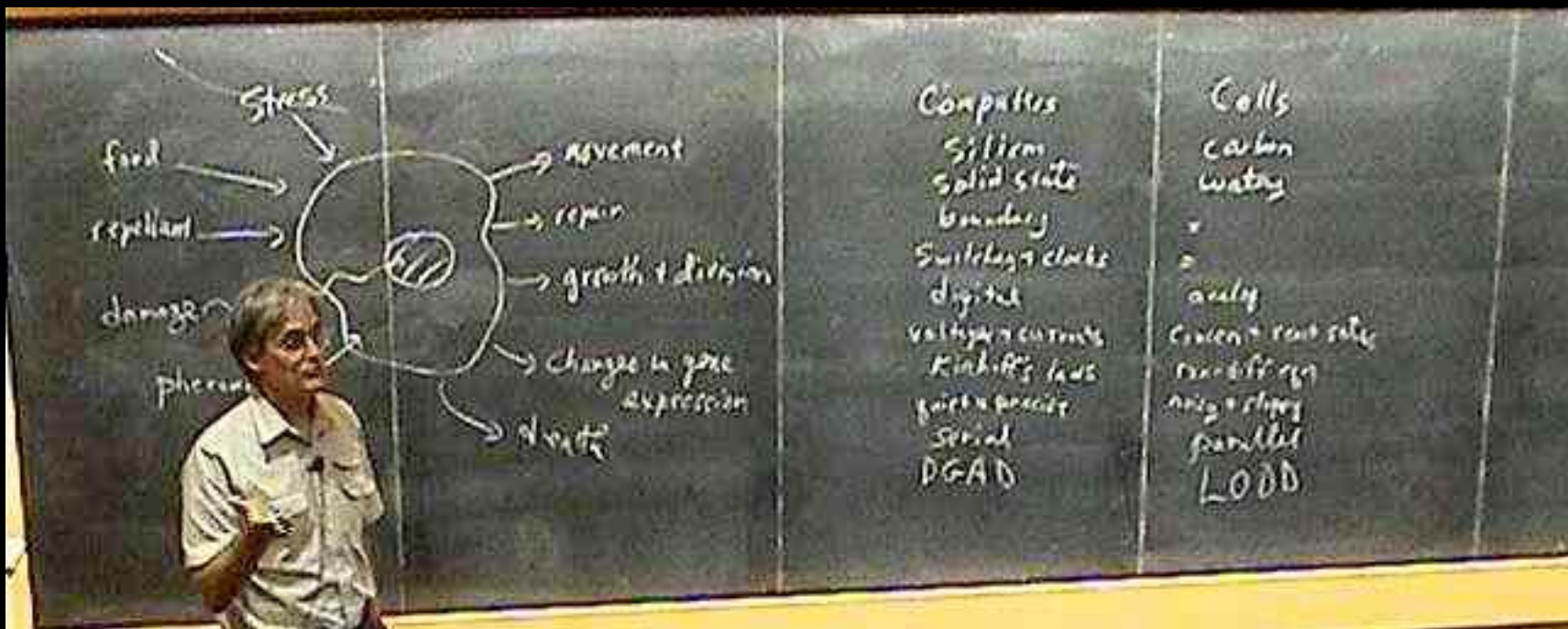


*βgF* *βgH*  
*βgC* *βgE* *βgG*  
*βgA* *βgB*  
*βgD* *βgF* *βgH*  
*βgI* *βgJ*  
*βgK* *βgL*  
*βgM* *βgN*  
*βgO* *βgP*  
*βgQ* *βgR*  
*βgS* *βgT*  
*βgU* *βgV*  
*βgW* *βgX*  
*βgY* *βgZ*  
*βgAA* *βgAB*  
*βgAC* *βgAD*  
*βgAE* *βgAF*  
*βgAG* *βgAH*  
*βgAI* *βgAJ*  
*βgAK* *βgAL*  
*βgAM* *βgAN*  
*βgAO* *βgAP*  
*βgAQ* *βgAR*  
*βgAS* *βgAT*  
*βgAU* *βgAV*  
*βgAW* *βgAX*  
*βgAY* *βgAZ*  
*βgBA* *βgBB*  
*βgBC* *βgBD*  
*βgBE* *βgBF*  
*βgBG* *βgBH*  
*βgBI* *βgBJ*  
*βgBK* *βgBL*  
*βgBM* *βgBN*  
*βgBO* *βgBP*  
*βgBQ* *βgBR*  
*βgBS* *βgBT*  
*βgBU* *βgBV*  
*βgBW* *βgBX*  
*βgBY* *βgBZ*  
*βgCA* *βgCB*  
*βgCC* *βgCD*  
*βgCE* *βgCF*  
*βgCG* *βgCH*  
*βgCI* *βgCJ*  
*βgCK* *βgCL*  
*βgCM* *βgCN*  
*βgCO* *βgCP*  
*βgCQ* *βgCR*  
*βgCS* *βgCT*  
*βgCU* *βgCV*  
*βgCW* *βgCX*  
*βgCY* *βgCZ*  
*βgDA* *βgDB*  
*βgDC* *βgDD*  
*βgDE* *βgDF*  
*βgDG* *βgDH*  
*βgDI* *βgDJ*  
*βgDK* *βgDL*  
*βgDM* *βgDN*  
*βgDO* *βgDP*  
*βgDQ* *βgDR*  
*βgDS* *βgDT*  
*βgDU* *βgDV*  
*βgDW* *βgDX*  
*βgDY* *βgDZ*  
*βgEA* *βgEB*  
*βgEC* *βgED*  
*βgEE* *βgEF*  
*βgEG* *βgEH*  
*βgEI* *βgEJ*  
*βgEK* *βgEL*  
*βgEM* *βgEN*  
*βgEO* *βgEP*  
*βgEQ* *βgER*  
*βgES* *βgET*  
*βgEU* *βgEV*  
*βgEW* *βgEX*  
*βgEY* *βgEZ*  
*βgFA* *βgFB*  
*βgFC* *βgFD*  
*βgFE* *βgFF*  
*βgFG* *βgFH*  
*βgFI* *βgFJ*  
*βgFK* *βgFL*  
*βgFM* *βgFN*  
*βgFO* *βgFP*  
*βgFQ* *βgFR*  
*βgFS* *βgFT*  
*βgFU* *βgFV*  
*βgFW* *βgFX*  
*βgFY* *βgFZ*  
*βgGA* *βgGB*  
*βgGC* *βgGD*  
*βgGE* *βgGF*  
*βgGG* *βgGH*  
*βgGI* *βgGJ*  
*βgGK* *βgGL*  
*βgGM* *βgGN*  
*βgGO* *βgGP*  
*βgGQ* *βgGR*  
*βgGS* *βgGT*  
*βgGU* *βgGV*  
*βgGW* *βgGX*  
*βgGY* *βgGZ*  
*βgHA* *βgHB*  
*βgHC* *βgHD*  
*βgHE* *βgHF*  
*βgHG* *βgHH*  
*βgHI* *βgHJ*  
*βgHK* *βgHL*  
*βgHM* *βgHN*  
*βgHO* *βgHP*  
*βgHQ* *βgHR*  
*βgHS* *βgHT*  
*βgHU* *βgHV*  
*βgHW* *βgHX*  
*βgHY* *βgHZ*  
*βgIA* *βgIB*  
*βgIC* *βgID*  
*βgIE* *βgIF*  
*βgIG* *βgIH*  
*βgII* *βgIJ*  
*βgIK* *βgIL*  
*βgIM* *βgIN*  
*βgIO* *βgIP*  
*βgIQ* *βgIR*  
*βgIS* *βgIT*  
*βgIU* *βgIV*  
*βgIW* *βgIX*  
*βgIY* *βgIZ*  
*βgJA* *βgJB*  
*βgJC* *βgJD*  
*βgJE* *βgJF*  
*βgJG* *βgJH*  
*βgJI* *βgJJ*  
*βgJK* *βgJL*  
*βgJM* *βgJN*  
*βgJO* *βgJP*  
*βgJQ* *βgJR*  
*βgJS* *βgJT*  
*βgJU* *βgJV*  
*βgJW* *βgJX*  
*βgJY* *βgJZ*  
*βgKA* *βgKB*  
*βgKC* *βgKD*  
*βgKE* *βgKF*  
*βgKG* *βgKH*  
*βgKI* *βgKJ*  
*βgKK* *βgKL*  
*βgKM* *βgKN*  
*βgKO* *βgKP*  
*βgKQ* *βgKR*  
*βgKS* *βgKT*  
*βgKU* *βgKV*  
*βgKW* *βgKX*  
*βgKY* *βgKZ*  
*βgLA* *βgLB*  
*βgLC* *βgLD*  
*βgLE* *βgLF*  
*βgLG* *βgLH*  
*βgLI* *βgLJ*  
*βgLK* *βgLL*  
*βgLM* *βgLN*  
*βgLO* *βgLP*  
*βgLQ* *βgLR*  
*βgLS* *βgLT*  
*βgLU* *βgLV*  
*βgLW* *βgLX*  
*βgLY* *βgLZ*  
*βgMA* *βgMB*  
*βgMC* *βgMD*  
*βgME* *βgMF*  
*βgMG* *βgMH*  
*βgMI* *βgMJ*  
*βgMK* *βgML*  
*βgMM* *βgMN*  
*βgMO* *βgMP*  
*βgMQ* *βgMR*  
*βgMS* *βgMT*  
*βgMU* *βgMV*  
*βgMW* *βgMX*  
*βgMY* *βgMZ*  
*βgNA* *βgNB*  
*βgNC* *βgND*  
*βgNE* *βgNF*  
*βgNG* *βgNH*  
*βgNI* *βgNJ*  
*βgNK* *βgNL*  
*βgNM* *βgNN*  
*βgNO* *βgNP*  
*βgNQ* *βgNR*  
*βgNS* *βgNT*  
*βgNU* *βgNV*  
*βgNW* *βgNX*  
*βgNY* *βgNZ*  
*βgOA* *βgOB*  
*βgOC* *βgOD*  
*βgOE* *βgOF*  
*βgOG* *βgOH*  
*βgOI* *βgOJ*  
*βgOK* *βgOL*  
*βgOM* *βgON*  
*βgOO* *βgOP*  
*βgOQ* *βgOR*  
*βgOS* *βgOT*  
*βgOU* *βgOV*  
*βgOW* *βgOX*  
*βgOY* *βgOZ*  
*βgPA* *βgPB*  
*βgPC* *βgPD*  
*βgPE* *βgPF*  
*βgPG* *βgPH*  
*βgPI* *βgPJ*  
*βgPK* *βgPL*  
*βgPM* *βgPN*  
*βgPO* *βgPP*  
*βgPQ* *βgPR*  
*βgPS* *βgPT*  
*βgPU* *βgPV*  
*βgPW* *βgPX*  
*βgPY* *βgPZ*  
*βgQA* *βgQB*  
*βgQC* *βgQD*  
*βgQE* *βgQF*  
*βgQG* *βgQH*  
*βgQI* *βgQJ*  
*βgQK* *βgQL*  
*βgQM* *βgQN*  
*βgQO* *βgQP*  
*βgQQ* *βgQR*  
*βgQS* *βgQT*  
*βgQU* *βgQV*  
*βgQW* *βgQX*  
*βgQY* *βgQZ*  
*βgRA* *βgRB*  
*βgRC* *βgRD*  
*βgRE* *βgRF*  
*βgRG* *βgRH*  
*βgRI* *βgRJ*  
*βgRK* *βgRL*  
*βgRM* *βgRN*  
*βgRO* *βgRP*  
*βgRQ* *βgRR*  
*βgRS* *βgRT*  
*βgRU* *βgRV*  
*βgRW* *βgRX*  
*βgRY* *βgRZ*  
*βgSA* *βgSB*  
*βgSC* *βgSD*  
*βgSE* *βgSF*  
*βgSG* *βgSH*  
*βgSI* *βgSJ*  
*βgSK* *βgSL*  
*βgSM* *βgSN*  
*βgSO* *βgSP*  
*βgSQ* *βgSR*  
*βgSS* *βgST*  
*βgSU* *βgSV*  
*βgSW* *βgSX*  
*βgSY* *βgSZ*  
*βgTA* *βgTB*  
*βgTC* *βgTD*  
*βgTE* *βgTF*  
*βgTG* *βgTH*  
*βgTI* *βgTJ*  
*βgTK* *βgTL*  
*βgTM* *βgTN*  
*βgTO* *βgTP*  
*βgTQ* *βgTR*  
*βgTS* *βgTT*  
*βgTU* *βgTV*  
*βgTW* *βgTX*  
*βgTY* *βgTZ*  
*βgUA* *βgUB*  
*βgUC* *βgUD*  
*βgUE* *βgUF*  
*βgUG* *βgUH*  
*βgUI* *βgUJ*  
*βgUK* *βgUL*  
*βgUM* *βgUN*  
*βgUO* *βgUP*  
*βgUQ* *βgUR*  
*βgUS* *βgUT*  
*βgUU* *βgUV*  
*βgUW* *βgUX*  
*βgUY* *βgUZ*  
*βgVA* *βgVB*  
*βgVC* *βgVD*  
*βgVE* *βgVF*  
*βgVG* *βgVH*  
*βgVI* *βgVJ*  
*βgVK* *βgVL*  
*βgVM* *βgVN*  
*βgVO* *βgVP*  
*βgVQ* *βgVR*  
*βgVS* *βgVT*  
*βgVU* *βgVV*  
*βgVW* *βgVX*  
*βgVY* *βgVZ*  
*βgWA* *βgWB*  
*βgWC* *βgWD*  
*βgWE* *βgWF*  
*βgWG* *βgWH*  
*βgWI* *βgWJ*  
*βgWK* *βgWL*  
*βgWM* *βgWN*  
*βgWO* *βgWP*  
*βgWQ* *βgWR*  
*βgWS* *βgWT*  
*βgWU* *βgWV*  
*βgWW* *βgWX*  
*βgWY* *βgWZ*  
*βgXA* *βgXB*  
*βgXC* *βgXD*  
*βgXE* *βgXF*  
*βgXG* *βgXH*  
*βgXI* *βgXJ*  
*βgXK* *βgXL*  
*βgXM* *βgXN*  
*βgXO* *βgXP*  
*βgXQ* *βgXR*  
*βgXS* *βgXT*  
*βgXU* *βgXV*  
*βgXW* *βgXX*  
*βgXY* *βgXZ*  
*βgYA* *βgYB*  
*βgYC* *βgYD*  
*βgYE* *βgYF*  
*βgYG* *βgYH*  
*βgYI* *βgYJ*  
*βgYK* *βgYL*  
*βgYM* *βgYN*  
*βgYO* *βgYP*  
*βgYQ* *βgYR*  
*βgYS* *βgYT*  
*βgYU* *βgYV*  
*βgYW* *βgYX*  
*βgYY* *βgYZ*  
*βgZA* *βgZB*  
*βgZC* *βgZD*  
*βgZE* *βgZF*  
*βgZG* *βgZH*  
*βgZI* *βgZJ*  
*βgZK* *βgZL*  
*βgZM* *βgZN*  
*βgZO* *βgZP*  
*βgZQ* *βgZR*  
*βgZS* *βgZT*  
*βgZU* *βgZV*  
*βgZW* *βgZX*  
*βgZY* *βgZZ*

# Discrete Dynamical Networks as Toy Models for Computation in the Cell

Stefan Bornholdt

Institute for Theoretical Physics  
University of Bremen

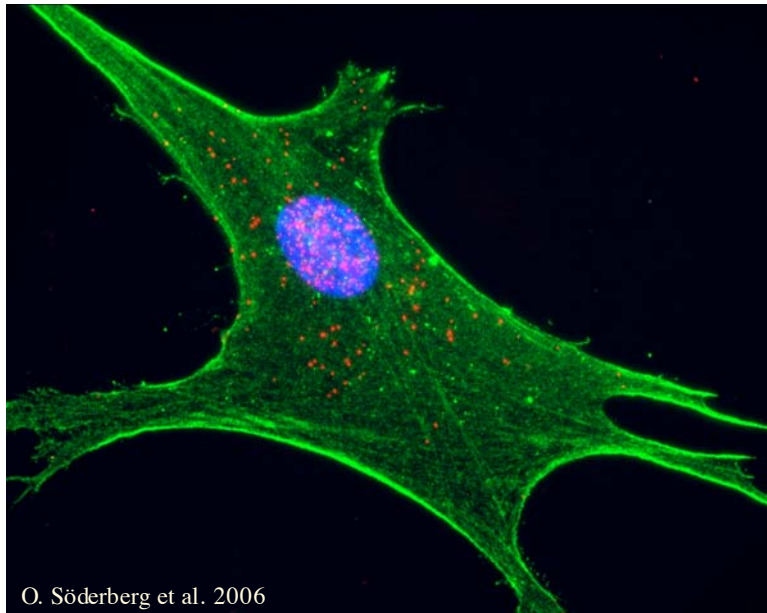


## How Do Cells Compute?

... KITP Blackboard  
lunch 23 July 2007

- Computers and the living cell
- Discrete networks as models for cellular computation
- A biological example: The yeast cell cycle
- Discrete network models and stochastic dynamics
- Applications and outlook

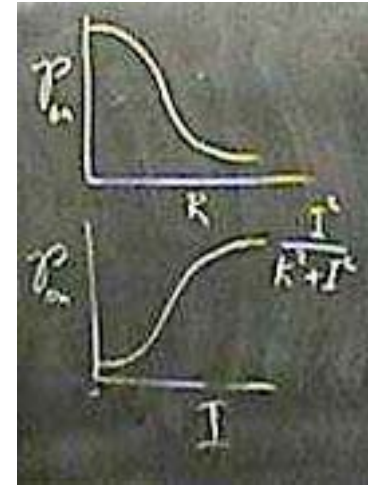
# Computation in the Cell?



- Adapting and reacting to environment: **analog computation?**
- Controlling sequences of events, cell-cycle, multicellular development, etc.: **digital computation?**

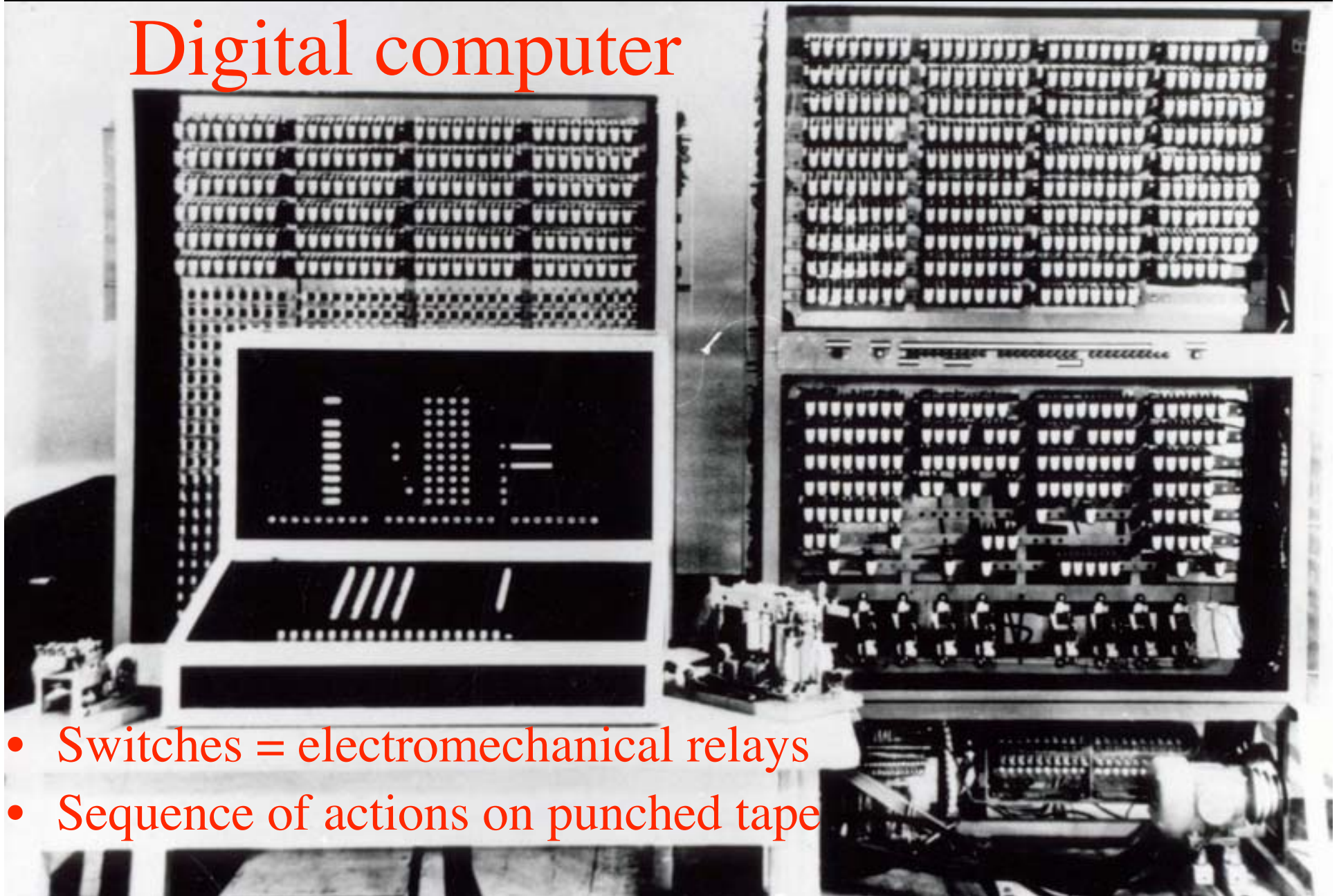
# Computational elements in the cell

- This workshop: We've seen many molecular regulatory elements with **binary** characteristics,
- and even **bistable** switches.



- Digital variables  $\{0,1\}$  can be represented:
- Elements for digital computation exist in the cell!
- So... is there any digital computation in the cell?

# Digital computer

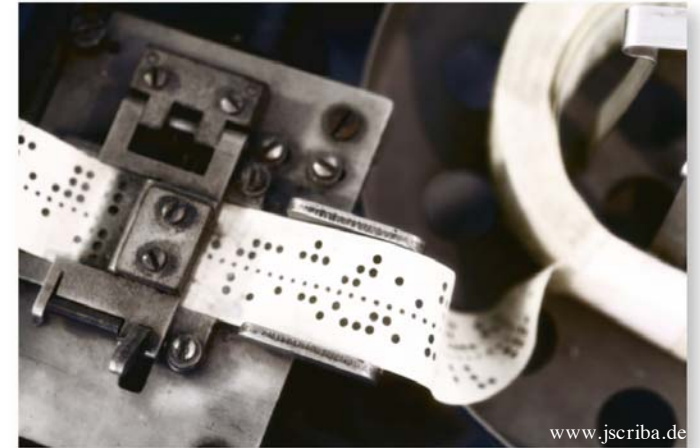


- Switches = electromechanical relays
- Sequence of actions on punched tape

(K. Zuse, 1941)

# Sequence control options

- **Computer:** Desired sequence of actions is stored on the tape
- **Dynamical System:** Sequence emerges as dynamical trajectory of the system, determined by the circuitry of the system



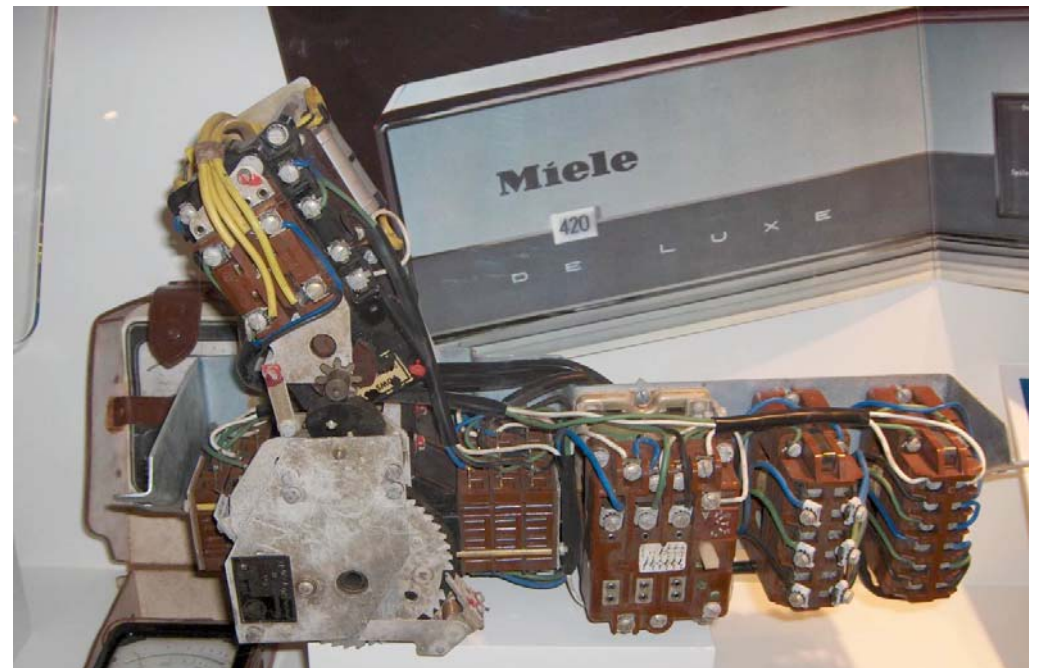
- If you had the choice: punched tape is easiest
- If all you have is squishy stuff (water, molecules, ...) and have to generate a sequence from that: There is the dynamical systems approach, only.
- Problem: How to reliably generate a sequence of actions from squishy building blocks?

# Engineering example: Controlling a washing machine



- **Software:** Sequence of switching events, controlling pumps, valves, motors, heater...
- **Input:** Switches, temperature probes, water level ...
- **Output:** Sequence of events, in response to selected program, temperature, water level, etc.

- **Hardware:** Similar to punched tape computer
- Switching disks mounted on common axle, driven by a motor





# Analogy:

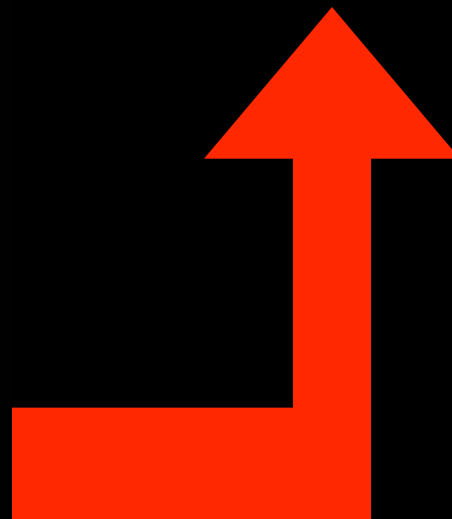
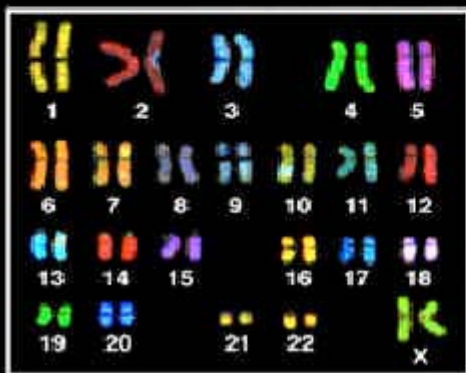
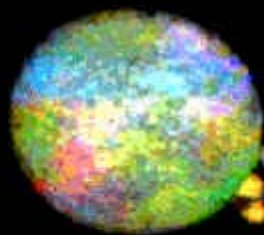
## Controlling the cell cycle

- **Software:** Desired **sequence** of Gene/Protein activation states
- **Input:** External signals, cell size, temperature, etc.
- **Output:** Sequence of molecular activation patterns in response to external and internal signals
  
- **Hardware:** Molecular network, analog, autonomous dynamics, continuously updated (no computer clock cycle), many elements with tendency to binary states.

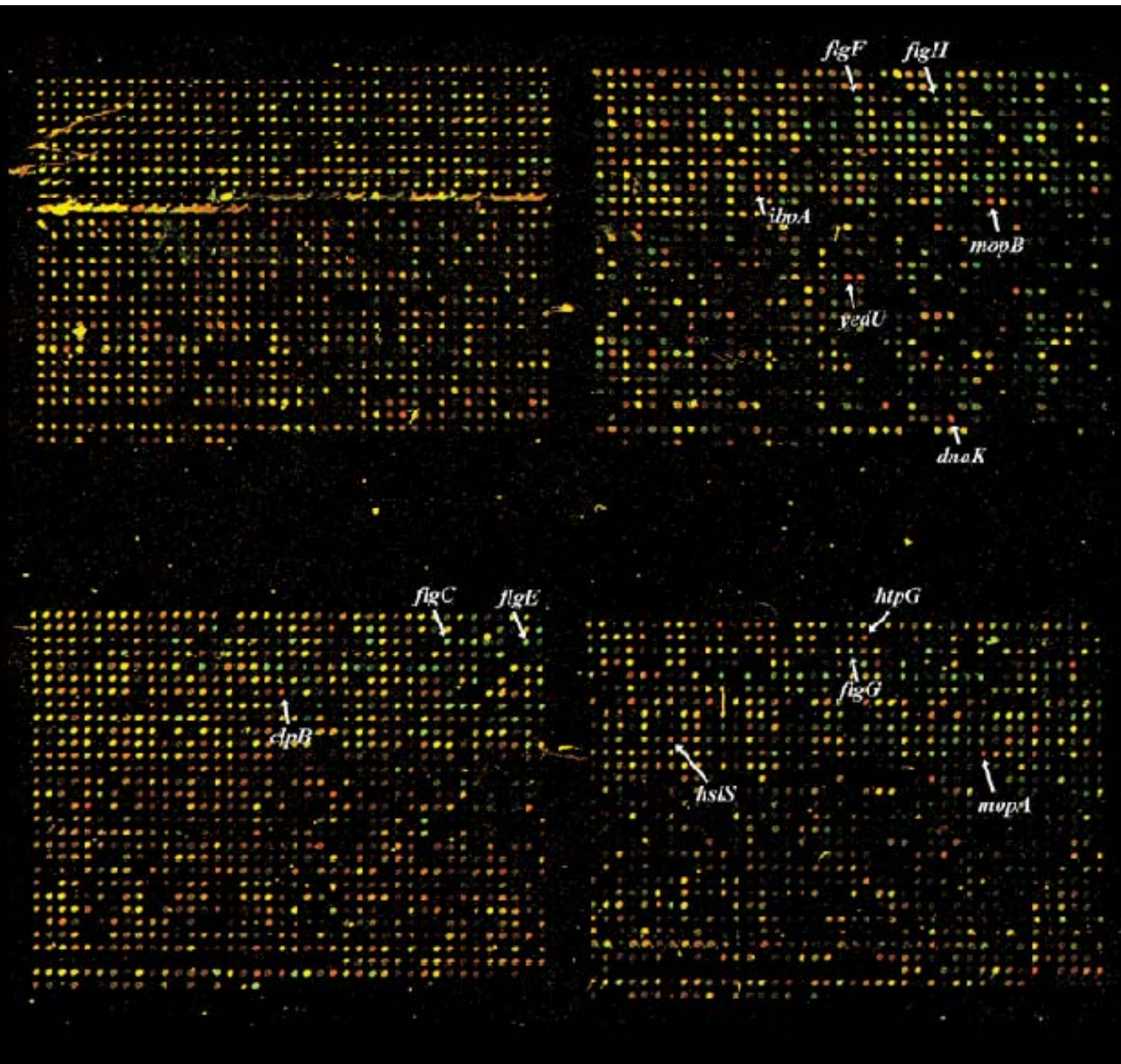
# Dynamics of networks of switches

- **Massive simplification** of biochemical networks, what can we learn from it?
- **Idea:**
  - Drop prediction of **time**.
  - Keep the requirement to predict **ordered sequences of activation patterns**.  
This is the „**Software**“ in the analogy picture.
- Engineering knowledge applicable to this „software layer“?
- How can networks of unreliable elements work reliably?

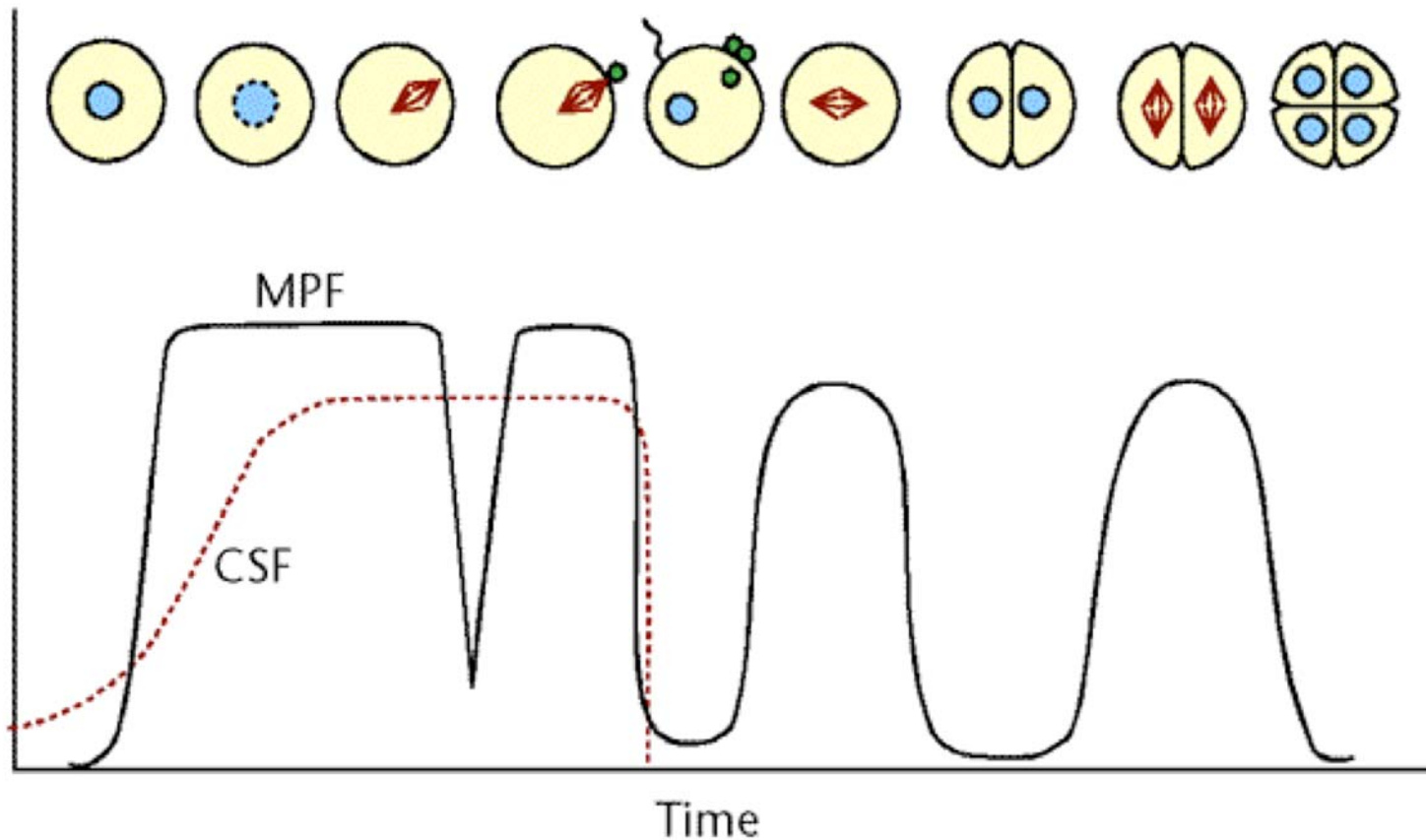
**Control on the systems level is very reliable!**



- Computers and the living cell
- Discrete networks as models for cellular computation
- A biological example: The yeast cell cycle
- Discrete network models and stochastic dynamics
- Applications and outlook



# Dynamics of genes (yeast)

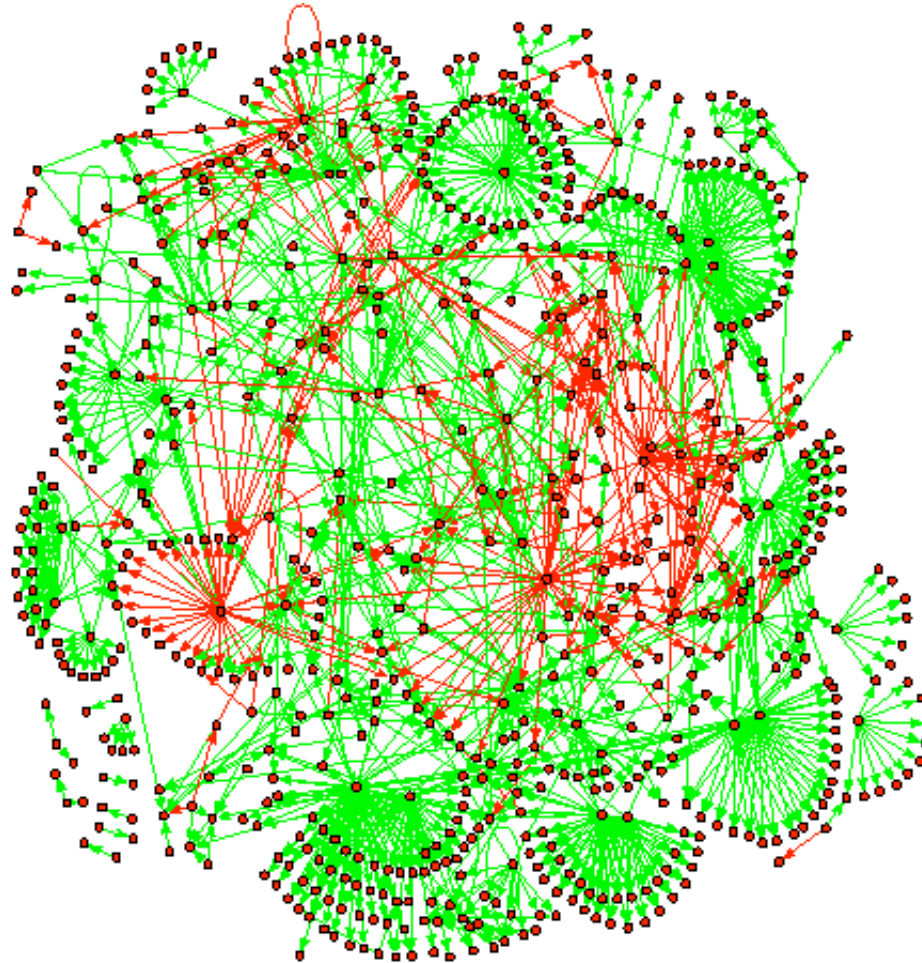


**Remarkable:** steep flanks and plateaus

**Hypothesis:** Represent a gene by a switch-like dynamics

# Gene regulation network of yeast

Simplified levels of **activation** and **inhibition**:



[S. Maslov and K. Sneppen, 2003]

# Discrete dynamical networks as models for gene regulation have been around: Boolean networks [Kauffman 1969]

$$S_i(t+1) = \text{sgn} \sum_{j=1}^N J_{ij} S_j(t) \quad S_i \in \{-1, +1\} \quad J_{ij} \in \{-1, 0, +1\}$$

asymmetric!

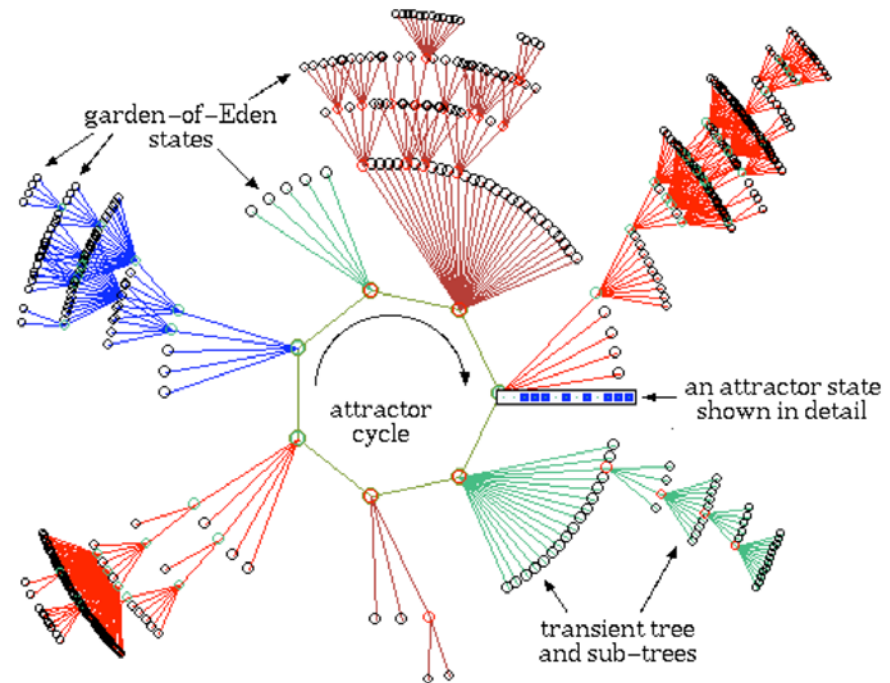
## Dynamics:

Transients, attractors with lengths scaling with system size as  $t \sim e^N$  for overcritical connectivity  $K > K_c \sim 2$

Few attractors:

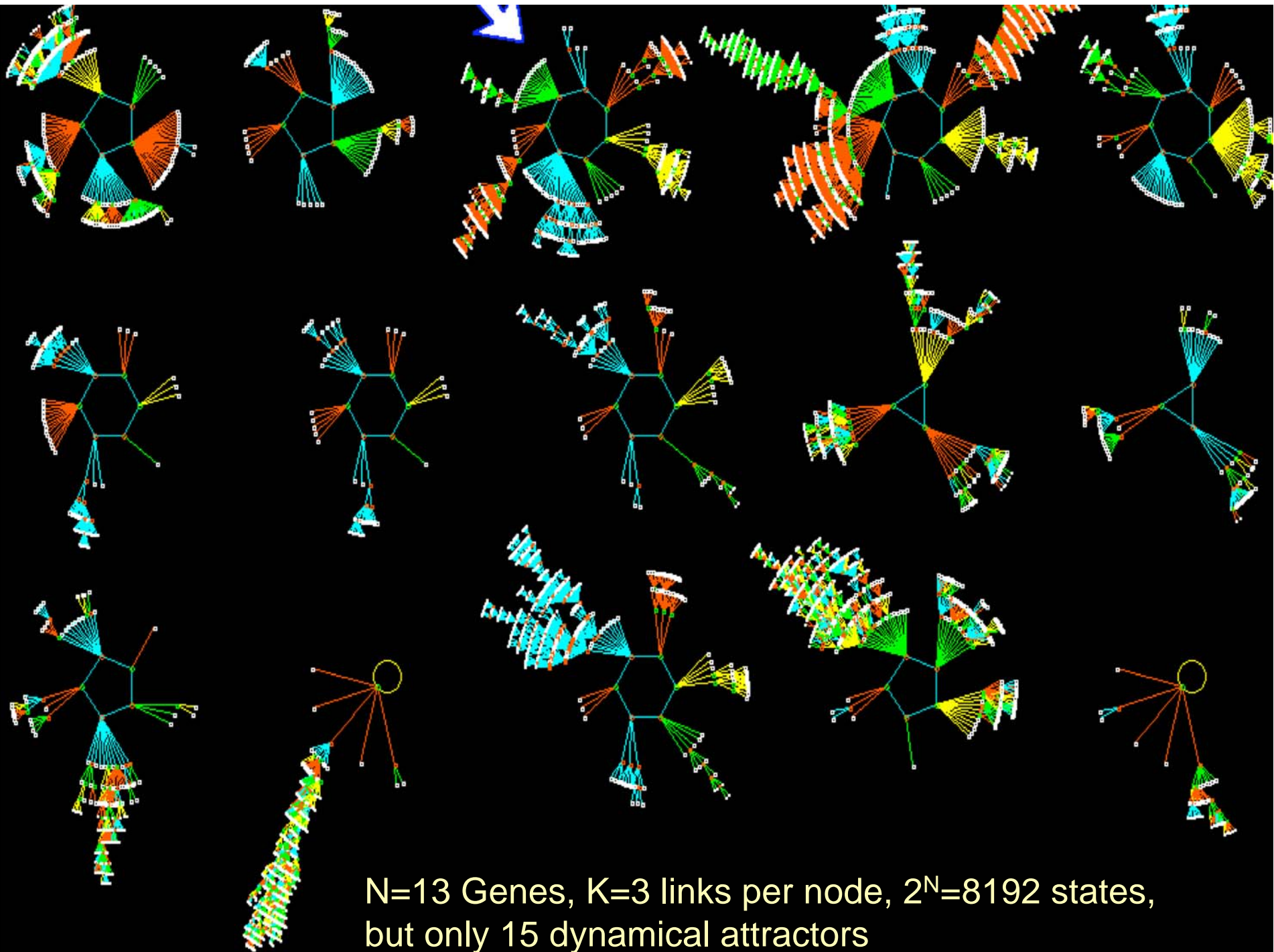
fixed points, limit cycles.

Large basins of attraction.



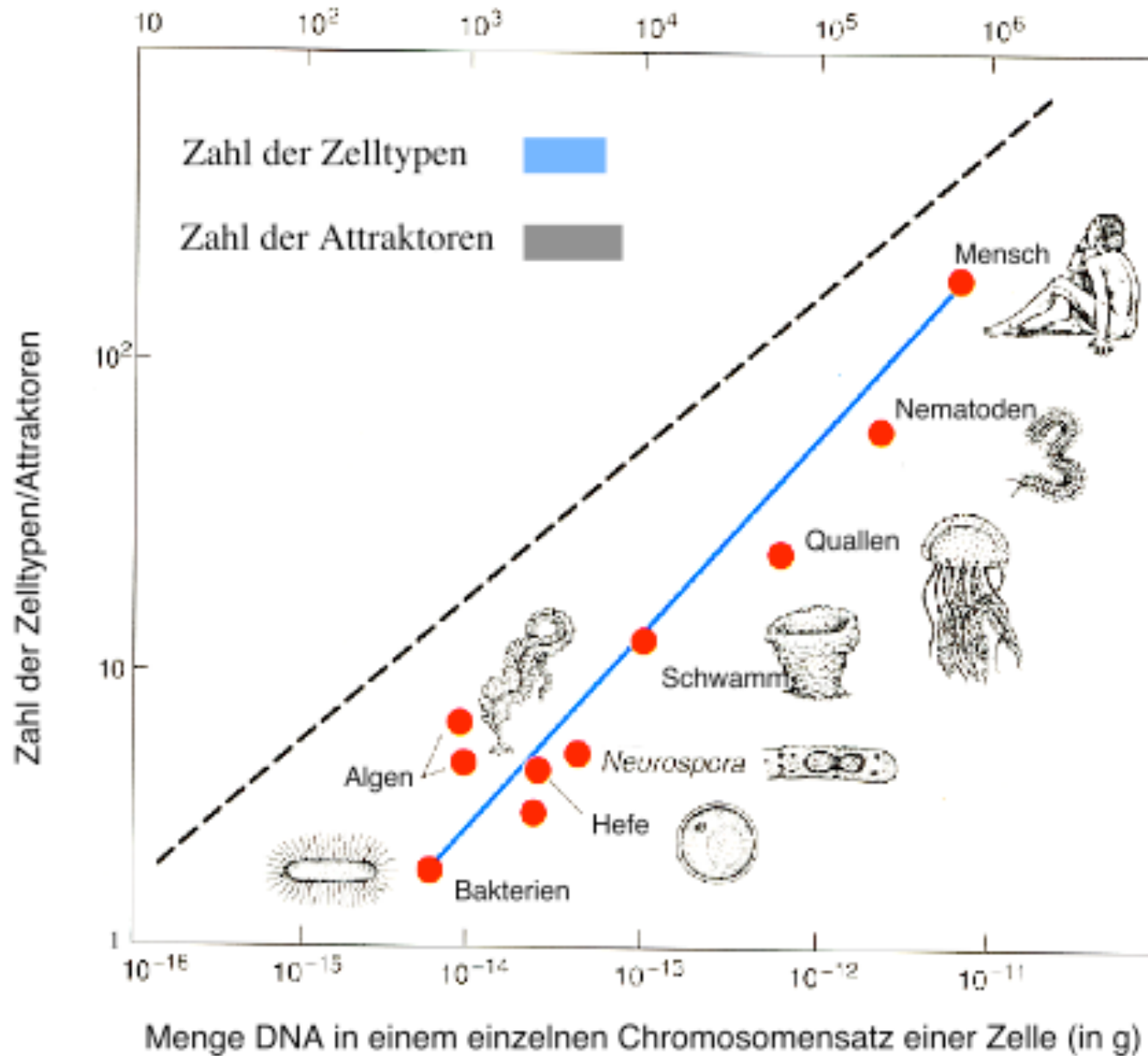
After A. Wuensche (1998)





N=13 Genes, K=3 links per node,  $2^N=8192$  states, but only 15 dynamical attractors

# Kauffman's Attractor Hypothesis (1969): „Attractors of gene networks determine cell types“



# Boolean models for regulatory networks

„more than anecdotal“ only very recently:

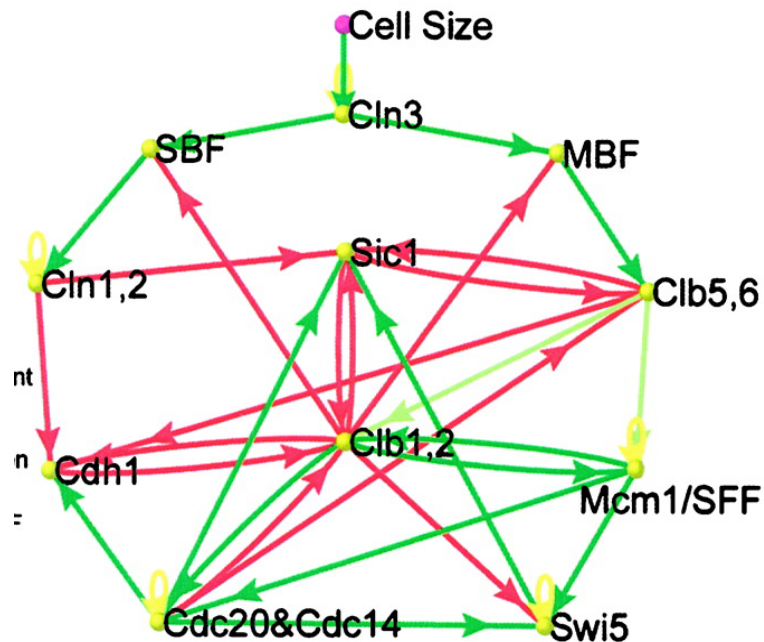
- R. Albert & H. Othmer: The topology of the regulatory interactions predicts the expression pattern of the **Drosophila segment polarity genes**, J Theor Biol 223 (2003) 1.
- F. Li, T. Long, Y. Lu, Q. Ouyang & C. Tang: The **yeast cell cycle network** is robustly designed, PNAS 101 (2004) 4781.

- Computers and the living cell
- Discrete networks as models for cellular computation
- A biological example: The yeast cell cycle
- Discrete network models and stochastic dynamics
- Applications and outlook

# Dynamical model of the yeast cell cycle

[Li et al., PNAS 2004]

- Threshold network:



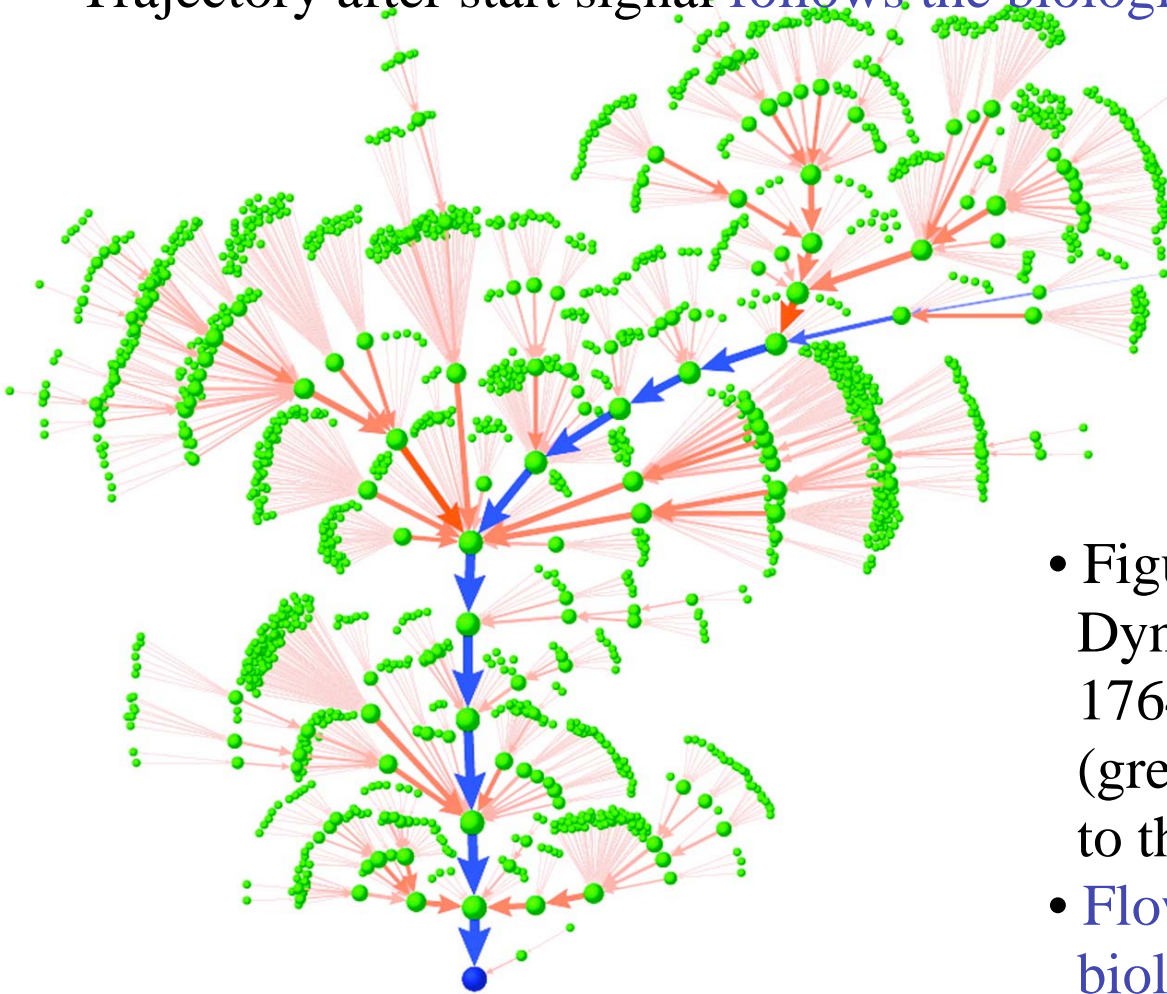
$$S_i(t + 1) = \begin{cases} 1, & \sum_j a_{ij} S_j(t) > 0 \\ 0, & \sum_j a_{ij} S_j(t) < 0 \\ S_i(t), & \sum_j a_{ij} S_j(t) = 0 \end{cases}$$

- Couplings activating/inhibitory  $a_{ij}=1/ a_{ij}=-1$
- Degradation  $S_i(t+1) = 0$  if no input for more than 1 time step
- Synchronous dynamics for all genes

# Largest attractor in state space

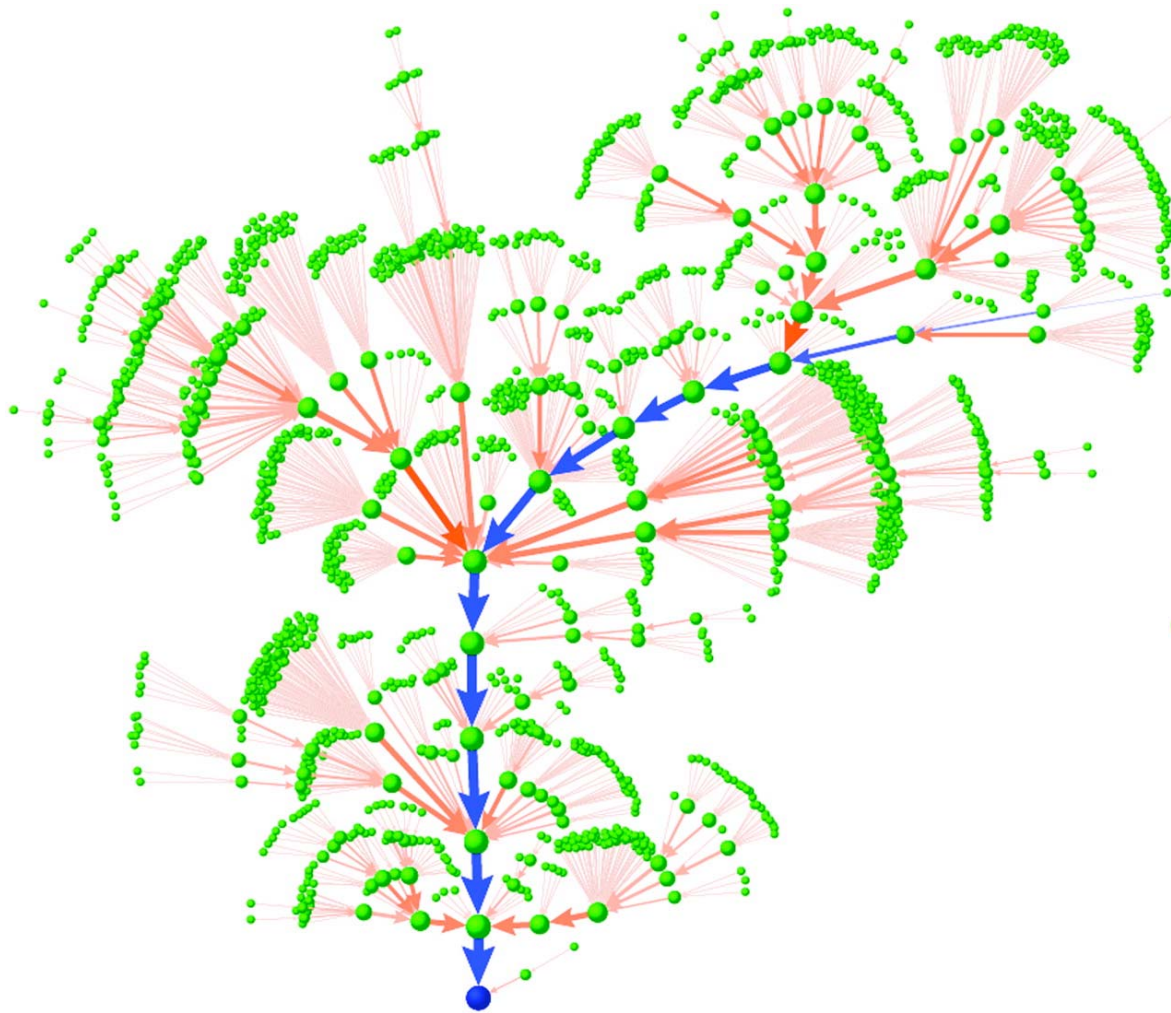
State space:  $2^{11}=2048$  states, 7 attractors (fixed points)

- Largest attractor (1764 states) = biologically stable final state
- Trajectory after start signal follows the biological time sequence

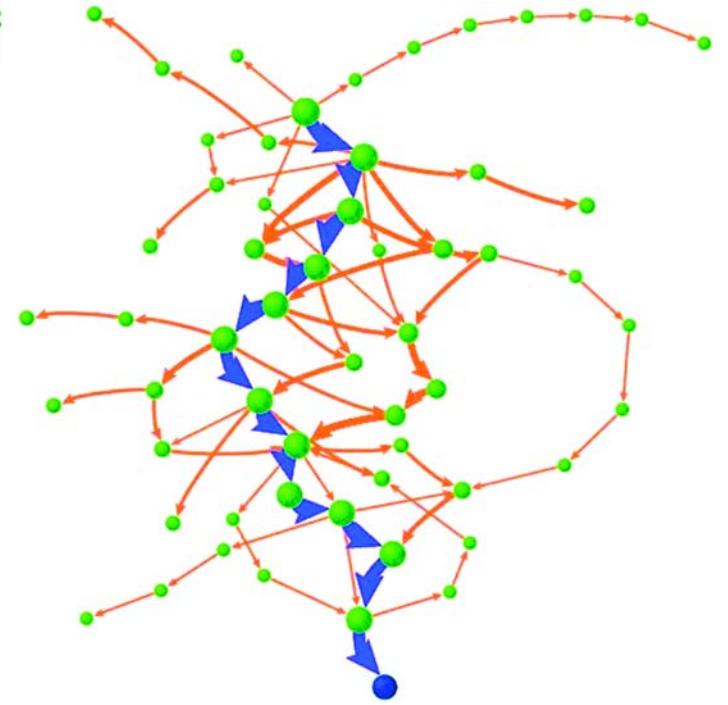


- Figure:  
Dynamical trajectories of the 1764 protein states (green nodes), that flow to the largest fixed point (G1)
- Flow converges to the biological path (blue)

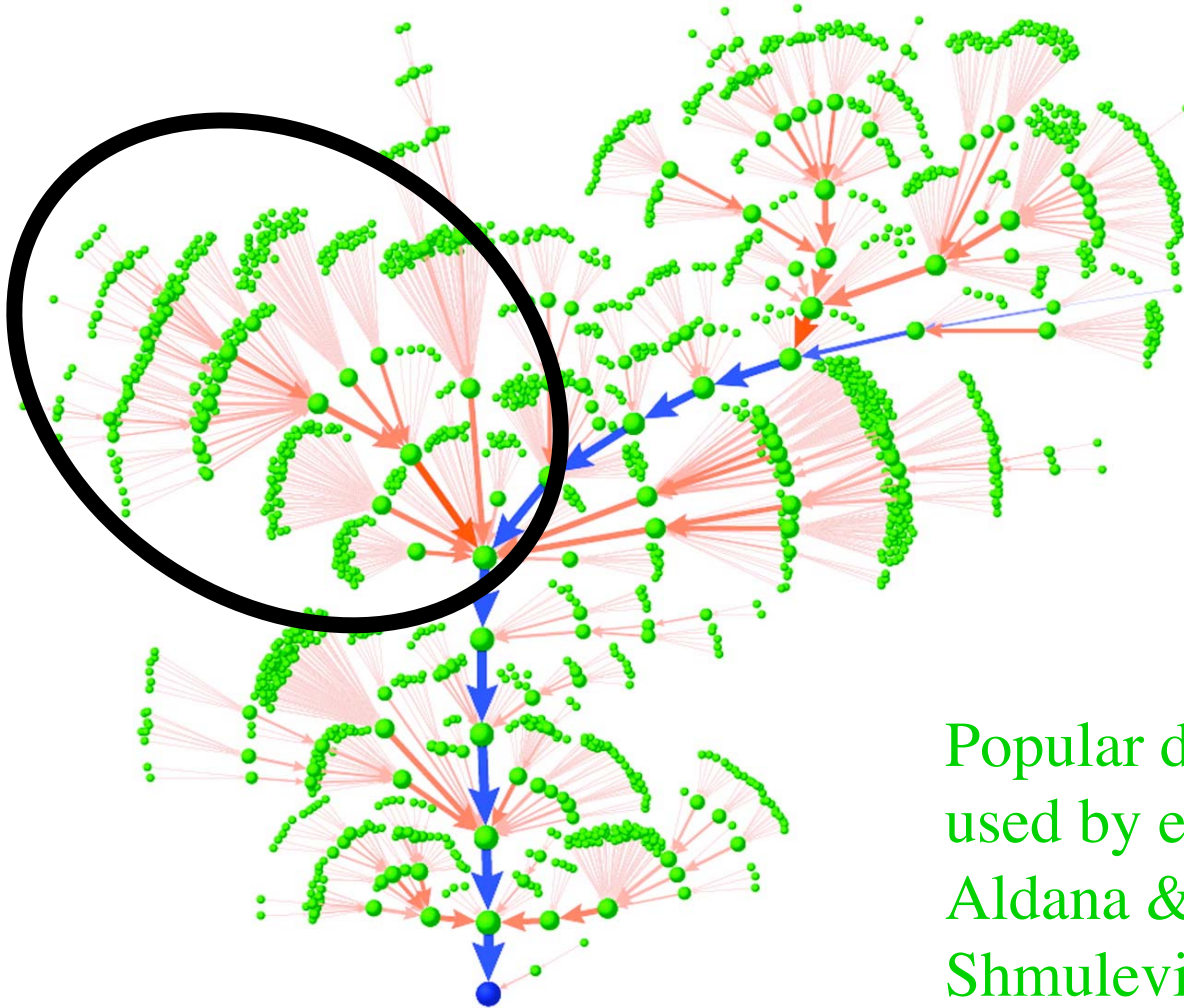
# Robustness I: Network mutations



**50% of mutations  
keep biological path:**



# Robustness II: Damage spreading after “spin flip”



Popular definition of Robustness  
used by e.g.

Aldana & Cluzel, PNAS 2003

Shmulevith et al., PNAS 2003

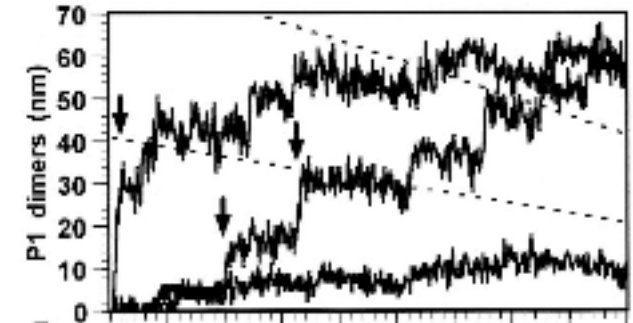
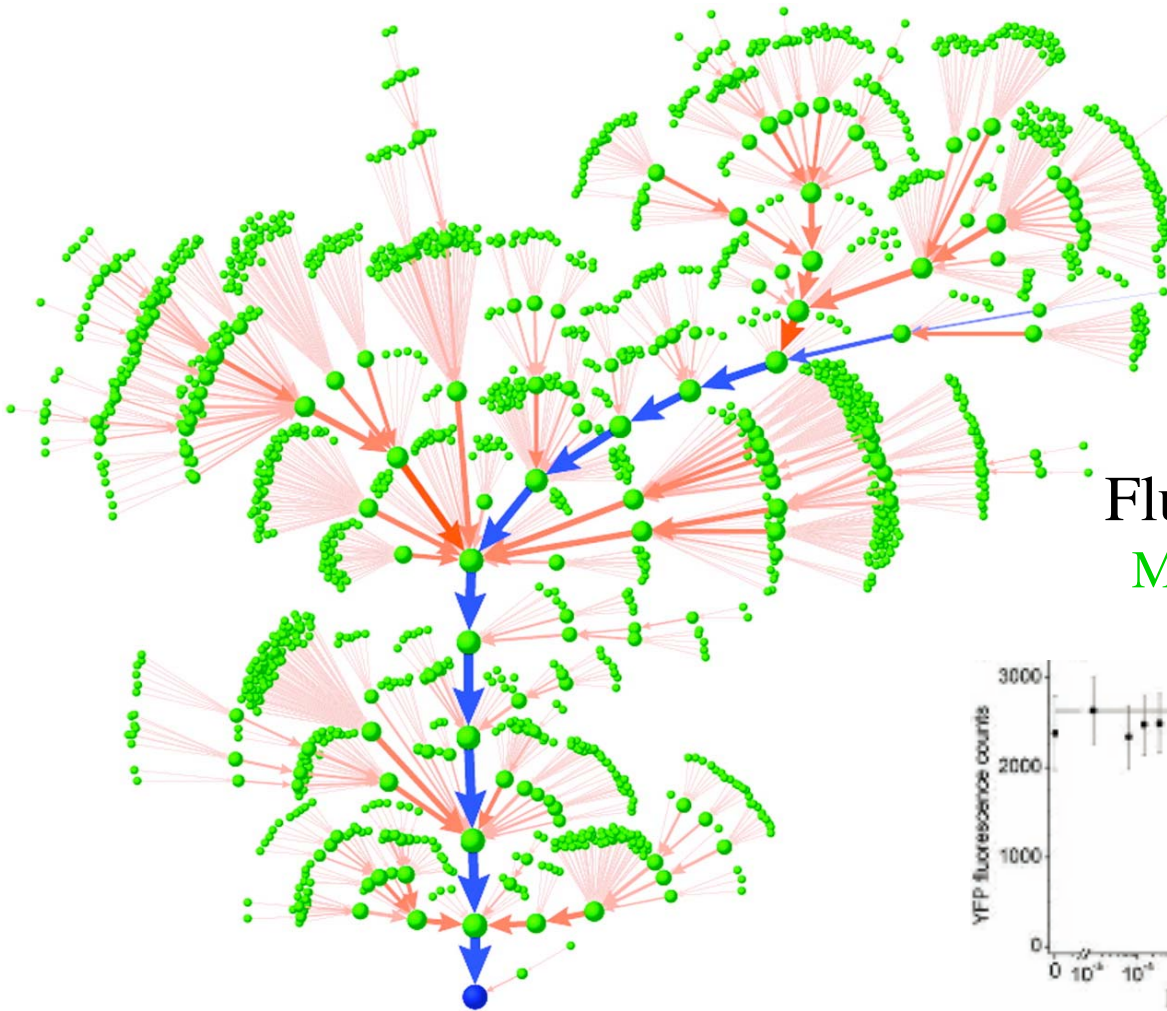
Kauffman et al., PNAS 2003

Kauffman et al., PNAS 2004

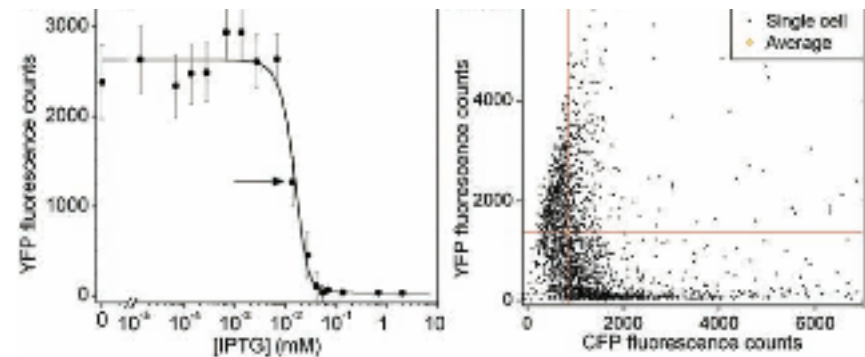


# Robustness III: Biochemical stochasticity

Genes are noisy:

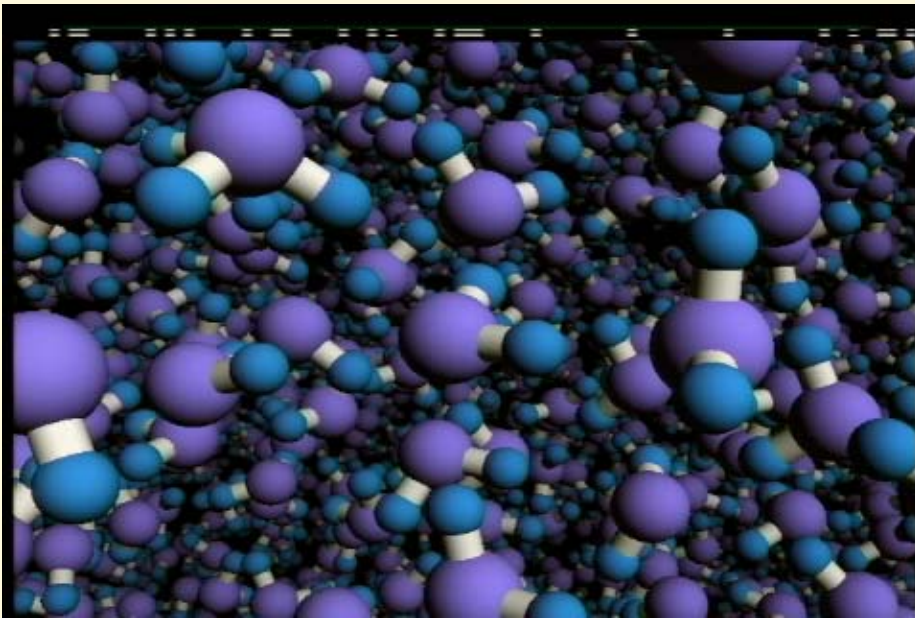


Fluctuating switching times  
McAdams & Arkin PNAS 1997



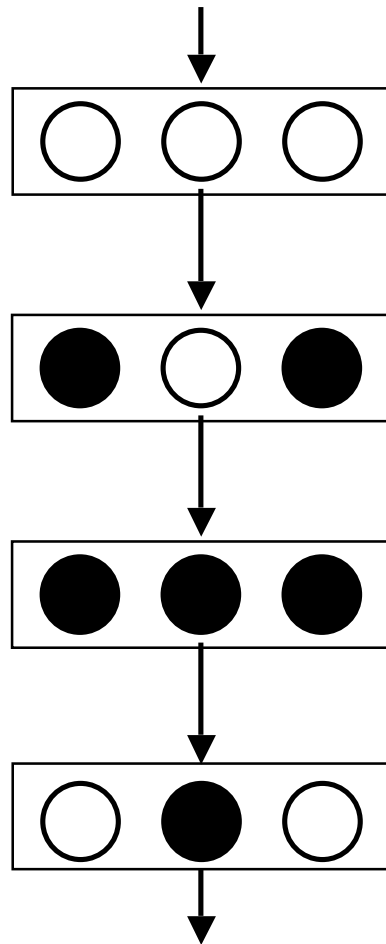
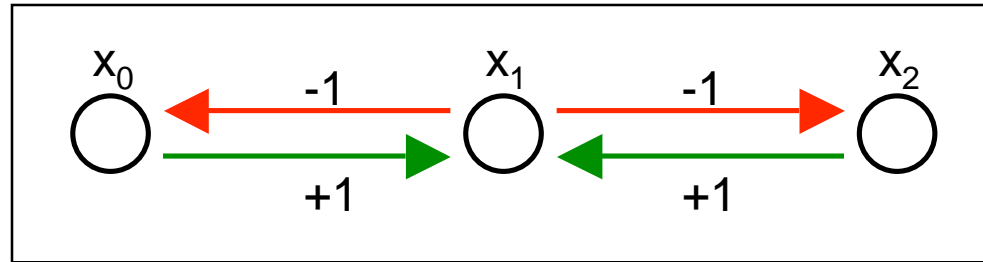
Fluctuating gene activity  
Pedraza & Oudenaarden Science 05

How does the cell achieve a clockwork-like reliability from molecular building blocks?

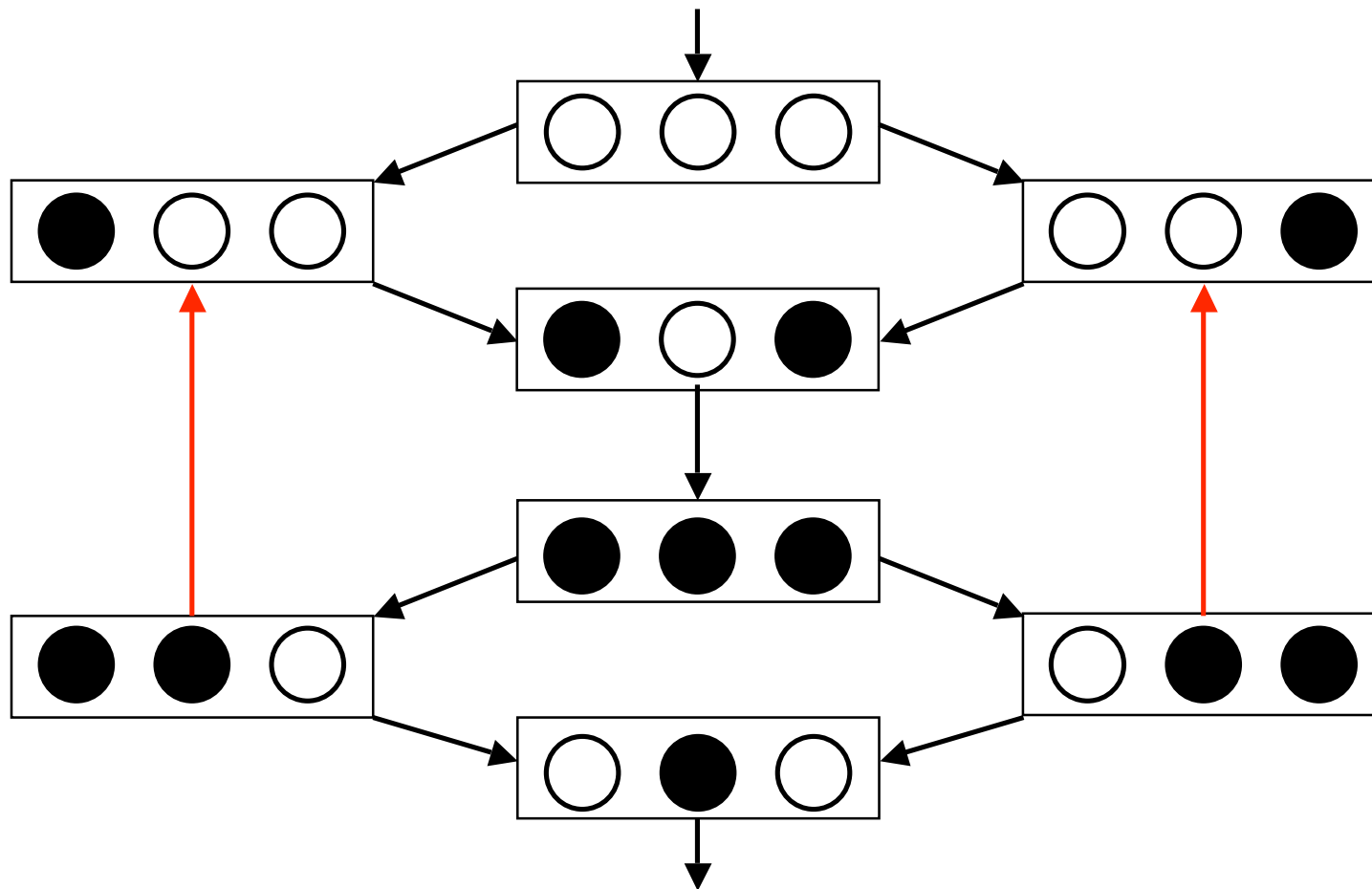
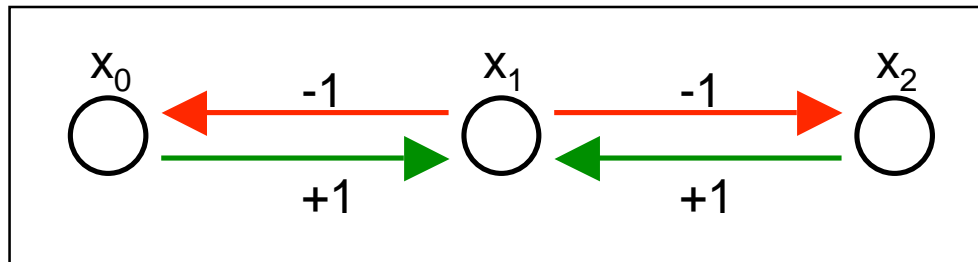


- Computers and the living cell
- **Discrete networks** as models for cellular computation
- A biological example: The **yeast cell cycle**
- **Discrete network models and stochastic dynamics**
- Applications and outlook

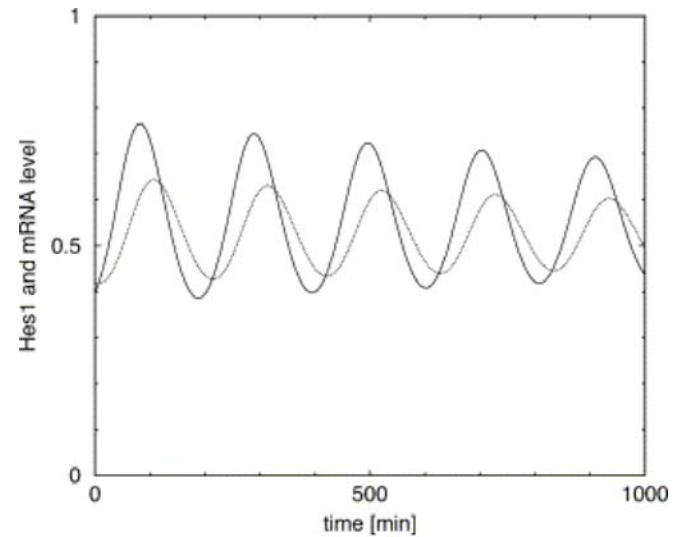
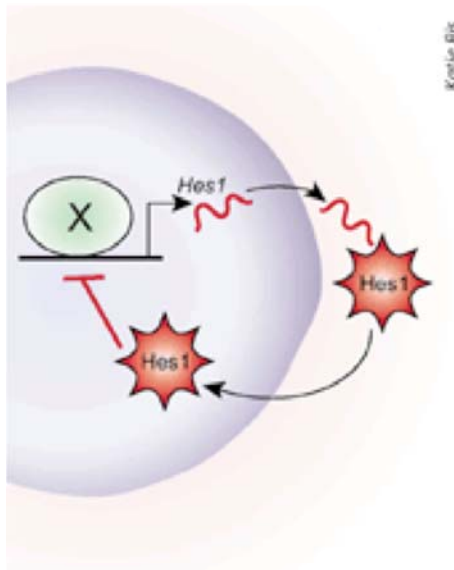
# Synchronous update (no noise)



# Asynchronous update (noisy)



# Simplest genetic circuit: Self-regulating switch Hes1



$$\frac{d[mRNA]}{dt} = \frac{\alpha k^h}{k^h + [Hes1(t - \tau)]} - \frac{[mRNA(t)]}{\tau_{RNA}}$$

$$\frac{d[Hes1]}{dt} = \beta[mRNA(t)] - \frac{[Hes1(t)]}{\tau_{Hes1}}$$

[M.H. Jensen, K. Sneppen, G. Tiana 2003]

## Simplest model that keeps timing info

- Keep **delay** and one low pass **filter**,  
difference equation for RNA concentration  $c$ :

$$\Delta c_i = \alpha [f(S_i(t - \tau)) - c_i(t)] \Delta t$$

- Let threshold sum now drive the concentration gradient:

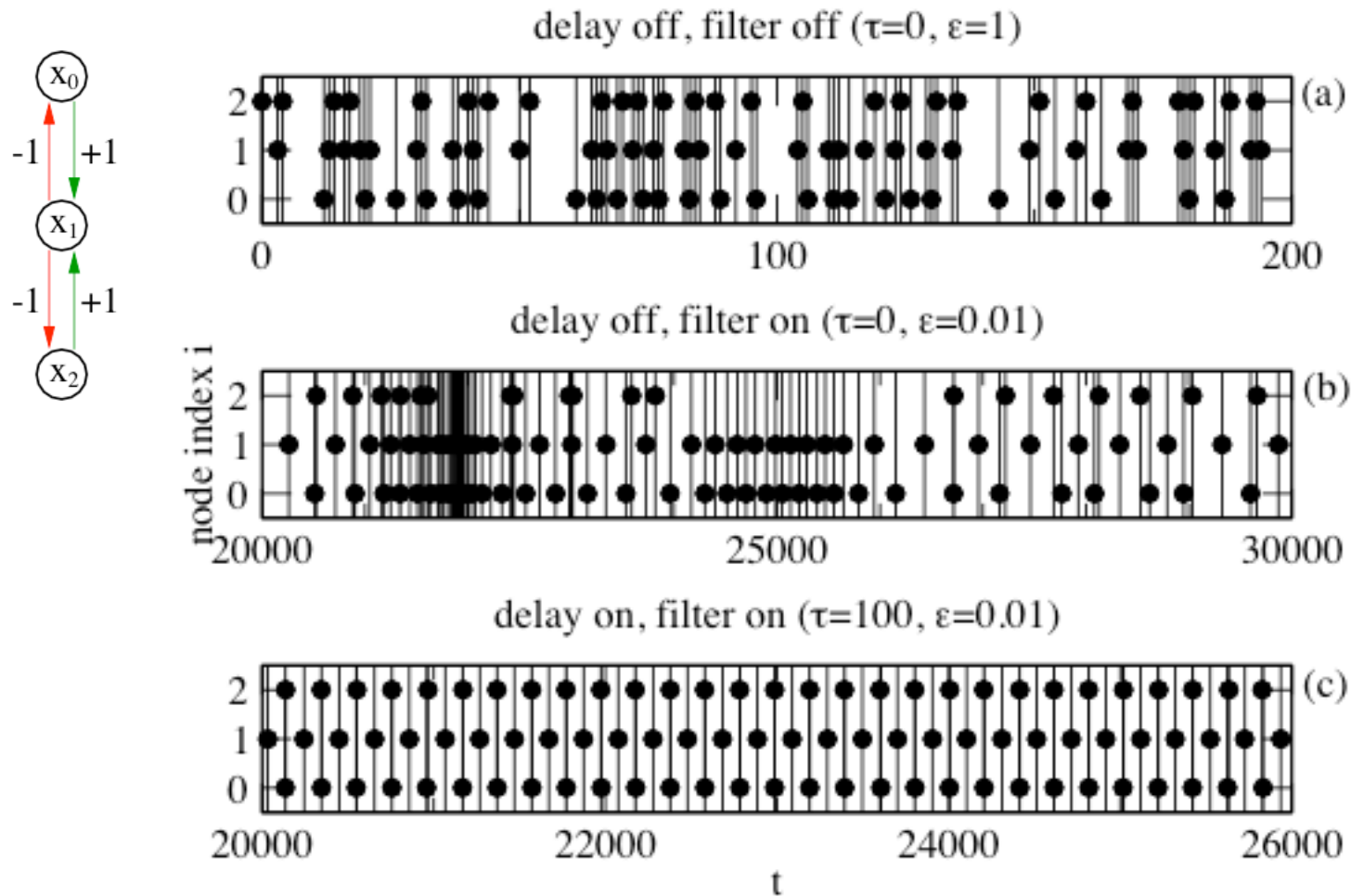
$$\Delta c_i = \varepsilon \cdot \text{sgn} \left( \sum_j J_{ij} S_j(t - \tau) - \vartheta_i \right)$$

$$\varepsilon = \alpha \Delta t \quad S_j = \theta(c_j - 1/2) \quad c_i \in [0,1]$$

- Stochastic numerical integration: **random sequential update**

[K. Klemm & S.B., q-bio/0309013]

# Switching pattern of nodes with filter and delay

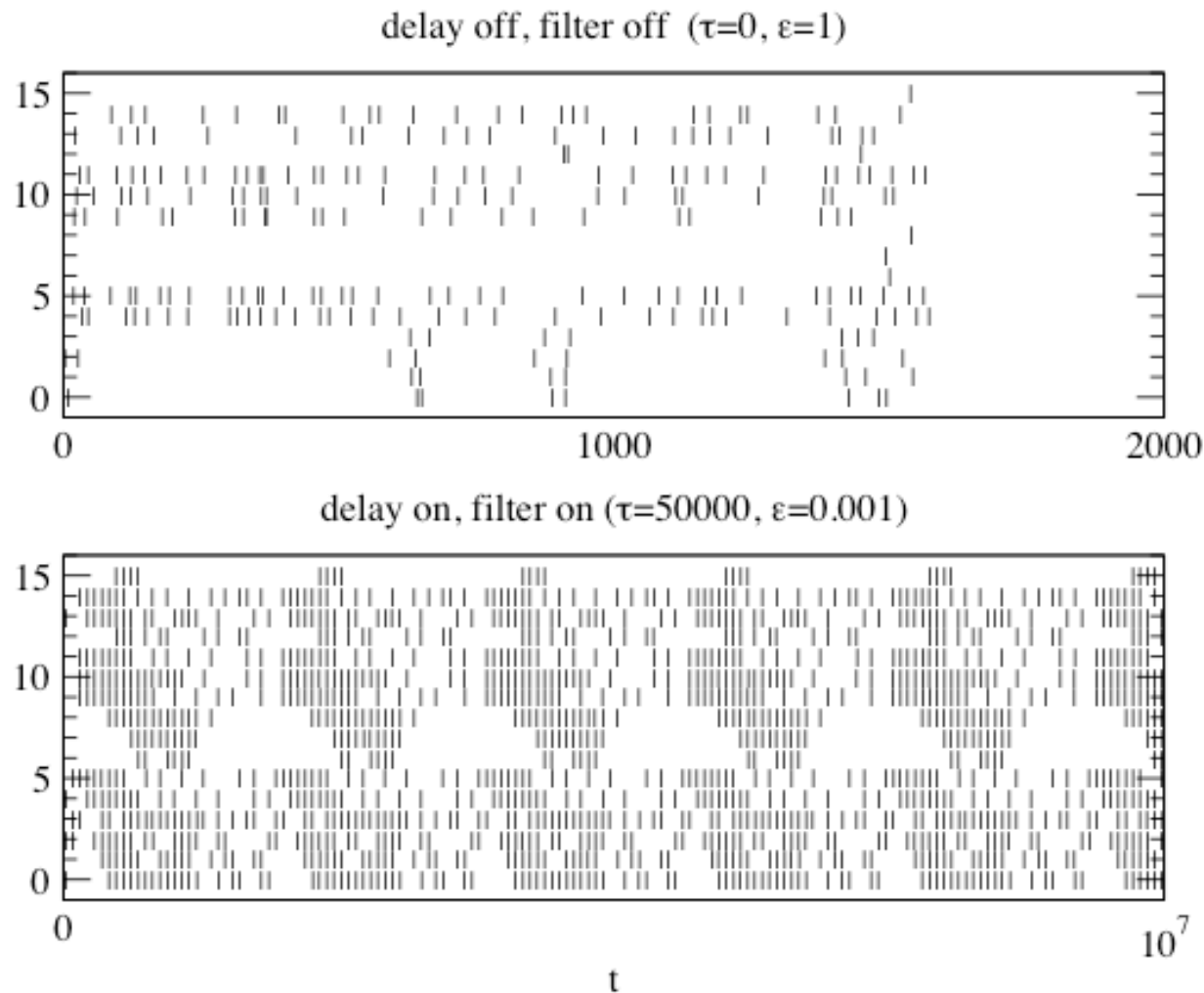


- Stable **quasi-deterministic dynamics** under **asynchronous** (random single spin) updates!



# Dynamics on a random network

$N=16$ ,  $K=3$ , random single spin updates



Parallel-update dynamics recovered under asynchronous dynamics!

[K. Klemm & S.B., q-bio/0309013]

# The yeast cell cycle network revisited: How does it deal with noise?

Multi-switch events can in principle de-synchronize:

Time	Cln3	MBF	SBF	Cln1,2	Cdh1	Swi5	Cdc20/ Cdc14	Clb5,6	Sic1	Clb1,2	Mcm1/ SFF	Phase
<b>1</b>	<b>1</b>	<b>0</b>	0	0	1	0	0	0	1	0	0	Start
<b>2</b>	0	<b>1</b>	<b>1</b>	0	1	0	0	0	1	0	0	G <sub>1</sub>
<b>3</b>	0	1	1	<b>1</b>	1	0	0	0	1	0	0	G <sub>1</sub>
<b>4</b>	0	1	1	1	<b>0</b>	0	0	0	<b>0</b>	0	0	G <sub>1</sub>
<b>5</b>	0	1	1	1	0	0	0	<b>1</b>	0	0	0	S
<b>6</b>	0	1	1	1	0	0	0	1	0	<b>1</b>	<b>1</b>	G <sub>2</sub>
<b>7</b>	0	<b>0</b>	<b>0</b>	1	0	0	<b>1</b>	1	0	1	1	M
<b>8</b>	0	0	0	<b>0</b>	0	<b>1</b>	1	<b>0</b>	0	1	1	M
<b>9</b>	0	0	0	0	0	1	1	0	<b>1</b>	1	1	M
<b>10</b>	0	0	0	0	0	1	1	0	1	<b>0</b>	1	M
<b>11</b>	0	0	0	0	<b>1</b>	1	1	0	1	0	<b>0</b>	M
<b>12</b>	0	0	0	0	1	1	<b>0</b>	0	1	0	0	G <sub>1</sub>
<b>13</b>	0	0	0	0	1	<b>0</b>	0	0	1	0	0	G <sub>1</sub>

[asynchronous modeling strategy e.g: M Chaves,  
R Albert, ED Sontag, J Theor Biol 235 (2005) 431]

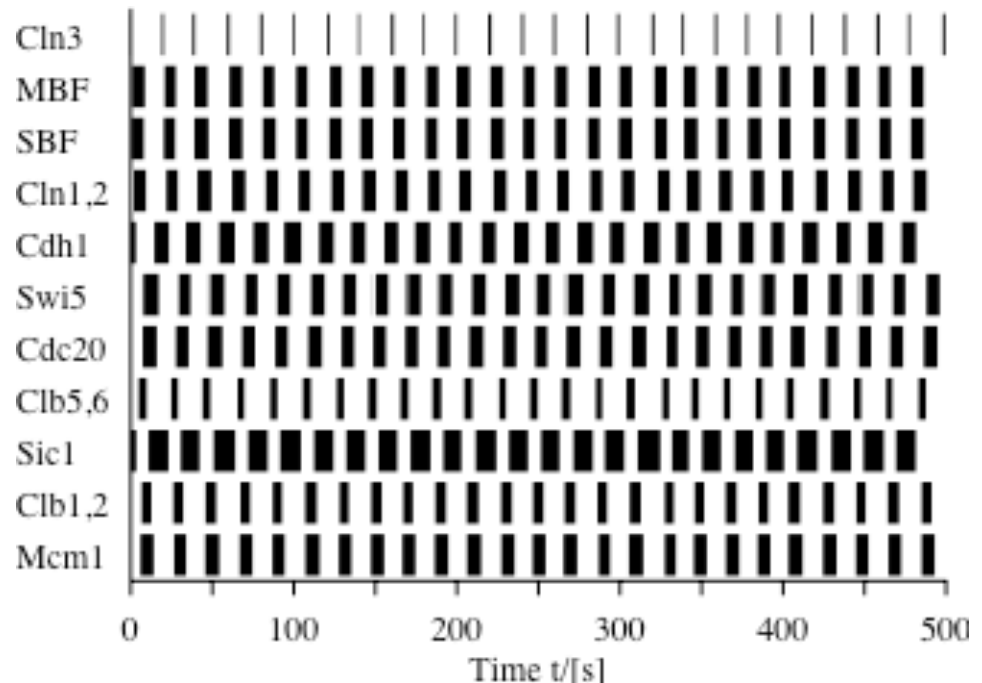
# A stochastic model of the yeast cell cycle network

Delay differential equation:  $\frac{dc_i(t)}{dt} = f_i(t, t_d) - \frac{c_i(t)}{\tau}$

$$f_i(t, t_d) = \begin{cases} 1, & \sum_j a_{ij} S_j(t - t_d) > 0 \\ 0, & \sum_j a_{ij} S_j(t - t_d) < 0 \end{cases}$$

$$S_i(t) = \begin{cases} 1, & c_i(t) > 0.5 \\ 0, & c_i(t) < 0.5 \end{cases}$$

Fluctuating delay  $t_d$ :



Order of switching events stable against timing fluctuations ---> Yeast cell cycle is stable

[S. Braunewell & S. Bornholdt, J. Theor. Biol. 245 (2007) 638]  
 [other stochastic yeast model: Zhang et al., Physica D 219 (2006) 35]

# Are we just lucky?

In fact most attractors in discrete dynamical networks are **artifacts of the synchronous update mode** (!) and disappear in the presence of noise...

[F. Greil & B. Drossel, Phys. Rev. Lett. 95 (2005) 048701; K. Klemm & S. Bornholdt, Phys. Rev. E 72 (2005) 055101(R)]

## **Independence of temporal order of flips requires:**

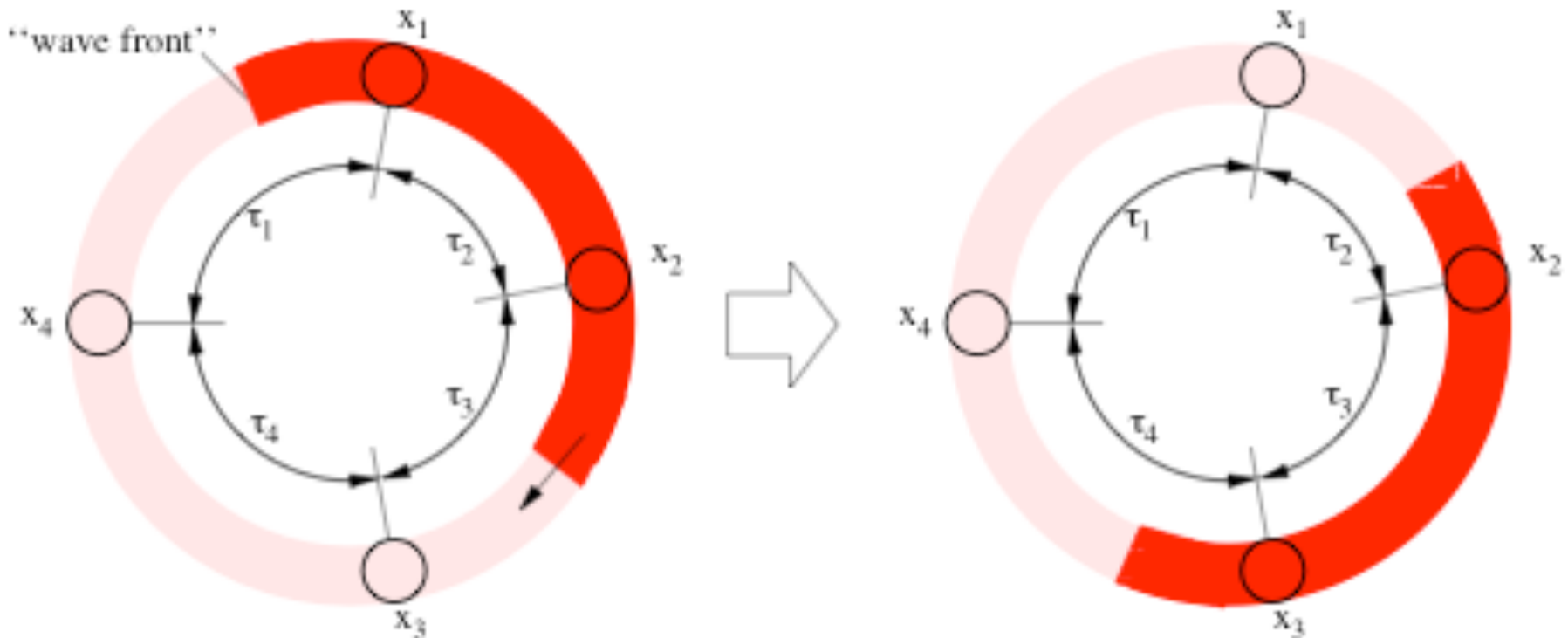
1. Nodes must have stable (non-fluctuating) input when they are required to flip.
2. Nodes that flip in the same macro time step must not influence each other.

**...these are the same rules as in electrical engineering!**

[K. Klemm & S. Bornholdt, PNAS 102 (2005) 18414]

# Requirement for robustness against noise poses constraints on network topology

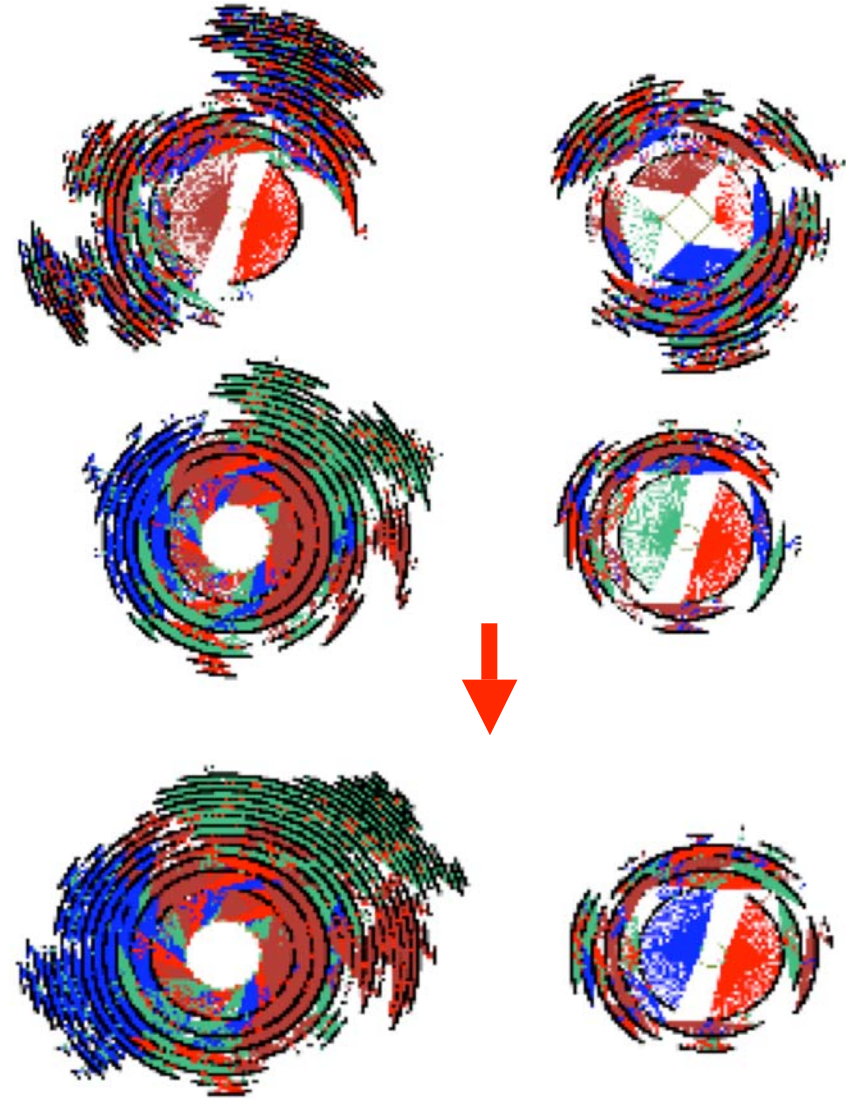
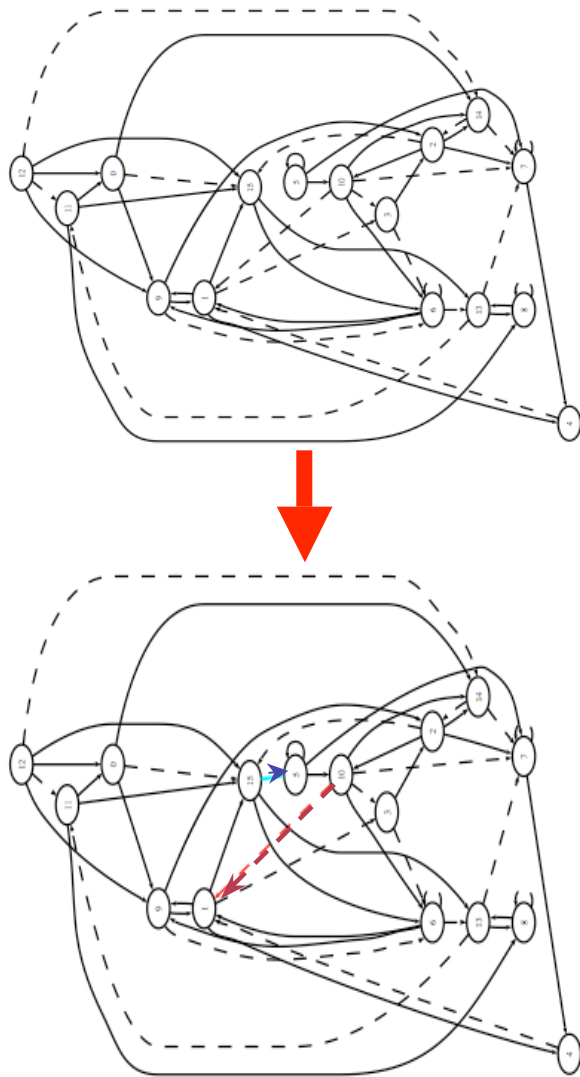
## Why the 4-node Repressilator is unstable....



Phase not conserved for loops with even # of inhibitory links

[K. Klemm & S. Bornholdt, PNAS 102 (2005) 18414]

# Reliable networks are evolvable

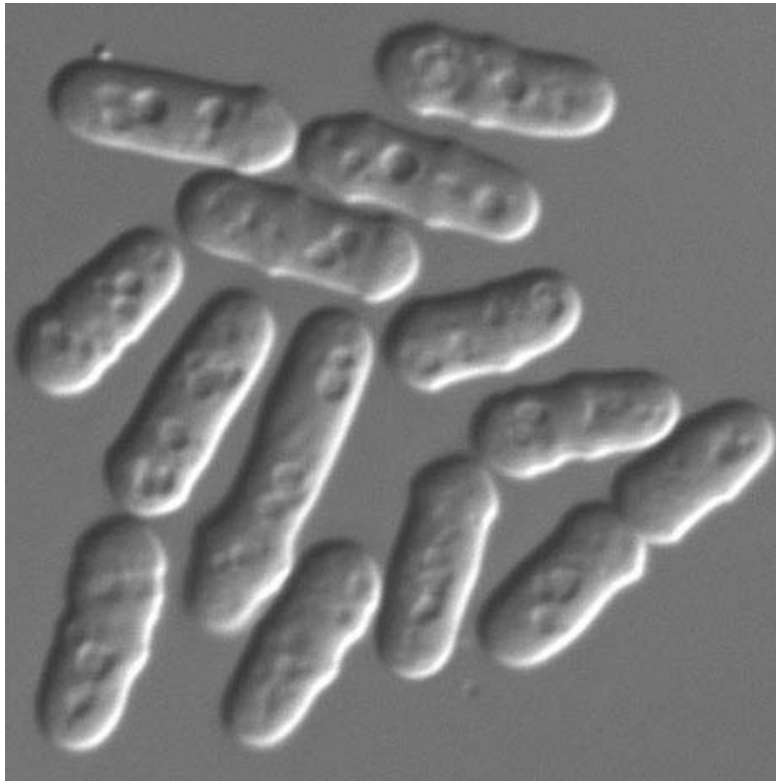


Single Mutations can stabilize a given attractor!

[S. Braunewell & S. Bornholdt (2007) [arxiv.org/abs/0707.1407](https://arxiv.org/abs/0707.1407)]

- Computers and the living cell
- **Discrete networks** as models for cellular computation
- A biological example: The **yeast cell cycle**
- Discrete network models and **stochastic dynamics**
- **Applications and outlook**

# Case study: Cell cycle control of fission yeast



Existing model: Differential equations with 40+ free parameters

$$\frac{d[\text{Cdc13}_7]}{dt} = k_1 M - (k_2' + k_2''[\text{Ste9}] + k_2'''[\text{Slp1}])([\text{Cdc13}_7]), \quad (1)$$

$$\begin{aligned} \frac{d[\text{preMPF}]}{dt} = & k_{\text{act}}([\text{Cdc13}_7] - [\text{preMPF}]) - k_{25}[\text{preMPF}] - (k_2' \\ & + k_2''[\text{Ste9}] + k_2'''[\text{Slp1}])([\text{preMPF}], \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{d[\text{Ste9}]}{dt} = & (k_3' + k_3''[\text{Slp1}]) \frac{1 - [\text{Ste9}]}{J_3 + 1 - [\text{Ste9}]} - (k_4'[\text{SK}] \\ & + k_4''[\text{MPF}]) \frac{[\text{Ste9}]}{J_4 + [\text{Ste9}]}, \end{aligned} \quad (3)$$

$$\frac{d[\text{Slp1}_7]}{dt} = k_5' + k_5'' \frac{[\text{MPF}]^2}{J_5' + [\text{MPF}]^2} - k_6[\text{Slp1}_7], \quad (4)$$

$$\begin{aligned} \frac{d[\text{Slp1}]}{dt} = & k_7[\text{IEP}] \frac{[\text{Slp1}_7] - [\text{Slp1}]}{J_7 + [\text{Slp1}_7] - [\text{Slp1}]} \\ & - k_8 \frac{[\text{Slp1}]}{J_8 + [\text{Slp1}]} - k_9[\text{Slp1}], \end{aligned} \quad (5)$$

$$\frac{d[\text{IEP}]}{dt} = k_9[\text{MPF}] \frac{1 - [\text{IEP}]}{J_9 + 1 - [\text{IEP}]} - k_{10} \frac{[\text{IEP}]}{J_{10} + [\text{IEP}]}, \quad (6)$$

$$\frac{d[\text{Rum1}_7]}{dt} = k_{11} - (k_{12} + k_{12}'[\text{SK}] + k_{12}''[\text{MPF}])([\text{Rum1}_7]), \quad (7)$$

$$\frac{d[\text{SK}]}{dt} = k_{13}[\text{TF}] - k_{14}[\text{SK}], \quad (8)$$

$$\frac{dM}{dt} = \mu M, \quad (9)$$

$$[\text{Trimer}] = \frac{2[\text{Cdc13}_7][\text{Rum1}_7]}{\Sigma + \sqrt{\Sigma^2 - 4[\text{Cdc13}_7][\text{Rum1}_7]}}, \quad (10)$$

$$[\text{MPF}] = \frac{([\text{Cdc13}_7] - [\text{preMPF}])([\text{Cdc13}_7] - [\text{Trimer}])}{[\text{Cdc13}_7]}, \quad (11)$$

$$[\text{TF}] = G(k_{15} M, k_{15}' + k_{15}''[\text{MPF}], J_{15}, J_{15}), \quad (12)$$

where

$$k_{\text{act}} = k_{\text{act}}' + (k_{\text{act}}'' - k_{\text{act}}') G(V_{\text{act}}, V_{\text{act}}[\text{MPF}], J_{\text{act}}, J_{\text{act}}),$$

$$k_{25} = k_{25}' + (k_{25}'' - k_{25}') G(V_{25}[\text{MPF}], V_{25}, J_{25}, J_{25}),$$

$$\Sigma = [\text{Cdc13}_7] + [\text{Rum1}_7] + K_{\text{act}},$$

$$G(a, b, c, d) = \frac{2ad}{b - a + bc + ad + \sqrt{(b - a + bc + ad)^2 - 4ad(b - a)}}$$

[B Novak et al., Chaos 11 (2001) 277]

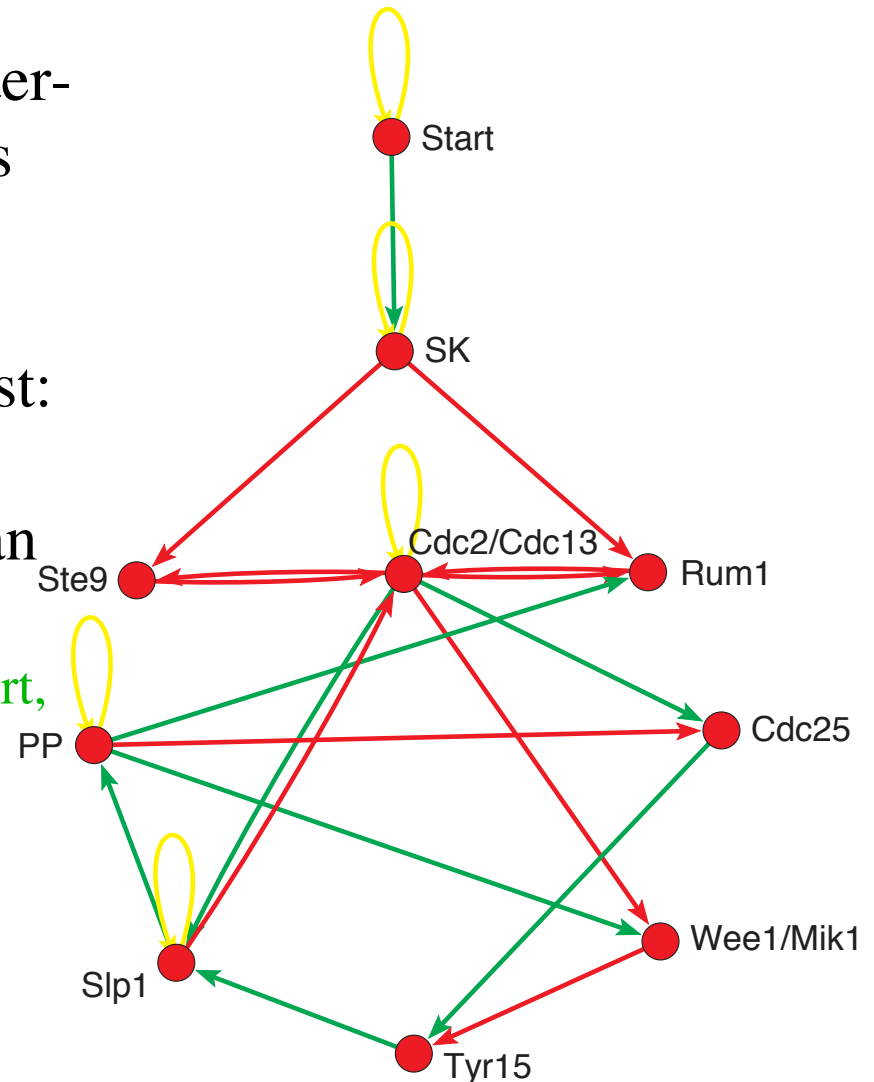


# Case study: Cell cycle control of fission yeast

- If a biological network is parameter-insensitive, can we drop parameters from the model altogether?

- Much different from budding yeast: This is largely a protein interaction network (non-transcriptional) --- can we model this the Boolean way?

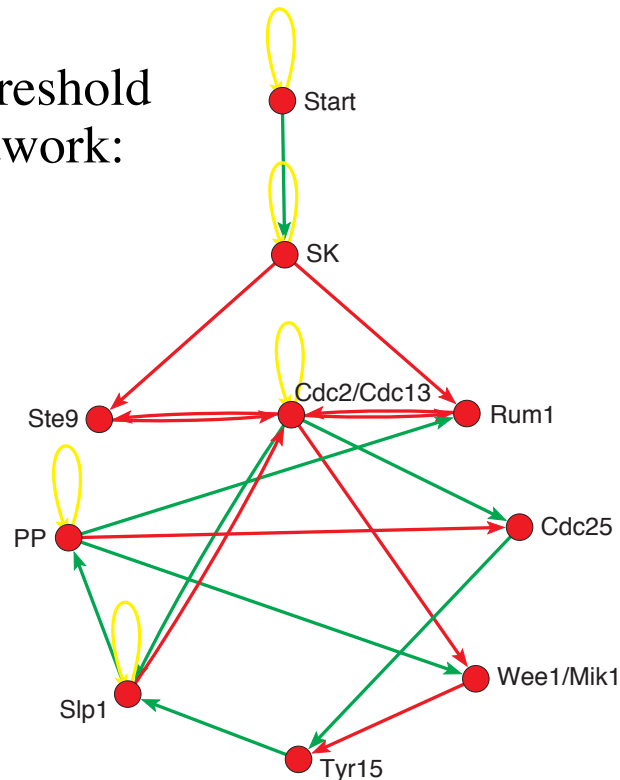
[this worked earlier: Li, Assmann, & Albert, PLoS Biology e312 (2006)]



[M. Davidich & S. Bornholdt (2007) [arxiv.org/abs/0704.2200](http://arxiv.org/abs/0704.2200)]

# Constructing a dynamical model of the fission yeast cell cycle

- Threshold network:



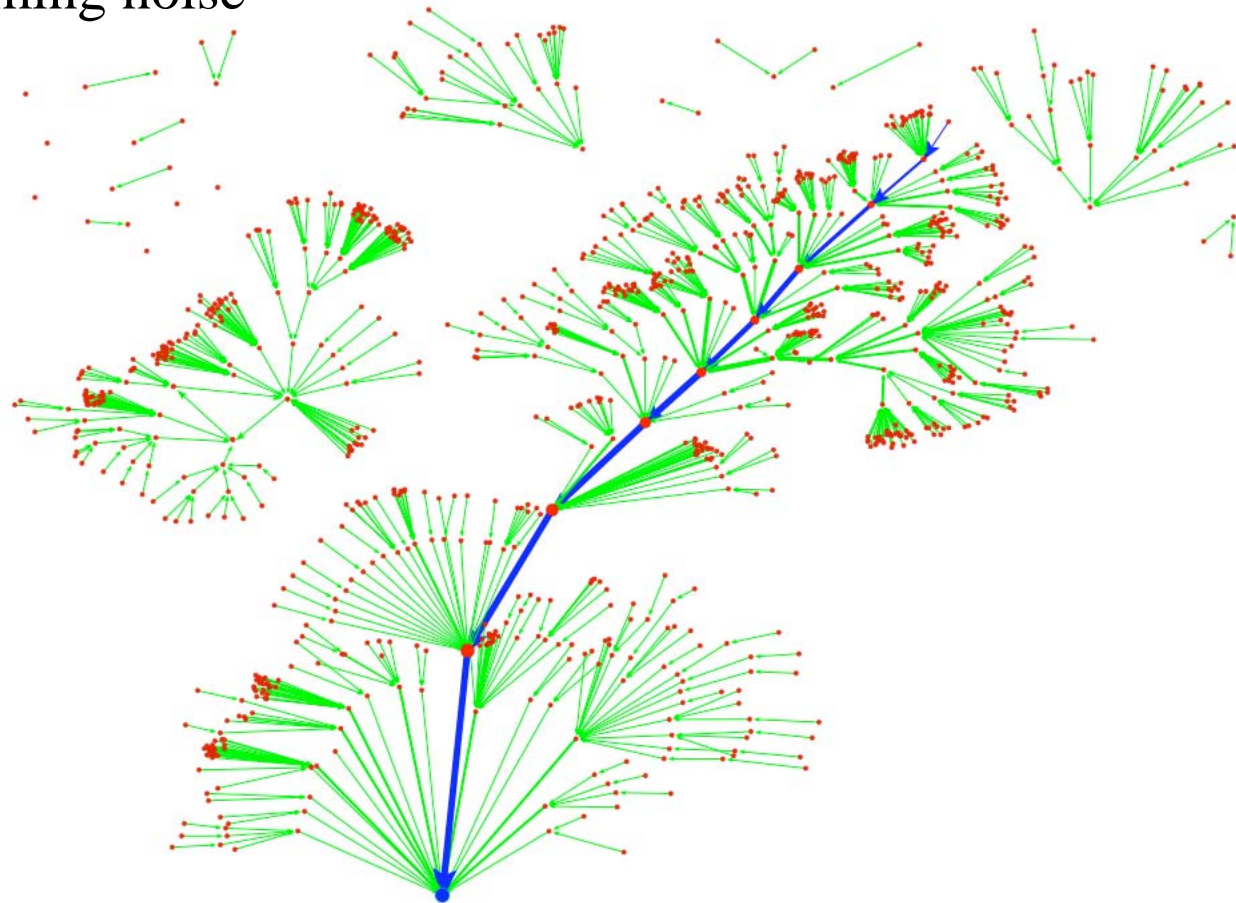
$$S_i(t + 1) = \begin{cases} 1, & \sum_j a_{ij} S_j(t) > 0 \\ 0, & \sum_j a_{ij} S_j(t) < 0 \\ S_i(t), & \sum_j a_{ij} S_j(t) = 0 \end{cases}$$

- Couplings activating/inhibitory  $a_{ij}=1/ a_{ij}=-1$
- Degradation  $S_i(t+1) = 0$  if no input for more than 1 time step
- Synchronous dynamics for all nodes (genes/proteins)

[M. Davidich & S. Bornholdt (2007) [arxiv.org/abs/0704.2200](https://arxiv.org/abs/0704.2200)]

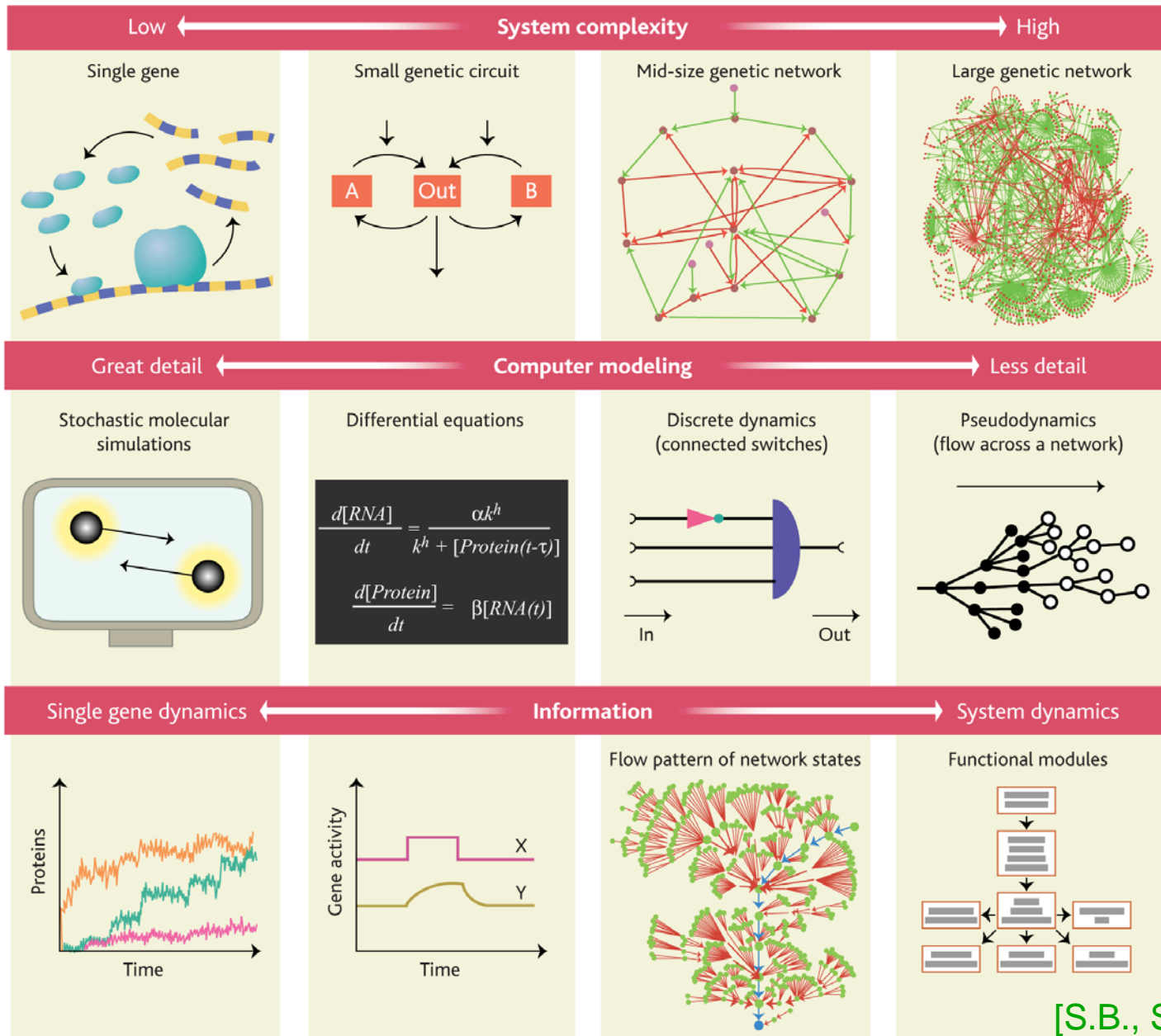
# Fission yeast state space: Attractor landscape

- State space:  $2^{10}=1024$  states, 18 attractors (fixed points)
- Largest attractor (722 states) = biologically stable final state
- Trajectory after start signal follows the biological time sequence
- insensitive to timing noise



[M. Davidich & S.Bornholdt (2007) [arxiv.org/abs/0704.2200](https://arxiv.org/abs/0704.2200)]

# Models of cellular networks



[S.B., Science 2005]

# Summary

- **Some computational tasks of the cell have digital character: Sequence control**
- **Dynamical networks can serve as „computers“ that generate reliable dynamics from unreliable elements**
- **Yeast cell cycle network models exhibit reliability under stochastic dynamics**
- **Requiring robustness against stochasticity has implications for network topology, but reliable networks are evolvable**

[www.itp.uni-bremen.de/complex](http://www.itp.uni-bremen.de/complex)

# Collaborators

**Konstantin Klemm,  
University of Leipzig,  
Germany**



**Maria Davidich,  
University of Bremen**



**Stefan Braunewell  
University of Bremen**



[www.itp.uni-bremen.de/complex](http://www.itp.uni-bremen.de/complex)