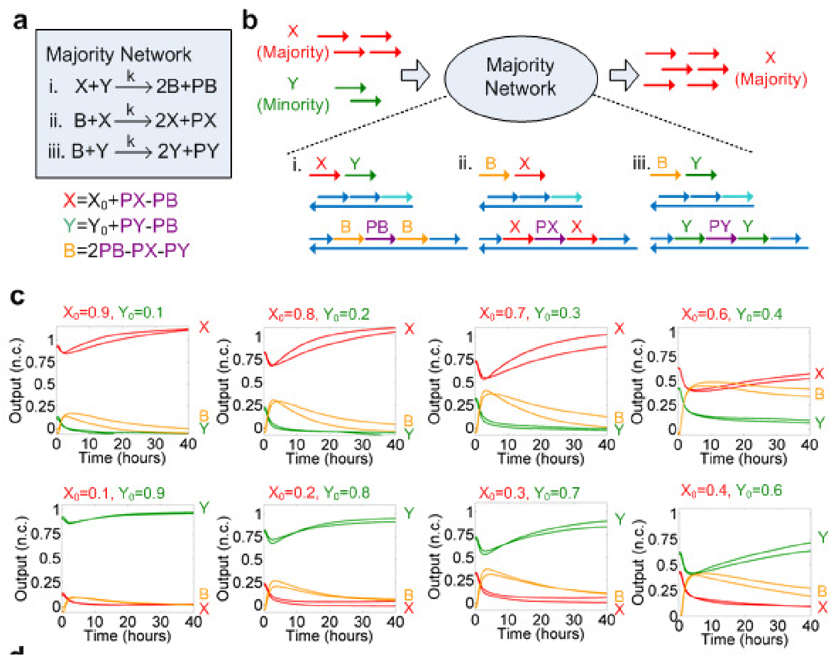
Direct Engineering

- The AM algorithm was not learned from nature
  - CC was invented ~2.7 billions years ago.
  - AM was invented ~6 years ago (but independently).

- But nature may have more tricks
  - If there is some clever population algorithm out there, how will we recognize it?
  - We need to understand better how nature operates.

In separate work...

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



A DNA Realization of Chemical Reaction Networks

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

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