

IV121: Computer science applications in biology

Qualitative Models in Biology

David Šafránek

February 25, 2013

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.



What is discrete? What does discrete mean?

- think of you communicating with your friends today
 - how many interactions have you noticed?
 - each one caused you a **discrete event**
- think of flies flying in the room
 - what interactions they have with each other?
 - what interactions they have with the room walls?
 - each one is a **discrete event**
- think of molecules in a solution or in a cell ...

Is there something what is not discrete?

- think of movement around you
- think of you moving
- think of time passing
- think of current in the power supply
 - electron flow through or around the wire material
 - *is it discrete or continuous?*

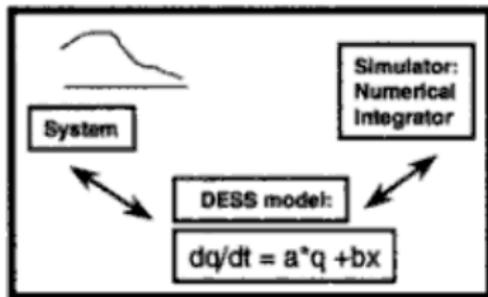
A way to understand the world...

- to understand (and capture) things happening around us we developed a modeling framework
 - discrete-time dynamics models
 - music “modeled” on CD
 - video clips filmed by your camera
 - ...
 - continuous-time dynamics models
 - mathematical model describing the flow of electrons in an electrical circuit
 - models of chemical reaction dynamics
 - ...
 - discrete-event models
 - model of a coffee machine
 - model of an elevator
 - models of chemical reaction dynamics
 - mixed models – e.g., hybrid systems ...

A way to understand the world...

- to understand (and capture) things happening around us we developed a modeling framework
 - discrete-time dynamics models
 - music “modeled” on CD
 - video clips filmed by your camera
 - ...
 - continuous-time dynamics models
 - mathematical model describing the flow of electrons in an electrical circuit
 - **models of chemical reaction dynamics**
 - ...
 - discrete-event models
 - model of a coffee machine
 - model of an elevator
 - **models of chemical reaction dynamics**
 - mixed models – e.g., hybrid systems ...

A way to understand the world...

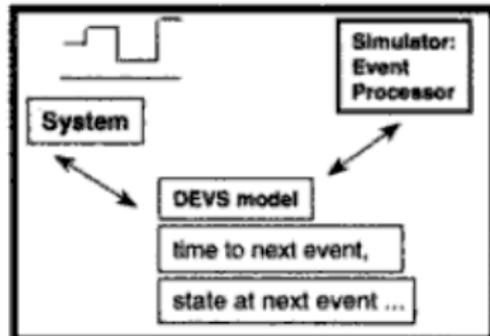
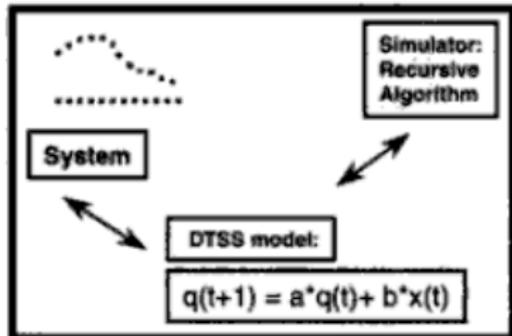


System Specification Formalism:

DESS: differential equation

DTSS: difference equation

DEVS: discrete event



A way to understand the world...

Time scales

- think of a lightning causing a fire
 - very fast electron flow → long-lasting fire
 - one of the views: a discrete event (lightning) changing the continuous world
 - is this a reasonable simplification?
- when modeling the world we have to abstract
- our abstract views (models) can make a hierarchy
- a drawback of simplifications: there can be feedbacks we silently omit ...

Time scales in a cell

Quantitative parameters	Values in E.Coli
cell size	$1\mu m^3$
number of protein molecules in a cell	$4 \cdot 10^6$
size of a protein molecule	$5nm$
concentration of a particular protein in a cell	$1nM$
amount of proteins in a cell	18%
time of protein diffusion	0, 1s
time of other molecules diffusion	1msec
number of genes	4500



A way to understand the world...

Abstractions

- abstraction of time
 - think of ticks of the clock. . .
 - continuous-time domain sampled into the discrete-time domain
 - a single discrete time-step can abstract some notion of time
 - one generation of bacteria population
 - time abstracted to day/night phases or a.m./p.m.
 - the lowest time observed between any two event occurrences
- abstraction of quantities
 - large (possibly real) valued amounts abstracted by several discrete levels (e.g., light intensity, temperature, . . .)
 - large valued amounts abstracted by real values (e.g., concentration of molecules in a cell of particular volume)
- abstraction of behaviour
 - neglecting some aspects (e.g., competitiveness in transcription factors-to-DNA binding)
 - theoretical constraints (e.g., well-stirred solutions, fixed conditions – temperature, pressure)

A way to understand the world...

Abstractions

- in computer science abstraction is a formal notion:
 - identify what you add and what you lose by abstracting
- when doing models of life we use approximative abstractions
 - what is lost and what is added?
- note that abstractions between continuous and discrete views can be bidirectional

A way to understand the world...

Using computers

- computation by computers is purely discrete
- quantities are stored in discrete memory
- memory has limited size
- computer-scientific abstractions are employed to automatically (technically) reduce the computational efforts

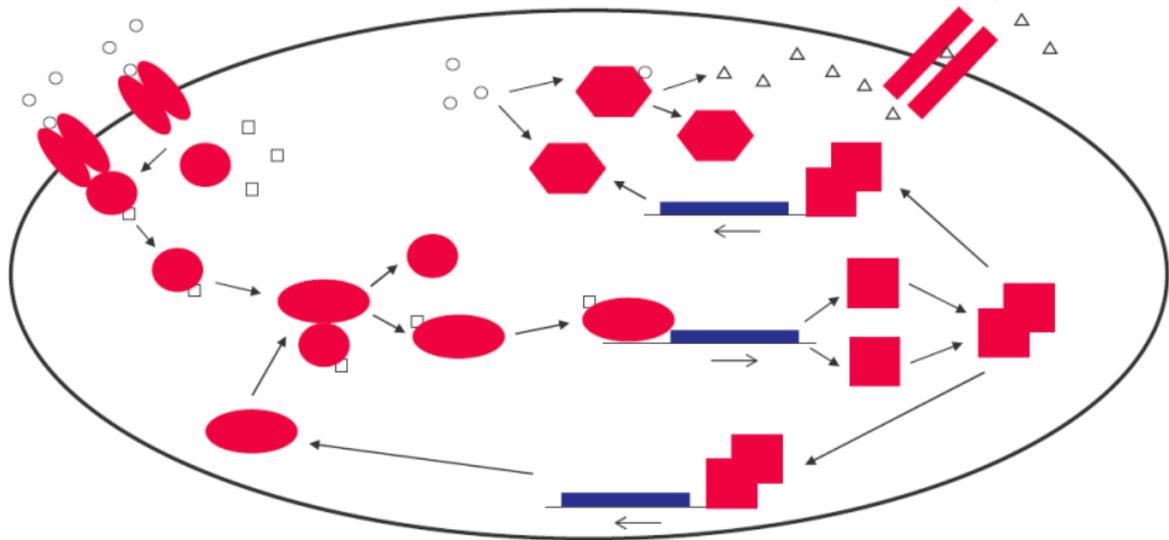
Systems View of the World

- systems models of the world (e.g., a living cell, a population)
- syntax of the systems model is a network:
 - components (nodes) – e.g. chemical substances
 - interactions (edges) – e.g. chemical reactions
- each component is assigned some quantity
 - discrete or continuous: number of particles, concentration
 - can be visualized by color intensity of a node
- dynamics is animation of colour intensity changing on nodes in time
 - driven by global rules (e.g., mass-action reactions)

Model Types

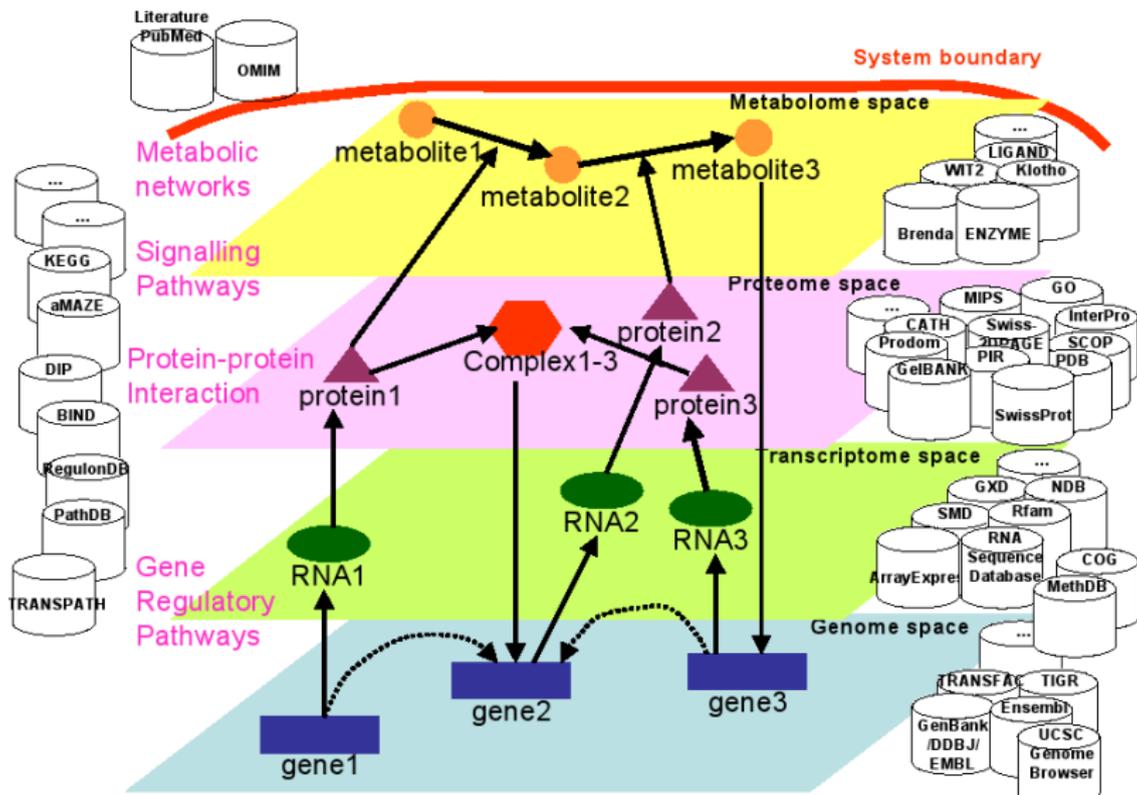


Systems View of a Cell: Biological Networks

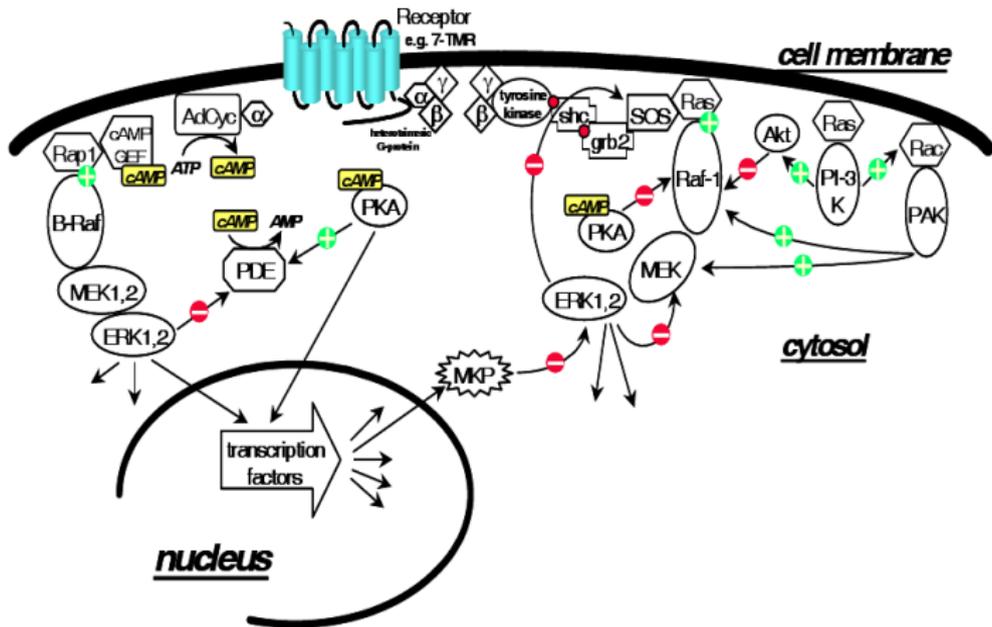


- identify substances (macromolecules, ligands, proteins, genes, ...)
- identify interactions ((de)complexation, (de)phosphorylation, ...)

Systems Biology of a Cell



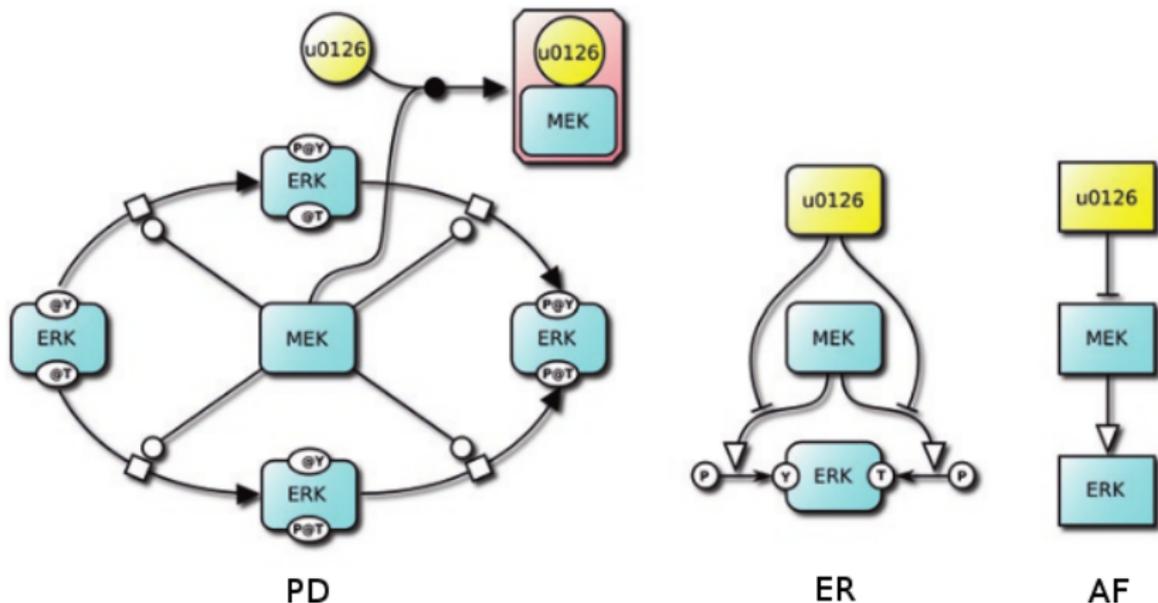
Biological Networks and Pathways



- what is the “right” meaning?
- in order to *analyse* we need to *formalise* a bit. . .

Graphical Specification in SBGN

Systems Biology Graphical Notation



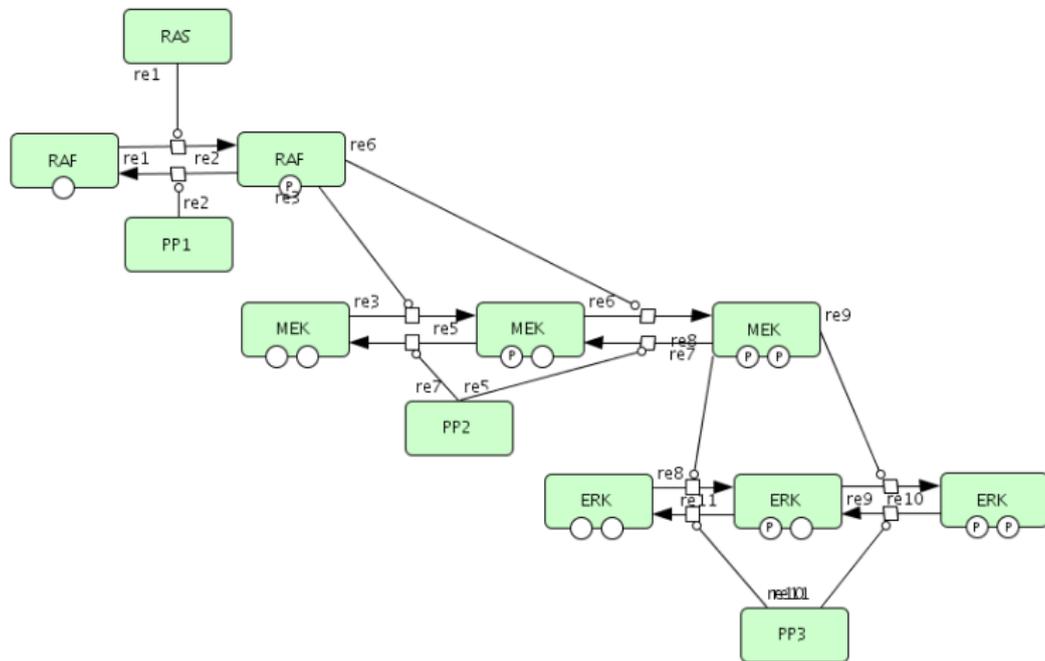
- PD: biochemical interaction level (the most concrete)
- ER: relations among components and interactions
- AF: abstraction to mutual interaction among activities

Graphical Specification in SBGN

Systems Biology Graphical Notation

- SBGN.org initiative (from 2008)
- standard notation for biological processes
- <http://sbgn.org>
- Nature Biotechnology (doi:10.1038/nbt.1558, 08/2009)
- three sub-languages:
 - SBGN PD (Process Description)
(doi:10.1038/npre.2009.3721.1)
 - SBGN ER (Entity Relationship)
(doi:10.1038/npre.2009.3719.1)
 - SBGN AF (Activity Flow) (doi:10.1038/npre.2009.3724.1)
- tool support:
 - SBGN PD supported by CellDesigner
 - SBGN-ED (<http://www.sbgn-ed.org>)

Kinase Cascade in CellDesigner (SBGN)

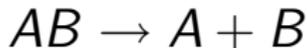


From Systems Structure to Systems Dynamics

Discrete Events View

- once the systems structure is captured we want to animate . . .
- each node in SBGN PD represents some quantity of a particular species
- species are assumed to be well stirred in the cell
- interactions are events affecting related species (in amounts given by stoichiometry)
 - abstract from time (no information when the event occurred, events last zero time)
 - usually abstract from space (no information where the event occurred)
 - but see agent-based models at the end of the lecture
 - purely **qualitative model** of systems dynamics

Qualitative Model of a Reaction



- state configuration captures number of molecules:

$$\langle \#[AB], \#[A], \#[B] \rangle$$

- global rule:
 - one molecule AB is removed from the solution
 - one molecule A is added to the solution
 - one molecule B is added to the solution

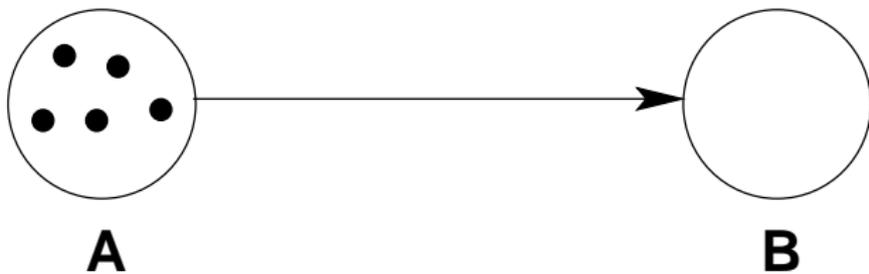
$$\#[AB](t + 1) = \#[AB](t) - 1$$

$$\#[A](t + 1) = \#[A](t) + 1$$

$$\#[B](t + 1) = \#[B](t) + 1$$

Example

Consider reaction: $A \rightarrow B$



$$\begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

Example

Consider reaction: $A \rightarrow B$



$$\begin{pmatrix} 5 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

Example

Consider reaction: $A \rightarrow B$



$$\begin{pmatrix} 5 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 4 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Example

Consider reaction: $A \rightarrow B$



$$\begin{pmatrix} 5 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 4 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 3 \\ 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Example

Consider reaction: $A \rightarrow B$



$$\begin{pmatrix} 5 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 4 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 3 \\ 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 \\ 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

Example

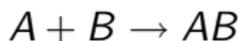
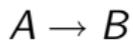
Consider reaction: $A \rightarrow B$



$$\begin{pmatrix} 5 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 4 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 3 \\ 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 \\ 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 4 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

Example

Consider three reactions:



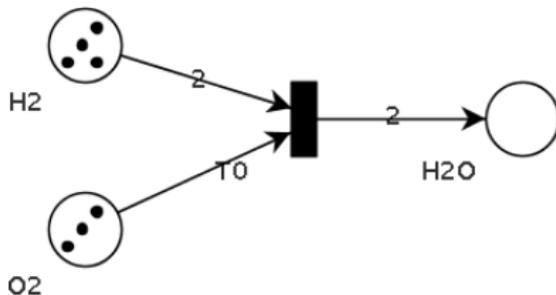
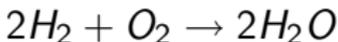
- state configuration has the form $\langle \#A, \#B, \#AB \rangle$
- consider, e.g., configuration $\langle 2, 2, 1 \rangle$
⇒ what is the next configuration?
- reactions run in parallel ...

Petri Nets – A Discrete Model of Chemical World



Adam Carl
Petri
1926–2010

- formal model for concurrent systems
- used for general modeling and simulation
- many simulation and analysis tools
- various semantics
- unambiguous system representation

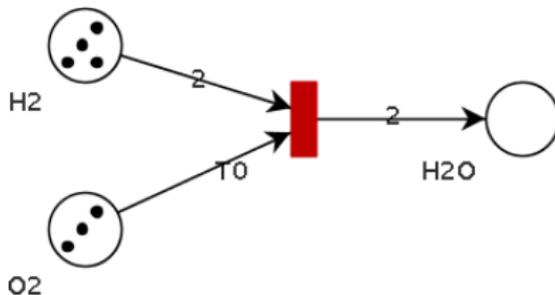
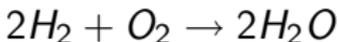


Petri Nets – A Discrete Model of Chemical World



Adam Carl
Petri
1926–2010

- formal model for concurrent systems
- used for general modeling and simulation
- many simulation and analysis tools
- various semantics
- unambiguous system representation

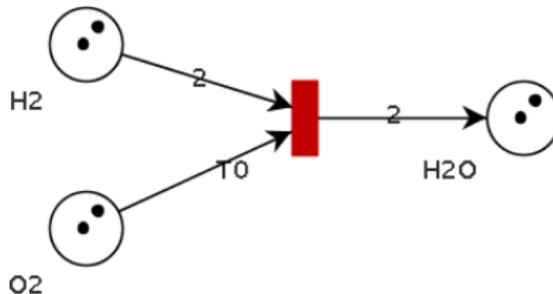
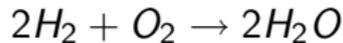


Petri Nets – A Discrete Model of Chemical World



Adam Carl
Petri
1926–2010

- formal model for concurrent systems
- used for general modeling and simulation
- many simulation and analysis tools
- various semantics
- unambiguous system representation

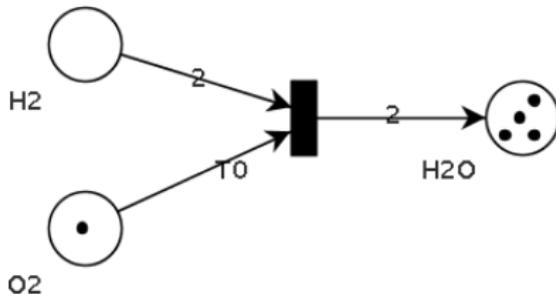
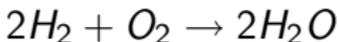


Petri Nets – A Discrete Model of Chemical World

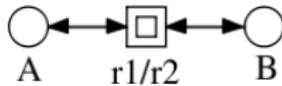
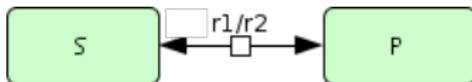
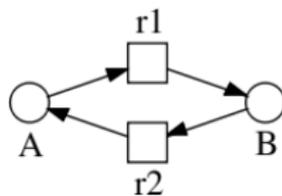
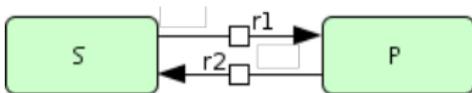
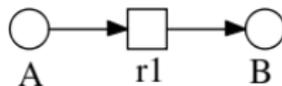


Adam Carl
Petri
1926–2010

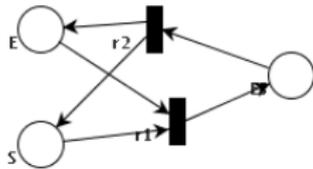
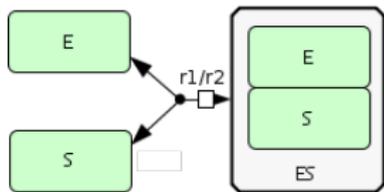
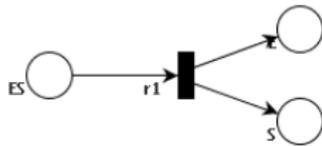
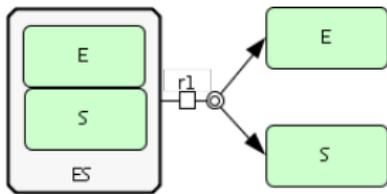
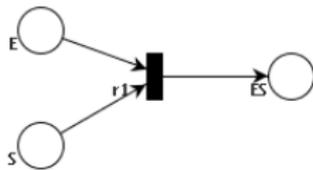
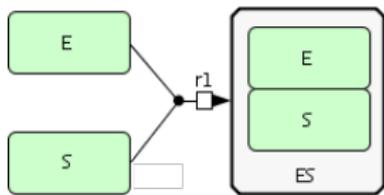
- formal model for concurrent systems
- used for general modeling and simulation
- many simulation and analysis tools
- various semantics
- unambiguous system representation



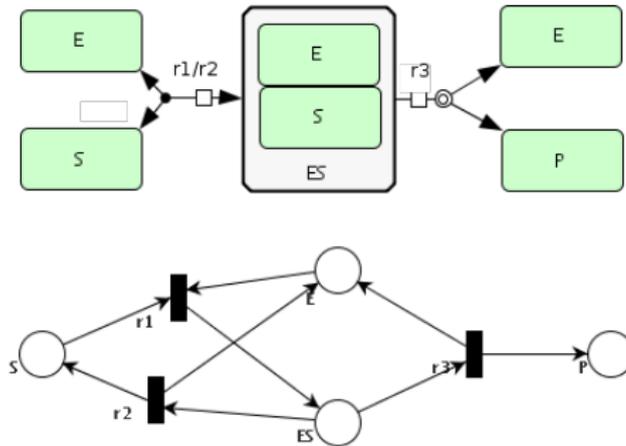
Petri Net Representation of Biological Networks



Petri Net Representation of Biological Networks



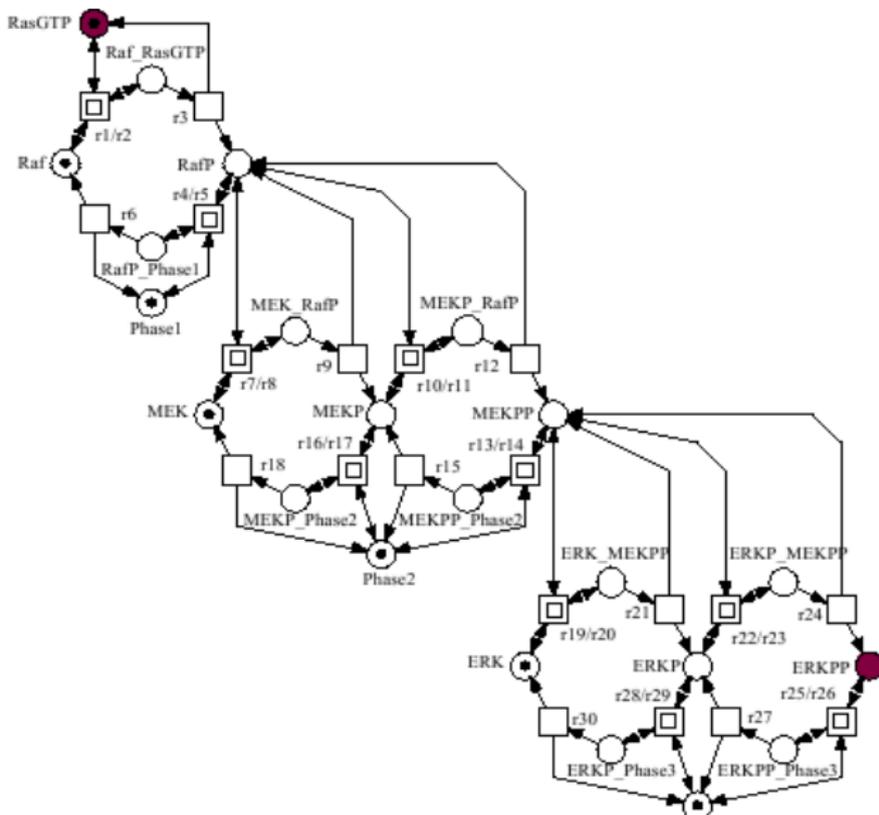
Petri Net Representation of Biological Networks



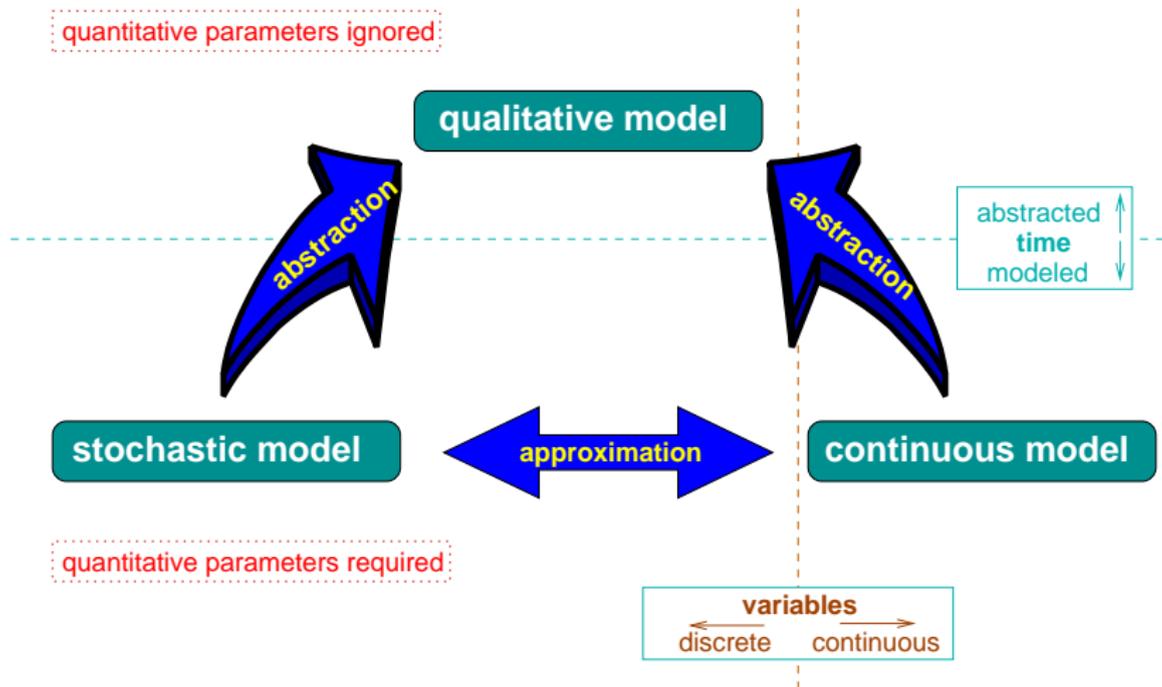
- visually infeasible but unambiguous, formal, and still graphical

Petri Net Representation of Biological Networks

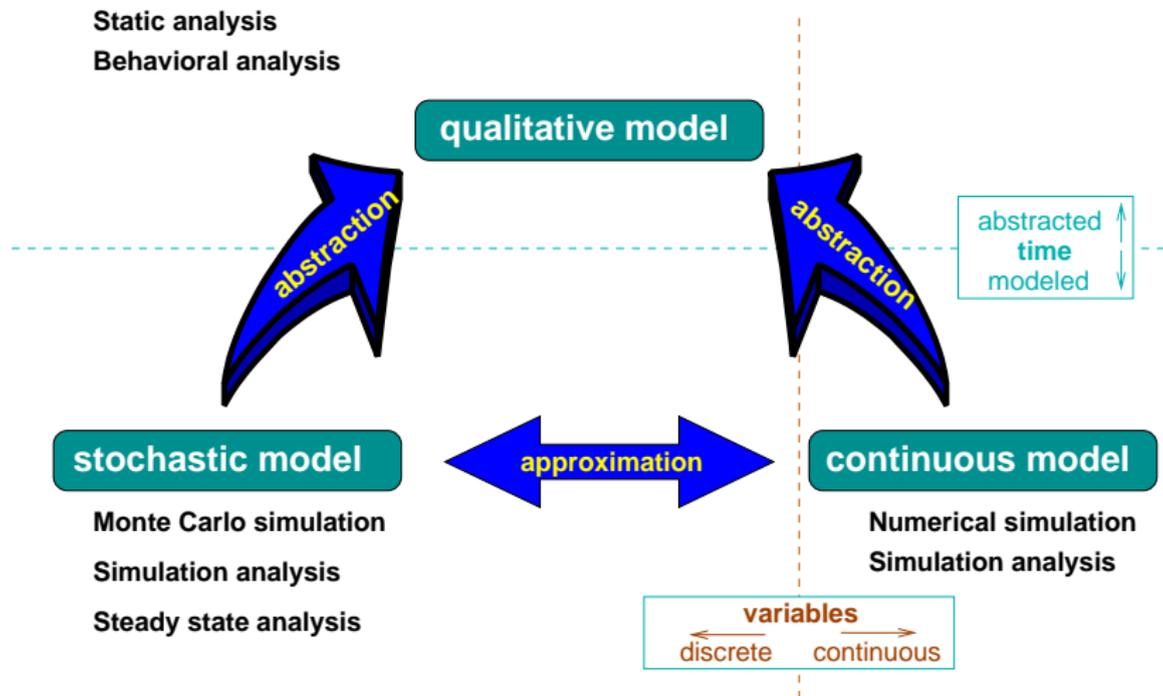
Kinase Cascade in MAPK/ERK Signalling Network



Petri Net Analysis Framework



Petri Net Analysis Framework



Petri Net Structure

(Place/Transition) Petri net is a quadruple $\mathcal{N} = \langle S, R, f, m(0) \rangle$ where

- S finite set of *places* (substances),
- T finite set of *transitions* (reactions),
- $f : ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}_0$ set of weighted edges,
 - $x\bullet = \{y \in S \cup R \mid f(x, y) \neq 0\}$ denotes target of x
 - $\bullet x = \{y \in S \cup R \mid f(y, x) \neq 0\}$ denotes source of x
 - weight represents stoichiometric coefficients
- $m(0) : S \rightarrow \mathbb{N}_0$ is *initial marking* (initial condition).

Petri Net Dynamics

Each place $s \in S$ is marked by a value in \mathbb{N}_0 (representing number of molecules of the respective species):

- in Petri net terminology the state configuration (configuration of places) is called *marking*: $m : S \rightarrow \mathbb{N}_0$
- marking is represented as an n -dimensional vector $m \in \mathbb{N}_0^n$

Dynamics of each *transition* $r \in R$ is the following:

- a transition $r \in R$ must be *enabled* iff $\forall s \in \bullet r. m(s) \geq f(s, r)$
- enabled transition can be fired, causing an update of related places (marking m is updated to marking m'):

$$\forall s \in S. m'(s) = m(s) - f(r, s) - f(s, r)$$

Static Analysis

Matrix Representation

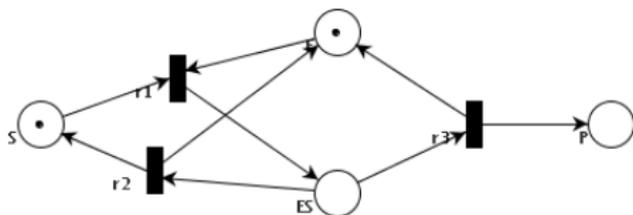
- Petri Net can be represented by *incidence matrix*

$\mathbb{M} : S \times R \rightarrow \mathbb{Z}$ defined as follows:

$$\mathbb{M}(s, r) = f(r, s) - f(s, r)$$

- equivalent to stoichiometric matrix

Note: If reversible reactions are considered, forward and backward matrices must be distinguished.



$$\mathbb{M} = \begin{pmatrix} -1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & -1 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

species indices: $s_1 \dots E, s_2 \dots S, s_3 \dots ES, s_4 \dots P$

reaction indices: $r_1 : E + S \rightarrow ES, r_2 : ES \rightarrow E + S, r_3 : ES \rightarrow P + E$

Static Analysis

Invariant Analysis – P -invariants

- species vector x is called P -invariant iff

$$xM = 0$$

- characteristic property of P -invariant x :

$$\forall r \in R. \sum_{i=1}^n M(i, r)x_i = 0$$

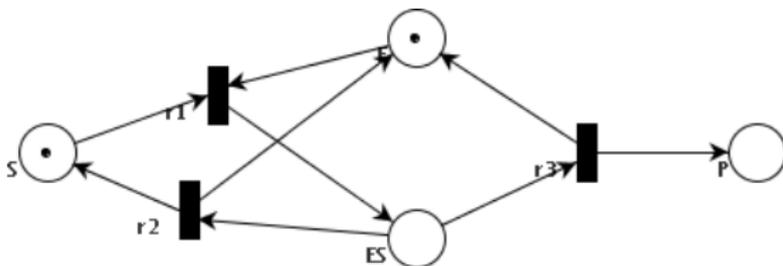
- Petri Net terminology translates the P -invariant property to:

$$\forall r \in R. \sum_{s \in \bullet r} x_s = \sum_{s \in r \bullet} x_s$$

- interpretation: conserved mass

Static Analysis

Invariant Analysis – P -invariants



species indices: $s_1 \dots E, s_2 \dots S, s_3 \dots ES, s_4 \dots P$

- minimal P -invariants:
 - $(1, 0, 1, 0)$: $m_E + m_{ES} = \text{const.}$
 - $(0, 1, 1, 1)$: $m_S + m_{ES} + m_P = \text{const.}$
- minimal P -invariants make the basis of \mathbb{M} left-null space
- non-minimal P -invariant example:
 - $(1, 1, 2, 1)$: $m_E + m_S + 2m_{ES} + m_P = \text{const.}$

Static Analysis

Invariant Analysis – T -invariants

- species vector y is called T -invariant iff

$$\mathbb{M}y = 0$$

- characteristic property of T -invariant y :

$$\forall s \in S. \sum_{j=1}^{|R|} \mathbb{M}(s, j)y_j = 0$$

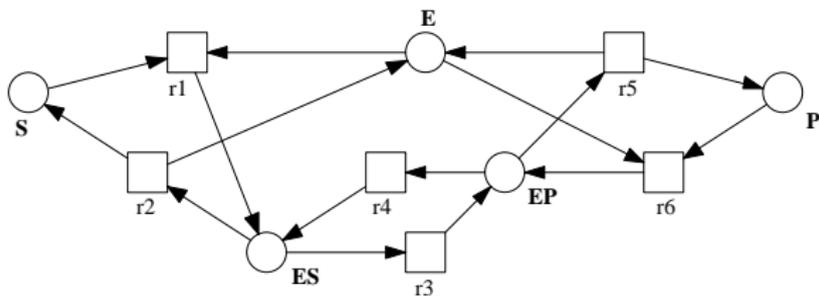
- Petri Net terminology translates the T -invariant property to:

$$\forall s \in S. \sum_{r \in \bullet s} y_r = \sum_{r \in s \bullet} y_r$$

- interpretation: stable flux mode

Static Analysis

Invariant Analysis – T -invariants



$$E + S \leftrightarrow ES \leftrightarrow EP \leftrightarrow P + E$$

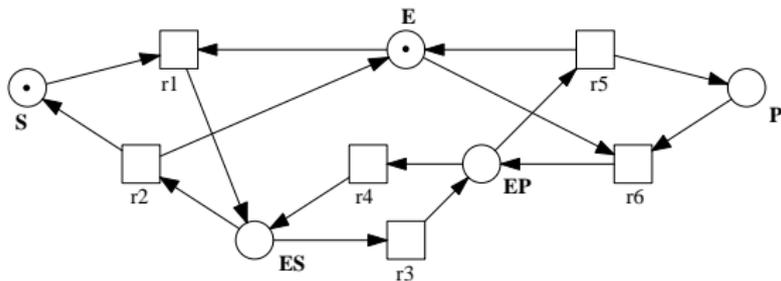
- minimal T -invariants:
 - $(1, 1, 0, 0, 0, 0)$: $r_1; r_2$ (trivial – reversibility)
 - $(0, 0, 1, 1, 0, 0)$: $r_3; r_4$ (trivial – reversibility)
 - $(0, 0, 0, 0, 1, 1)$: $r_5; r_6$ (trivial – reversibility)
- minimal T -invariants make the basis of \mathbb{M} null space
- non-minimal (and non-trivial) T -invariant example:
 - $(1, 1, 1, 1, 1, 1)$: $r_1; r_2; r_3; r_4; r_5; r_6$
 - this T -invariant represents significant input/output behaviour of the network ($S \longrightarrow P$)

Qualitative Behavioral Properties

- boundedness
 - no species concentration expands indefinitely
 - can be decided statically (P -invariant coverability)
- liveness
 - weak form – always at least one reaction is working
 - can be decided statically in some cases
 - strong form – every reaction is always eventually working
 - can be decided statically in some cases (T -invariant coverability)
- reversibility
 - reversible process (each flux mode keeps realizing)
 - can be decided only by dynamical analysis
- dynamic properties independent of time and quantity

Qualitative Behavioral Properties

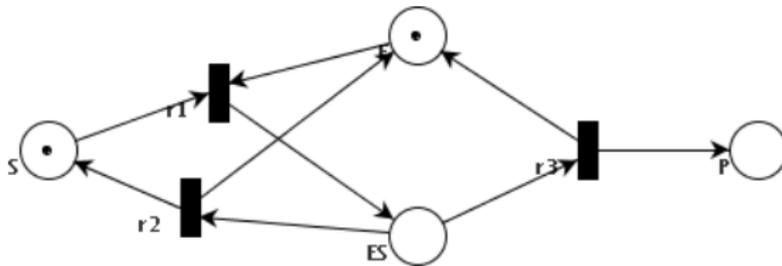
Example Michaelis Menten Reversible



- bounded, strongly live, reversible
- S and E is never finally consumed (marked zero)
- P finally marked non-zero

Qualitative Behavioral Properties

Example Michaelis Menten Kinetics



- bounded, not live, not reversible
- S finally consumed
- E consumed but recovered finally

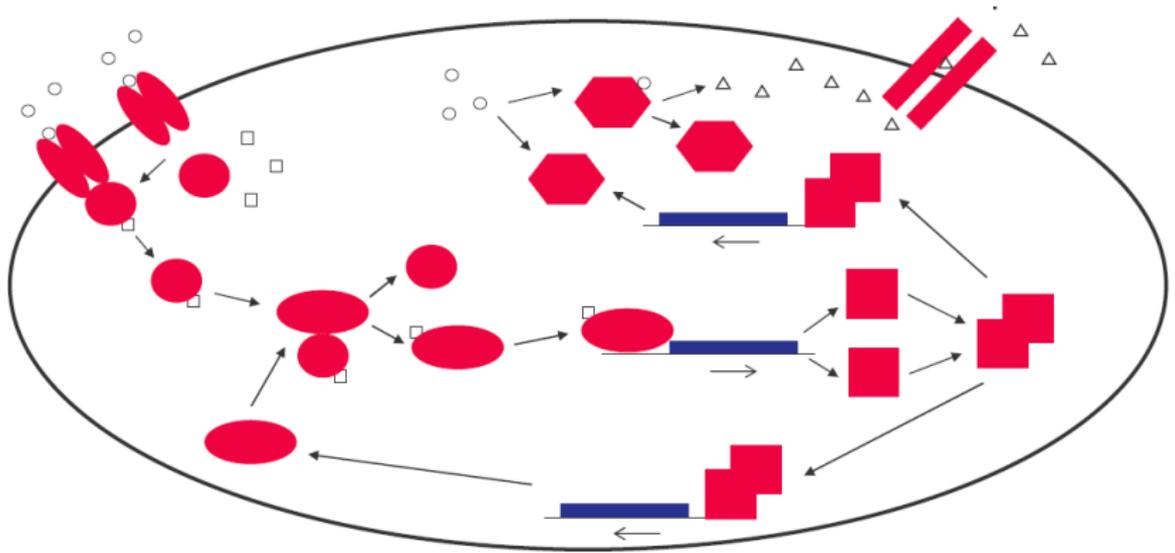
Tool Support

- format transformation and editing
 - Snoopy – visual editing, SBML export/import, Petri net variants transformation
 - PIPE – visual editing
- static analysis
 - Charlie (Petri Net invariant analysis)
 - also can be used: Matlab/Octave (stoichiometric analysis)
- qualitative behavioral analysis
 - Charlie (liveness, boundedness, and more ...)
 - MARCIE (qualitative dynamics analysis)
- Petri Net simulation
 - Snoopy – simulation of qualitative and stochastic nets
 - Cell Illustrator – professional tool, hybrid semantics

Literature

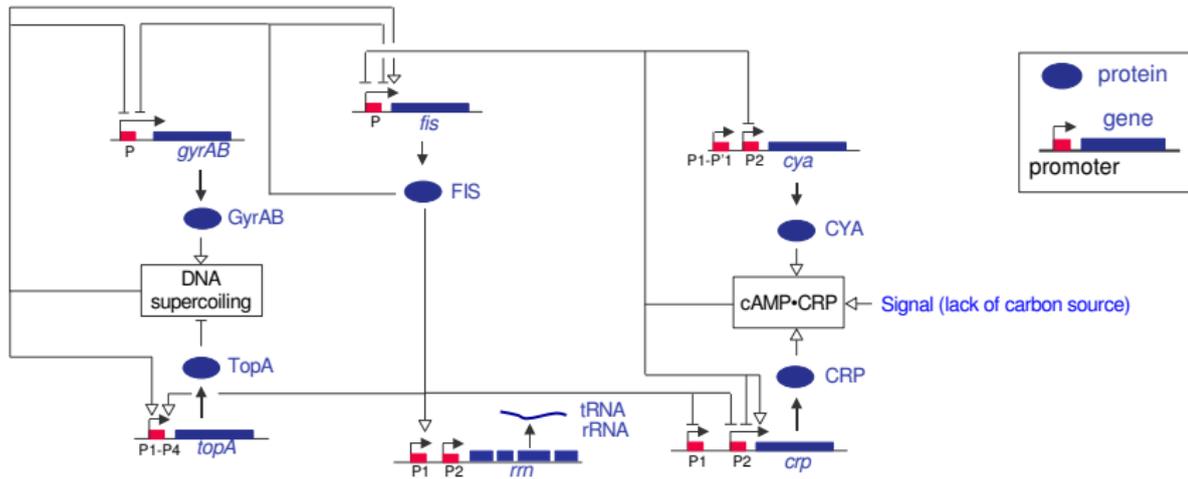
-  M. Heiner & D. Gilbert. *How Might Petri Nets Enhance Your Systems Biology Toolkit*. Petri Nets 2011: 17-37
-  M. Heiner, D. Gilbert & R. Donaldson. *Petri Nets for Systems and Synthetic Biology*. SFM 2008: 215-264
-  Koch, I., Reisig, W. & Schreiber, F. *Modeling in Systems Biology: The Petri Net Approach*. Computational Biology, Vol. 16, Springer-Verlag, 2011.
-  Pinney, J.W., Westhead, D.R. & McConkey, G.A. *Petri Net representations in systems biology*. Biochem Soc Trans. 2003 Dec;31(Pt 6):1513-5.
-  Reddy, V.N., Mavrovouniotis M.L.& Liebman M.N. *Petri net representation in metabolic pathways*. Proc Int Conf Intell Syst Mol Biol 1993, 328-336.

Regulatory Networks of Cellular Processes



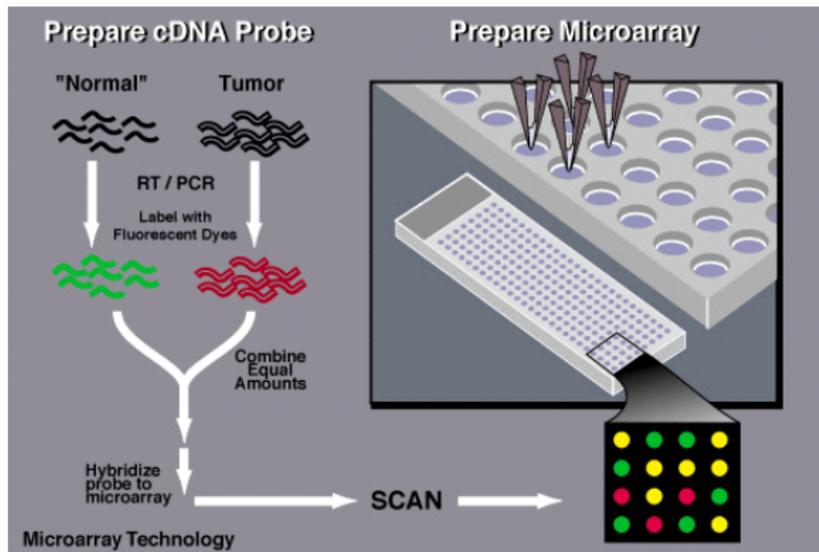
- identify substances (proteins, genes)
- identify interactions (transcriptional activation, repression – do we know reactions behind?)

Example of a gene regulatory network

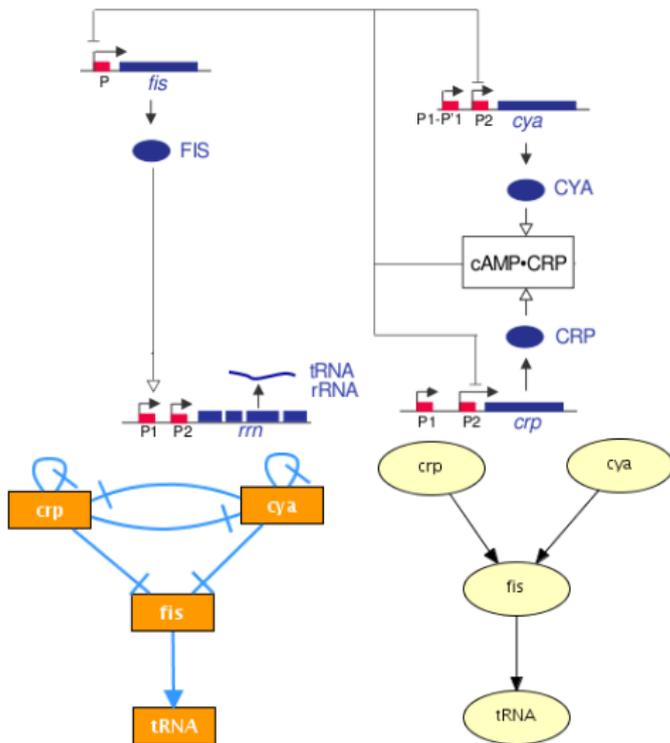


Identification of regulatory dynamics

- systems measurement of transcriptome (mRNA concentration) is imprecise and discrete!
- interactions can be partially identified by analysis of transcription factor binding sites (e.g., TRANSFAC)
- microarray experiments can be reversed engineered

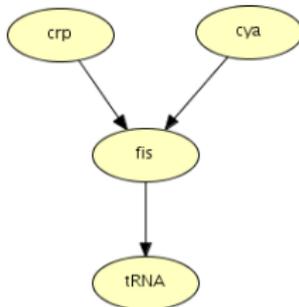
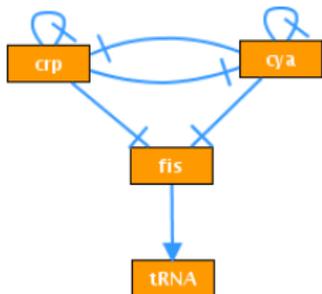


Identification of regulatory dynamics



Identification of regulatory dynamics

Boolean and Bayesian networks



$$crp(t+1) = \neg crp(t) \wedge \neg cya(t)$$

$$cya(t+1) = \neg cya(t) \wedge \neg crp(t)$$

$$fis(t+1) = \neg crp(t) \wedge \neg cya(t)$$

$$tRNA(t+1) = fis(t)$$

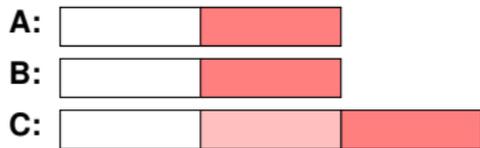
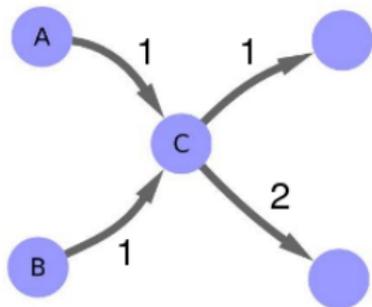
$$P(X_{crp})$$

$$P(X_{cya})$$

$$P(X_{fis} | X_{crp}, X_{cya})$$

$$P(X_{tRNA} | X_{fis})$$

From Structure to Dynamics



$$\emptyset \rightarrow 0$$



$$\{A\} \rightarrow 2$$

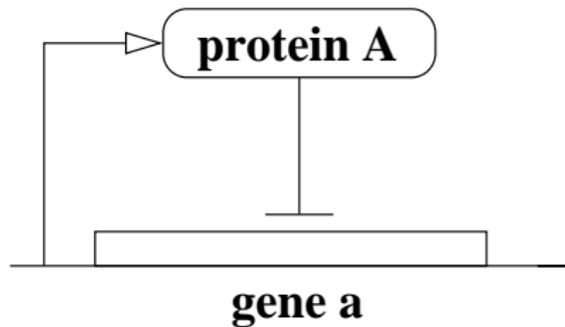


$$\{B\} \rightarrow 2$$

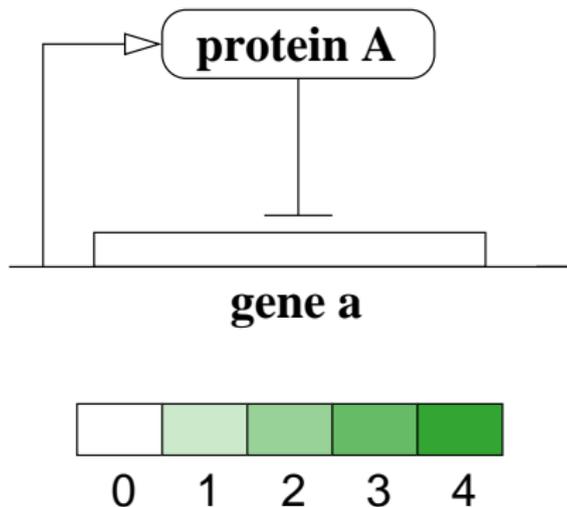


$$\{A, B\} \rightarrow 1$$

Model example – autoregulation

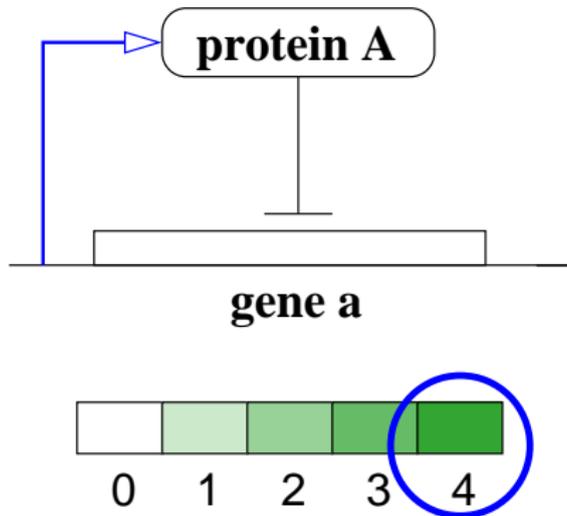


Model example – autoregulation



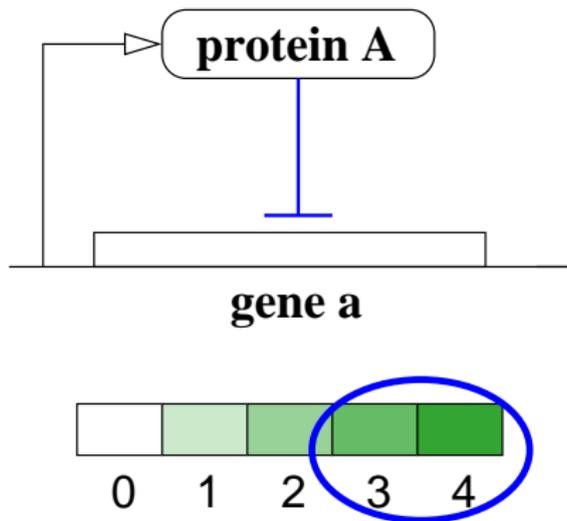
- identification of discrete expression levels

Model example – autoregulation



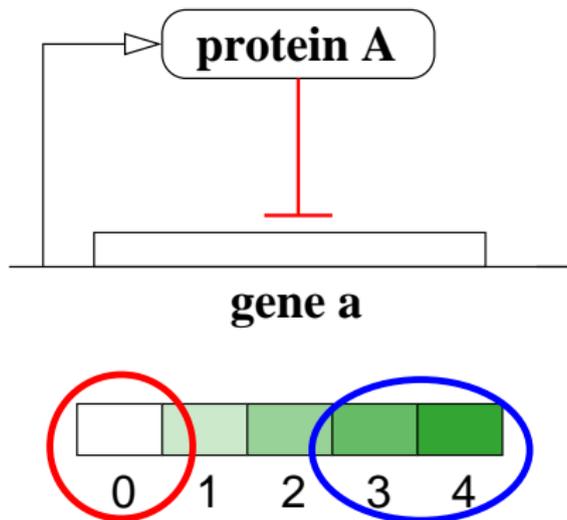
- spontaneous (basal) transcription: $A \rightarrow 4$

Model example – autoregulation



- range of regulatory activity ($A \in \{3, 4\} \Rightarrow$ regulation active)

Model example – autoregulation



- target level ($A \in \{3, 4\} \Rightarrow A \rightarrow 0$)

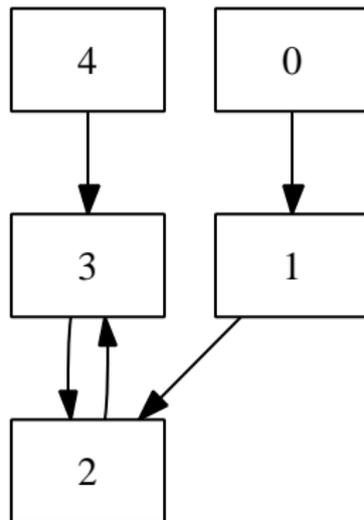
State space – autoregulation

- state transition system $\langle S, T, S_0 \rangle$
 - S state set, $S \equiv \{0, 1, 2, 3, 4\}$
 - $S_0 \subseteq S$ initial state set
 - $T \subseteq S \times S$ transition function:

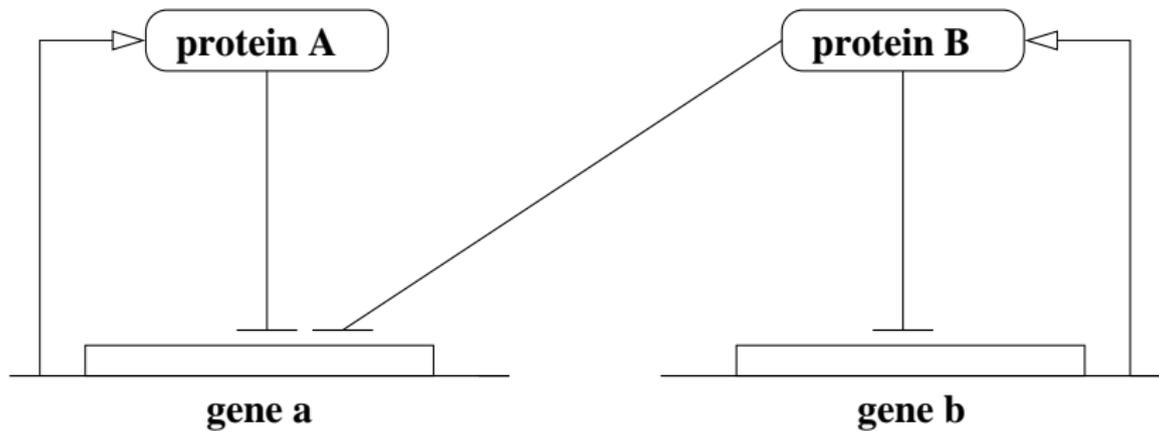
source state	active regulation	target state
0	$\emptyset; [A \rightarrow 4]$	1
1	$\emptyset; [A \rightarrow 4]$	2
2	$\emptyset; [A \rightarrow 4]$	3
3	$A \rightarrow^- A; [A \rightarrow 0]$	2
4	$A \rightarrow^- A; [A \rightarrow 0]$	3

State space – autoregulation

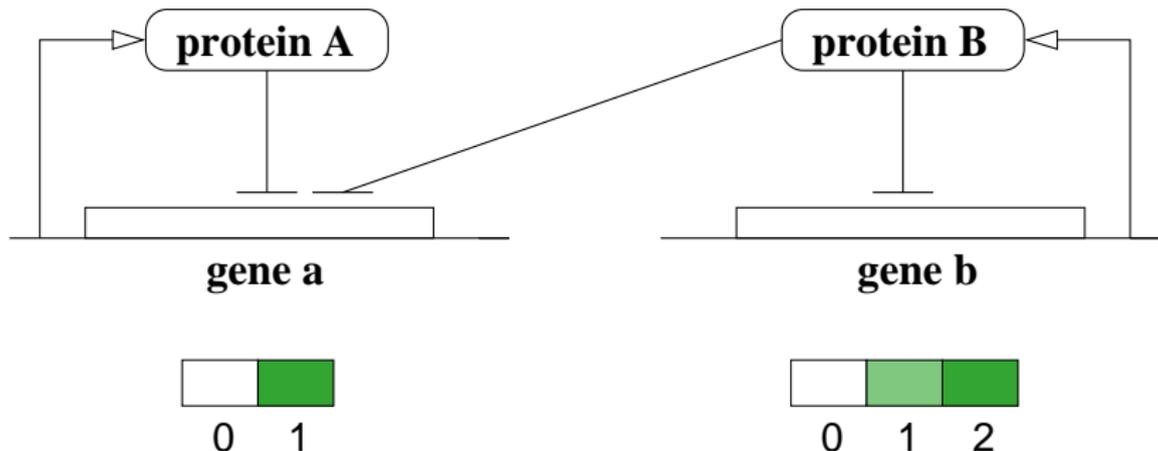
state transition system for negative autoregulation $\langle S, T, S_0 = S \rangle$:



Combined regulation

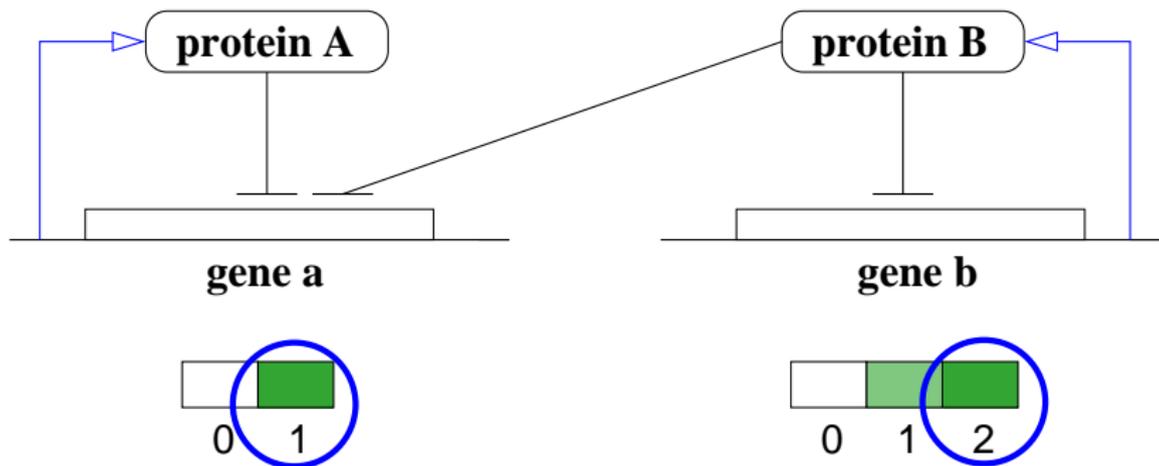


Discrete characteristics of dynamics



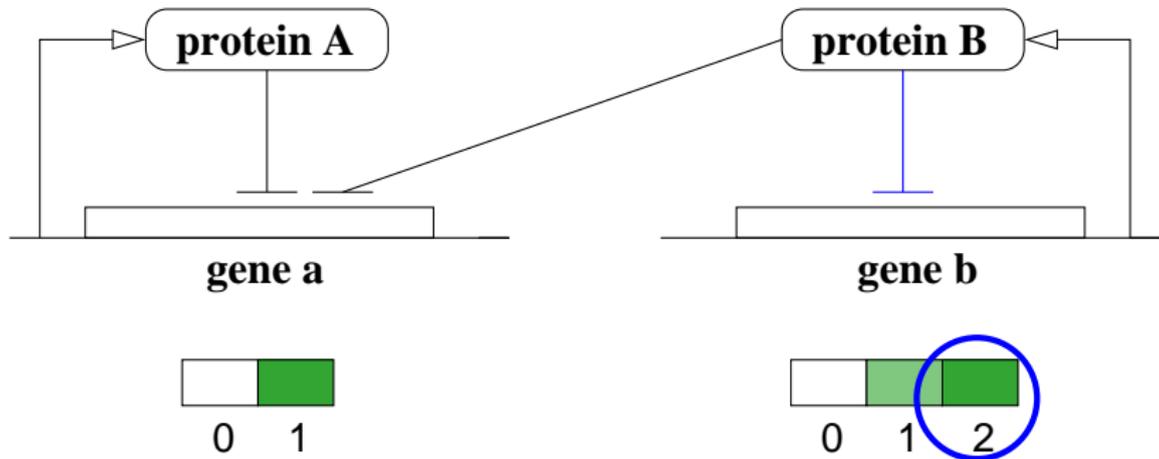
- identification of discrete levels of expression

Discrete characteristics of dynamics



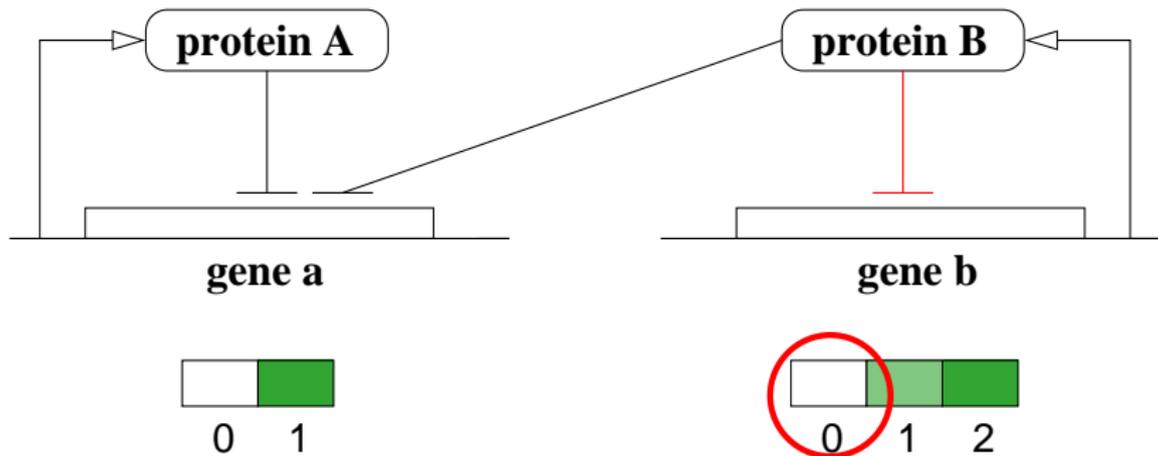
- spontaneous (basal) transcription: $A \rightarrow 1$, $B \rightarrow 2$

Characteristics of regulation



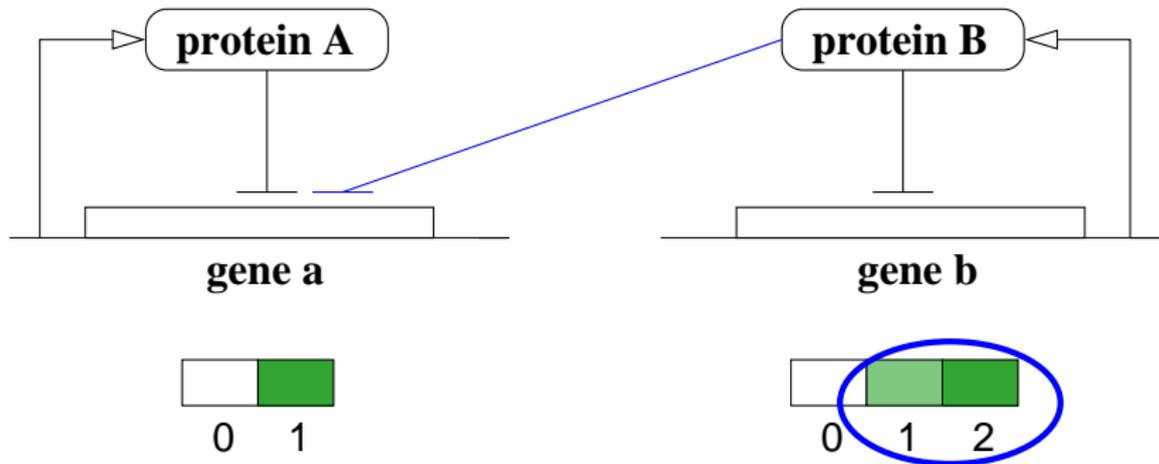
- range of regulatory activity $B \rightarrow^- B$ ($B = 2 \Rightarrow$ regulation active)

Characteristics of regulation



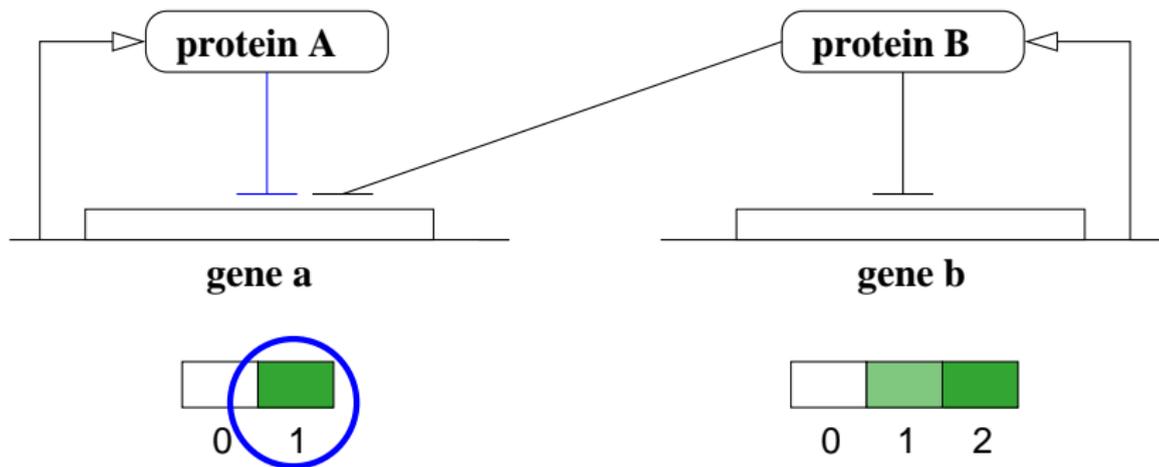
- target level $B \rightarrow^- B$ ($B = 2 \Rightarrow B \rightarrow 0$)

Characteristics of regulation – input function



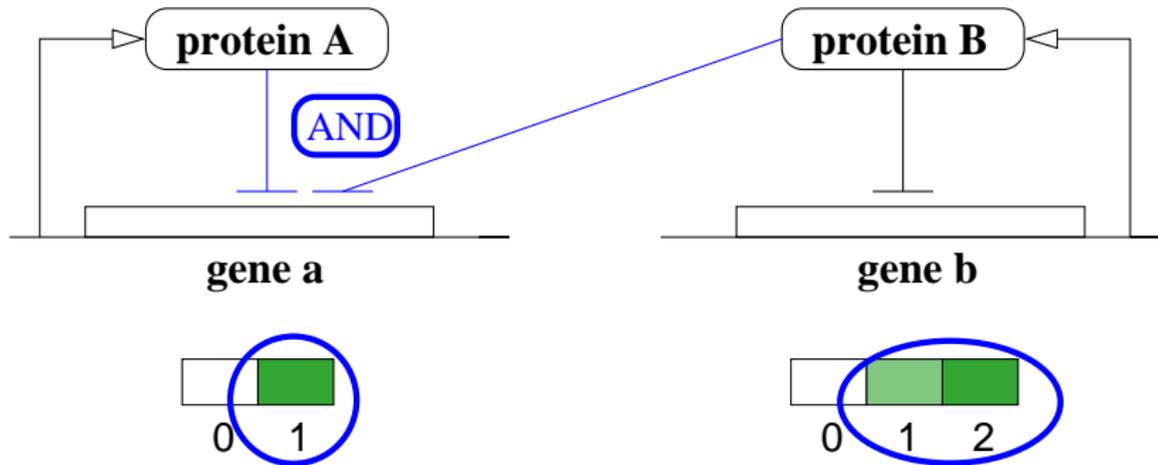
- range of regulatory activity $B \rightarrow^- A$ ($B \in \{1, 2\} \Rightarrow$ reg. active)

Characteristics of regulation – input function



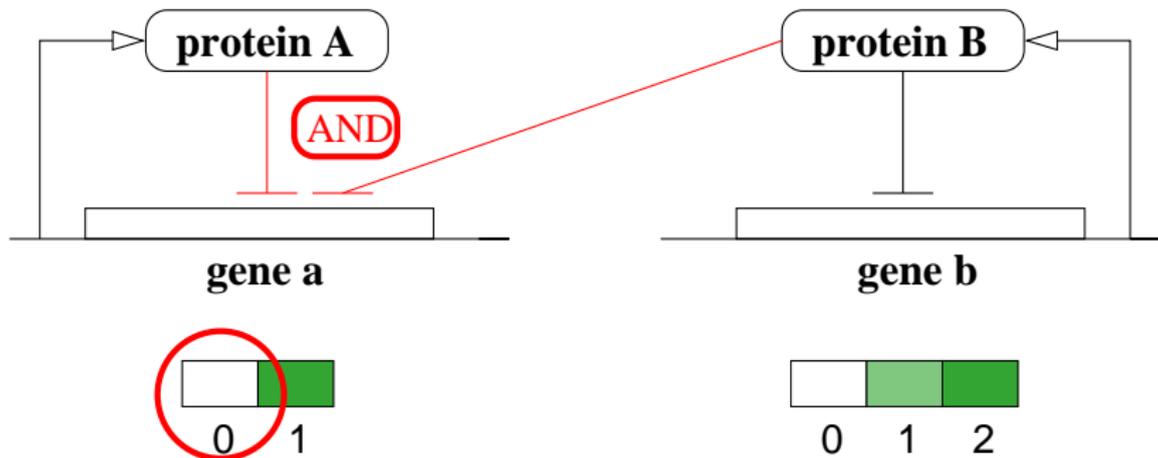
- range of regulatory activity $A \rightarrow^- A$ ($A = 1 \Rightarrow$ reg. active)

Characteristics of regulation – input function



- AND-combined regulation $A \rightarrow^- A \wedge B \rightarrow^- A$:
 $A = 1 \wedge B \in \{1, 2\} \Rightarrow$ regulation active

Characteristics of regulation – input function



- target levels of combined regulation $A \rightarrow^- A \wedge B \rightarrow^- A$:
 $A = 1 \wedge B \in \{1, 2\} \Rightarrow A \rightarrow 0$

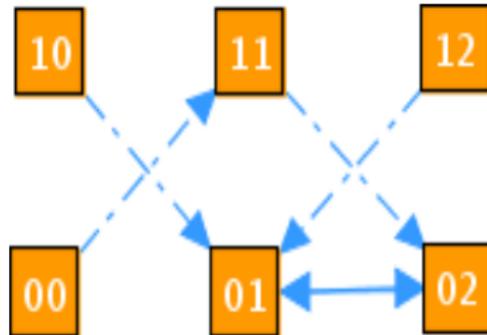
State space – synchronous semantics

- state transition system $\langle S, T, S_0 \rangle$
 - $S \equiv \{0, 1\} \times \{0, 1, 2\}$
 - $S_0 \subseteq S$, we consider $S_0 = S$
 - $T \subseteq S \times S$ transition function:

source state	active regulation	target state
[0, 0]	$\emptyset; [A \rightarrow 1, B \rightarrow 2]$	[1, 1]
[0, 1]	$B \rightarrow^- A; [A \rightarrow 0, B \rightarrow 2]$	[0, 2]
[0, 2]	$B \rightarrow^- B \wedge B \rightarrow^- A; [A \rightarrow 0, B \rightarrow 0]$	[0, 1]
[1, 0]	$A \rightarrow^- A; [A \rightarrow 0, B \rightarrow 2]$	[0, 1]
[1, 1]	$A \rightarrow^- A \wedge B \rightarrow^- A; [A \rightarrow 0, B \rightarrow 2]$	[0, 2]
[1, 2]	$A \rightarrow^- A \wedge B \rightarrow^- A \wedge B \rightarrow^- B; [A \rightarrow 0, B \rightarrow 0]$	[0, 1]

State space – synchronous semantics

state transition system $\langle S, T, S_0 = S \rangle$:



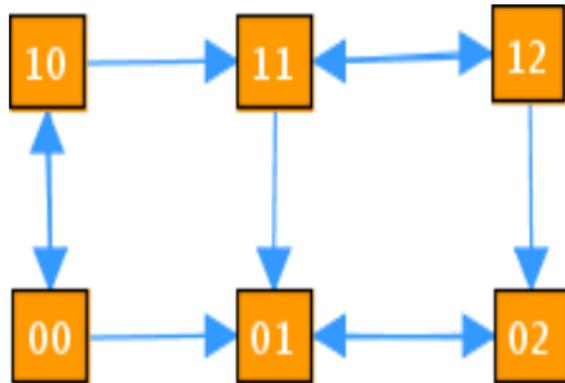
State space – asynchronous semantics

- state transition system $\langle S, T, S_0 \rangle$
 - $S \equiv \{0, 1\} \times \{0, 1, 2\}$
 - $S_0 \subseteq S$, we consider $S_0 = S$
 - $T \subseteq S \times S$ transition function:

source state	active regulation	target states
[0, 0]	$\emptyset; [A \rightarrow 1, B \rightarrow 2]$	[1, 0], [0, 1]
[0, 1]	$B \rightarrow^- A; [A \rightarrow 0, B \rightarrow 2]$	[0, 2]
[0, 2]	$B \rightarrow^- B \wedge B \rightarrow^- A; [A \rightarrow 0, B \rightarrow 0]$	[0, 1]
[1, 0]	$A \rightarrow^- A; [A \rightarrow 0, B \rightarrow 2]$	[0, 0], [1, 1]
[1, 1]	$A \rightarrow^- A \wedge B \rightarrow^- A; [A \rightarrow 0, B \rightarrow 2]$	[0, 1], [1, 2]
[1, 2]	$A \rightarrow^- A \wedge B \rightarrow^- A \wedge B \rightarrow^- B; [A \rightarrow 0, B \rightarrow 0]$	[0, 2], [1, 1]

State space – asynchronous semantics

state transition system $\langle S, T, S_0 = S \rangle$:



Properties of discrete semantics

- synchronous semantics
 - effect of active regulations is realized in terms of a single event
 - strong approximation leading to deterministic state transition system
- asynchronous semantics
 - effect of active regulations is realized for each gene/protein individually in terms of single events
 - nondeterminism models all possible serializations (so called interleaving)
 - approximation is rather conservative

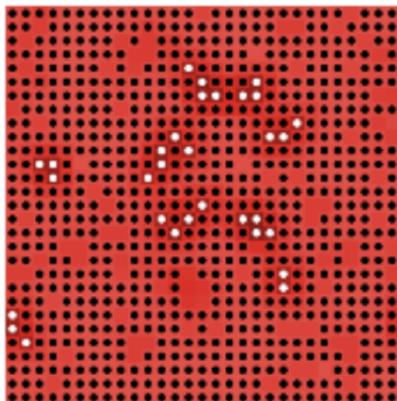
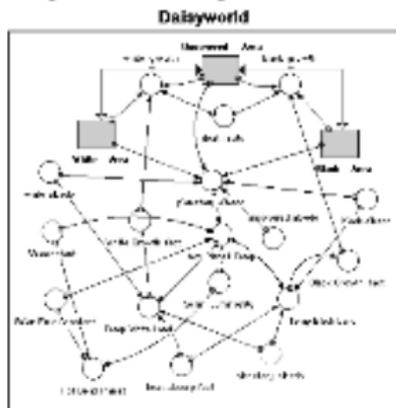
Free Tool Support

- Gene Interaction Network simulation (GINsim)
<http://gin.univ-mrs.fr/GINsim/accueil.html>
- asynchronous and synchronous simulation
 - allows to get rough understanding of regulatory logic
 - allows to identify potential steady states of regulation
 - purely qualitative modelling and analysis
- directly allow application of a large set of computer scientific tools
 - graph algorithms for state space graph analysis
 - model checking

Literature

-  Thomas, R. *Regulatory networks seen as asynchronous automata : a logical description*. J. Theor. Biol. 153 ,(1991) 1-23.
-  de Jong. *Modeling and simulation of genetic regulatory systems: A literature review*. Journal of Computational Biology (2002), 9(1):69-105
-  Bower, J.M. & Bolouri, H. *Computational Modeling of Genetic and Biochemical Networks*. Bradford Book, 2001.
-  A.G. Gonzalez, A. Naldi, L. Sanchez, D.Thieffry, C. Chaouiya. *GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks*. Biosystems (2006), 84(2):91-100

Top-down vs. bottom-up view



top-to-bottom modeling systems view

bottom-to-top modelling agent-based view

Agent-based modeling

- discrete time and discrete space
- strictly local interactions
- dynamics driven by local rules

Agent-based modeling – history

- 1940-60: studying self-reproduction (von Neumann)
 - found a theoretical machine that copies itself
- 1970: Game of Life (Conway)
 - strong simplification of von Neumann machine
 - preserves theoretical computational power of a Turing machine
- 1983: formalization of **cellular automata and applications in physics**

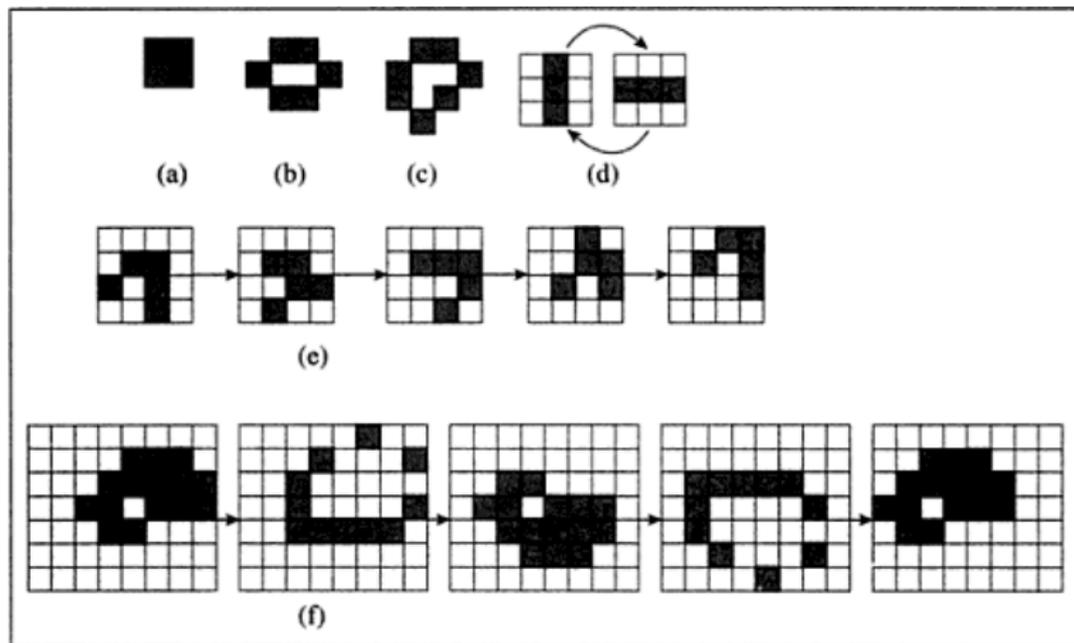
Agent-based modeling – cellular automata

- discrete space (a grid of cells)
- homogeneity (all cell identical)
- finite discrete states of a cell
- interaction strictly local – next state of a cell depends on its nearest neighbours
- discrete-time dynamics – system evolves in discrete time-steps

Game of Life

- unbounded 2D grid of cells
- each cell has two states: dead or live
- dynamics evolves in turns
- local rules for a living cell:
 - **if** less than two neighbours **then** die from loneliness
 - **if** more than three neighbours **then** die from congestion
 - stay alive, **otherwise**
- local rules for a dead cell:
 - **if** just three neighbours **then** go alive
 - stay dead, **otherwise**

Game of Life



B.P. Zeigler, H. Praehofer, T. Gon Kim. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

Game of Life

- problem: is an infinite behaviour possible in the game of life?
- proven true!
- game of life has universal Turing power (a computer with unlimited memory and no time constraints)

Cellular automata – Wolfram's classification

class I dynamics always leads to a stable (non-changing) state

class II dynamics leads to simple repeating (periodical) situations

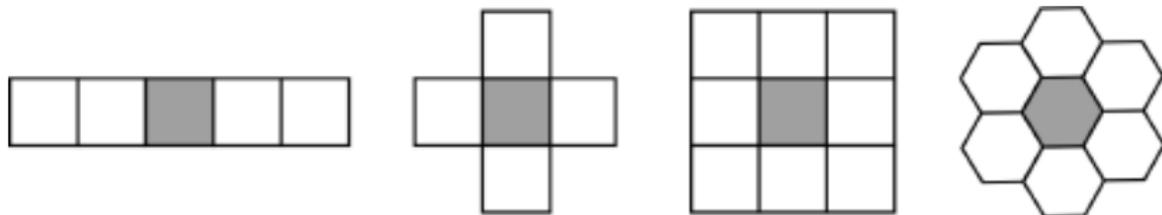
class III dynamics leads to aperiodic, chaotic behaviour

class IV complex patterns moving in the space

Examples:

<http://www.youtube.com/watch?v=XcuBvj0pw-E>

Cellular automata – definition



- each cell i has a neighbourhood $N(i)$
- finite set of local states $\Sigma = \{0, \dots, k - 1\}$
- state of cell i in time t is denoted $\sigma_i(t) \in \Sigma$
- local dynamics rule Φ :
 - $\Phi : \Sigma^n \rightarrow \Sigma$
 - $\sigma_i(t + 1) = \Phi(\prod_{j \in N(i)} \sigma_j(t))$

Cellular automata – definition

- semantics is defined by a state transition system (automaton)
- a configuration is determined as the current state of all cells
- state update (discrete transition event) is defined by (synchronous) application of local rule to each cell of the source configuration
- automaton is deterministic
- finite grid implies finite number of configurations

Cellular automata – example

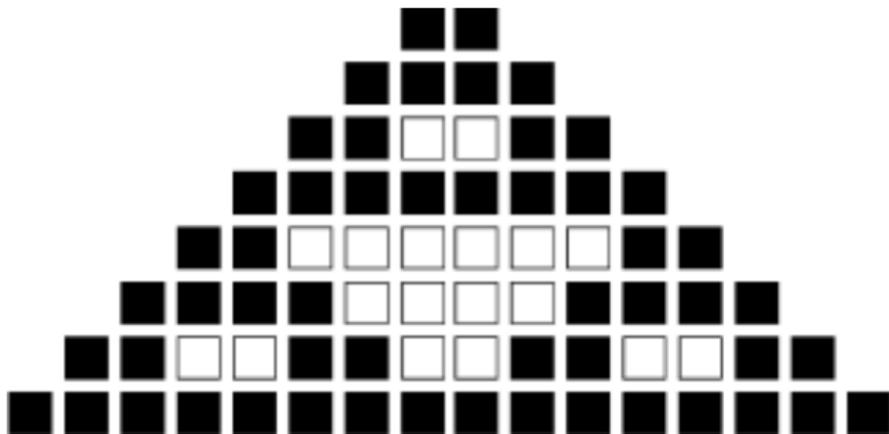
Example of a state update on a 1D grid

$c_{i-1}(t-1)$	$c_i(t-1)$	$c_{i+1}(t-1)$	$c_i(t)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



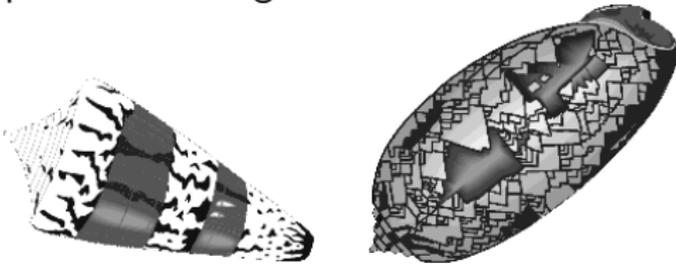
Cellular automata – example

Example of dynamics on a 1D grid



Cellular automata – application in biology

- models in developmental biology
- pattern forming simulation



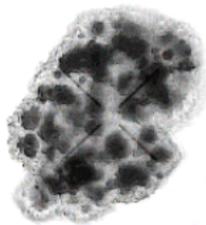
- modeling and simulation of population models (e.g., sheeps and wolfs)

Cellular automata – example

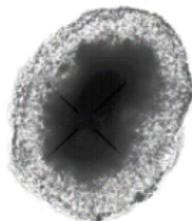
Retinal Cell Development in Chicken Embryo



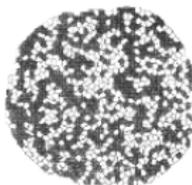
(a)



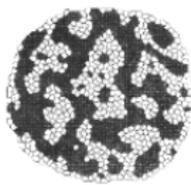
(b)



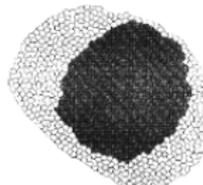
(c)



(a)



(b)



(c)

- light cells – neural retinal cells
- dark cells – pigmented retinal cells
- dynamics – (a) 10 hours, (b) 40 hours, (c) 72 hours

Free Tool Support

- CellLab – library for cellular automata programming
<http://www.fourmilab.ch/cellab/>
- NetLogo – multi-agent programmable modeling environment
 - education platform for agent-based modelling
 - large library of models
 - many extensions (continuous and stochastic simulation)
 - <http://ccl.northwestern.edu/netlogo/>

Literature

-  Deutsch, A. & Dormann, S. *Cellular Automaton Modeling of Biological Pattern Formation*. Modeling and Simulation in Science, Engineering and Technology. Springer-Verlag, 2005.
-  Ermentrout G.B. & Edelstein-Keshet L. *Cellular Automata Approaches to Biological Modeling*. J. theor. Biol. (1993), 160:97-133
-  M. Alber, M. Kiskowski, J. Glazier & Y. Jiang. *On cellular automaton approaches to modeling biological cells.*, 2002.