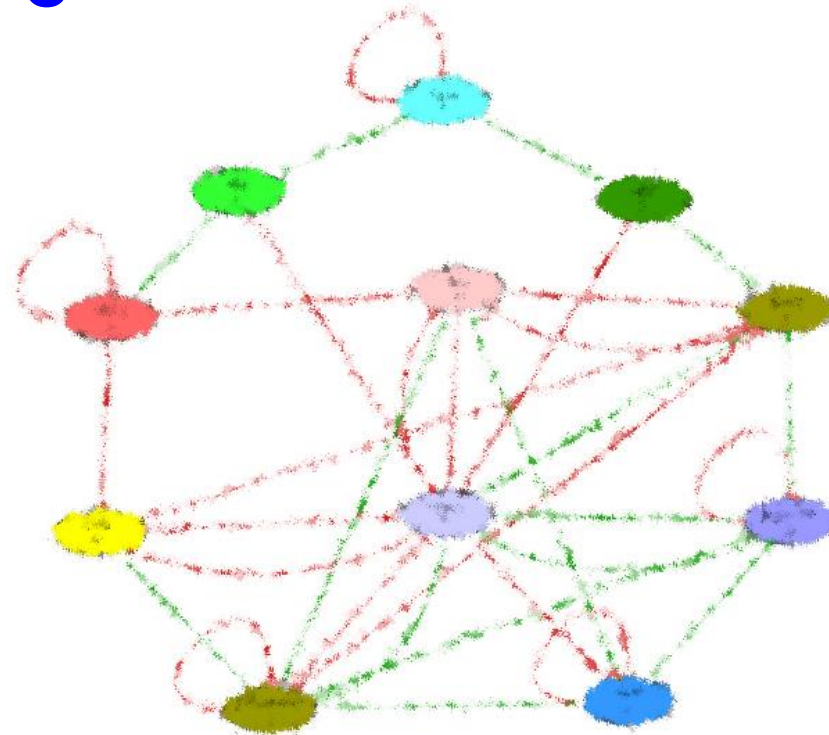


Workshop:

BioNSi - Biological Network Simulation Tool



Amir Rubinstein
CS @ TAU

Outline

- **Part I** – Basic notions:
 - Modeling and simulation
 - Crash intro to **Graph Theory**
- **Part II** – a **Boolean** model for **regulatory network** simulation
- **Part III** - **BioNSi** tool hands-on

Mathematical **Models** and Computer **Simulation**

- **Simulation**: an imitation of how a real-world process or system operates over time



A **computer simulation** (aka *in silico* experiments) is a simulation run on computers.

- In biology, computer simulation is used to replace / complement some **tedious** and **costly** lab experiments.

It enables conducting numerous “experiments” under various conditions, in a scale that is **infeasible experimentally**.

Mathematical **Models** and **Simulation** (cont.)

- Running a computer simulation requires constructing a **mathematical model** - some formal representation of the system using, e.g., **equations** and **algorithms**.

- A model is an **abstraction** of reality.

A valuable model should capture the relevant aspects of the system, with the appropriate level of detail.

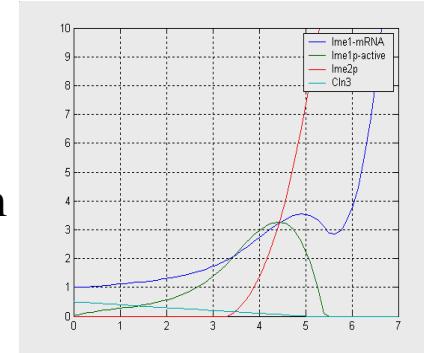
- Advantages of using math. models and simulation in biology:
 - 1) **Descriptive**: forces clarity of expression and precision in describing systems / processes / hypotheses
 - 2) **Analytic**: promotes understanding and provides insights
 - 3) **Predictive**: enables predictions regarding the behavior of the system under various conditions

Quantitative vs. Qualitative Models

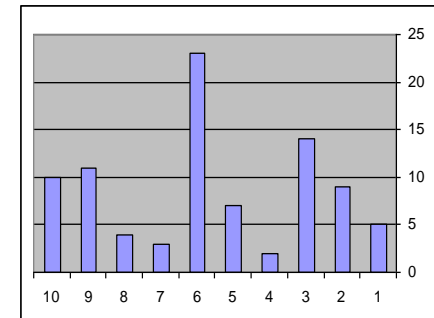
- **Qualitative*** models predict trends, types of dynamics, and general properties, such as:
 - Robustness to mutations
 - Stability against perturbations (e.g., change in a cell's conditions)
 - Fail-safety
 - Conditions for cyclic behavior
- **Quantitative**** models predict specific values that can be compared to actual experimental data, such as:
 - Kinetics
 - Concentrations
 - Expression levels

Continuous vs. Discrete Models

- **Continuous***: infinitely divisible (*e.g. reals*)
 - Usually in the form of **differential equations**
 - Provide a high resolution description of the biological system



- **Discrete**** : made of distinct, indivisible units (*e.g. integers*)
 - Discrete **time**: time progresses in discrete steps (clock tics)
 - Discrete **space**: biological quantities are discrete
- Discrete models tend to be simpler, more computationally efficiency, and require less detailed biological data.

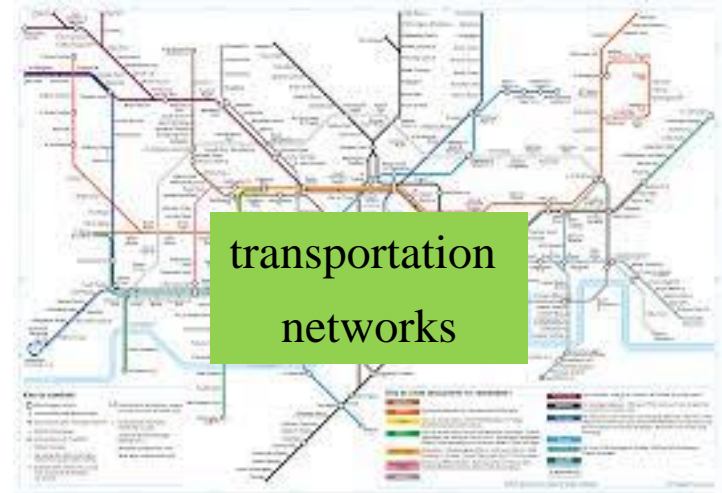
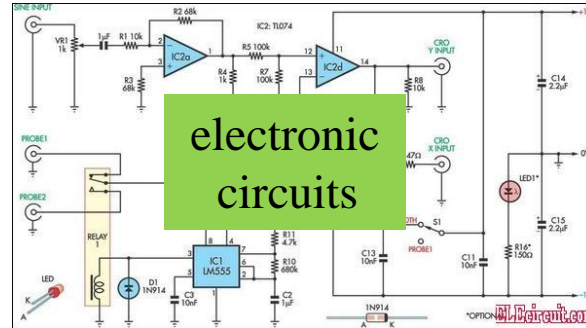
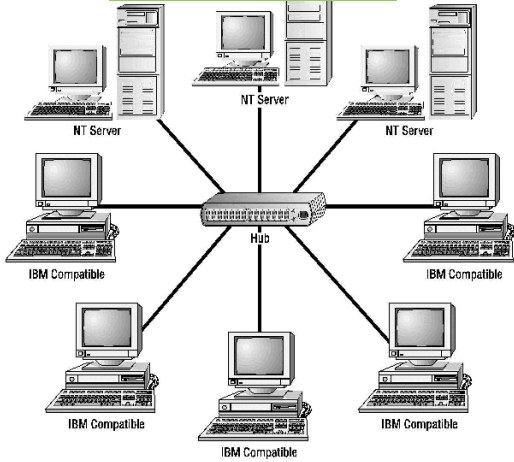


Continuous vs. Discrete Models (cont.)

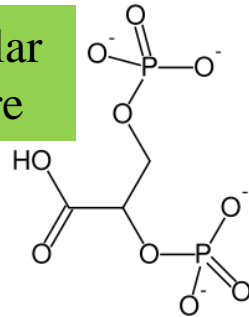
- What about biological quantities? Are they discrete or continuous?
e.g.: interactions, concentrations, reaction times, signals, etc.
- Whether these are really continuous or discrete is a physical question, or maybe even a philosophical one.
But anyway, **modeling does not have to conform with the nature of the modeled entity.**

Networks

computer networks

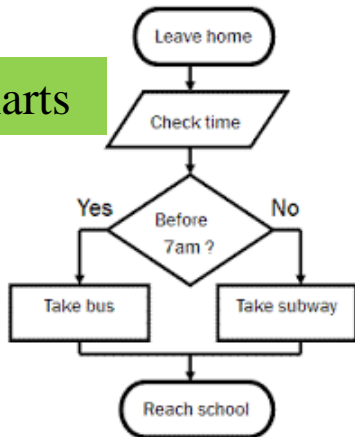


molecular structure



and...
biological networks!

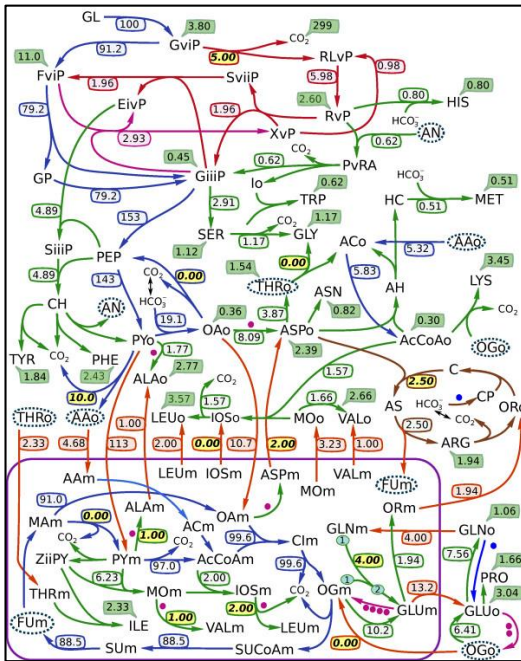
flow charts



social networks

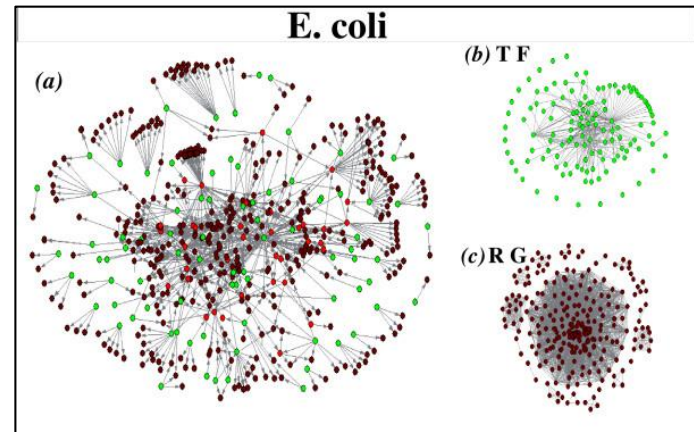


Biological Networks - Examples



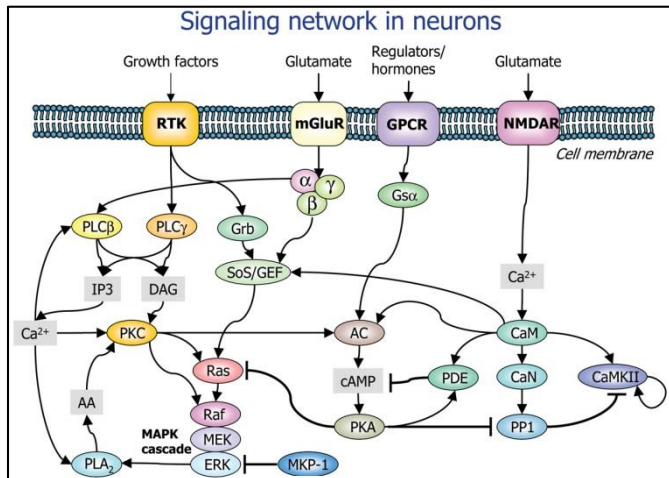
Metabolic and amino acid biosynthesis pathways of yeast

[Schryer et al., BMC Systems Biology. \(2011\)](#)



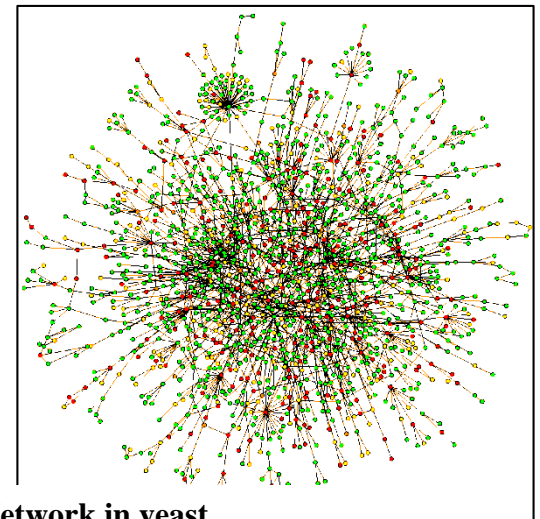
E. coli transcriptional regulatory network

[Guzmán-Vargas et al., BMC Systems Biology \(2008\)](#)



Signaling network in neurons

[Klipp et al., BMC neuroscience \(2006\).](#)



The PPI Network in yeast

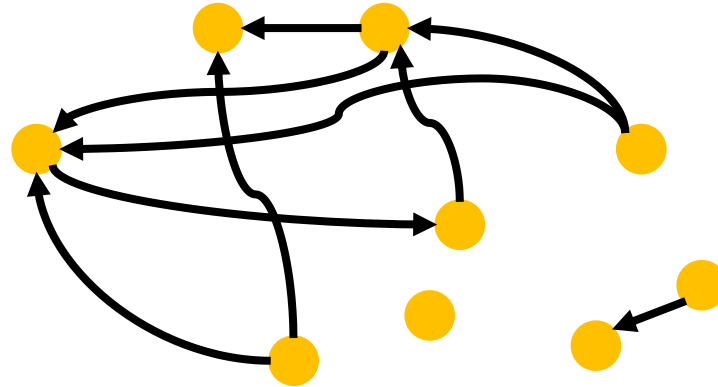
[Jeong et al., Nature \(2001\)](#)

Visualization and Analysis of Biological Networks

- There are various **tools** and software packages for the visualization of networks.
- When the networks are large and dense, it is sometimes difficult to **extract meaningful information** from their visual representation.
- Computational analyses of networks enable valuable insights into their structure, properties and behavior.
- The mathematical structure used to model networks is called a **graph**. **Graph theory** deals with studying various aspects of graphs.

Introduction to Graph Theory

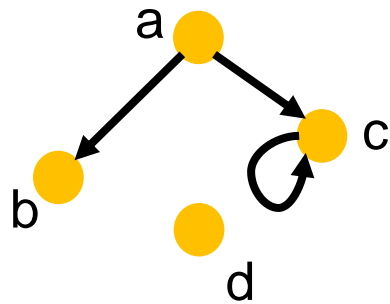
- A **graph** is a set of interactions, or relationships, between pairs of objects.



- The objects are called **nodes**^{*}, and the interactions are termed **edges**^{**}.
- If the edges have **directions**, the graph is called a **directed graph** (or **digraph**). Otherwise it is an **undirected** graph.

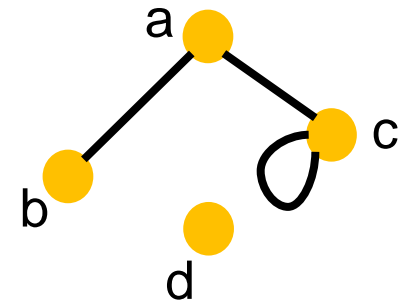
Graphs – More Formally

- A graph G is a pair $G = (V, E)$ where:
 - V is a set of element (called nodes)
 - E is a set of **pairs** from V (called edges)



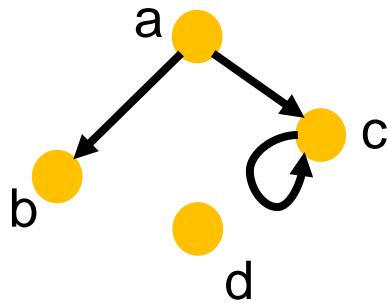
$$V = \{a, b, c, d\}$$

$$E = \{(a, b), (a, c), (c, c)\}$$



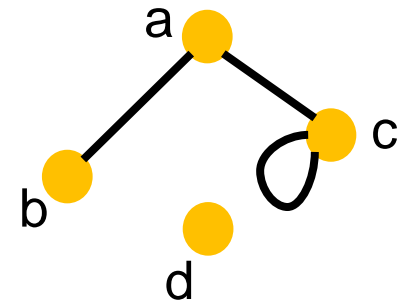
- In an undirected graph we **ignore the order** of nodes in an edge.

Basic Notions



$$V = \{a, b, c, d\}$$

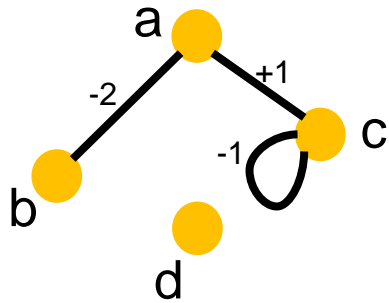
$$E = \{(a, b), (a, c), (c, c)\}$$



- Common notations: $|V| = n$, $|E| = m$
- neighboring / adjacent / connected nodes
- neighborhood of a node
- degree of a node (for directed graphs: **in**-degree and **out**-degree)
- a loop

Weighted Graphs

- A **weighted graph** is a graph in which edges are assigned **values**, called weights.



- What can weights resemble in a biological context?

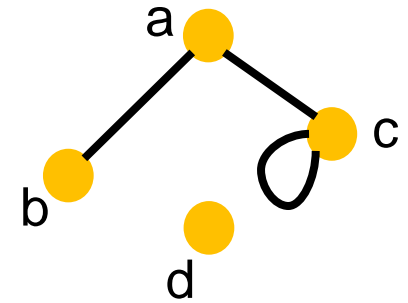
Paths and Connectivity

- A **path** p in a graph $G = (V, E)$ is a sequence of nodes

$$p = (v_1, v_2, \dots, v_k)$$

Such that $(v_i, v_{i+1}) \in E$ for every $1 \leq i < k$

If $v_1 = v_k$ then the path is called a **cycle**.



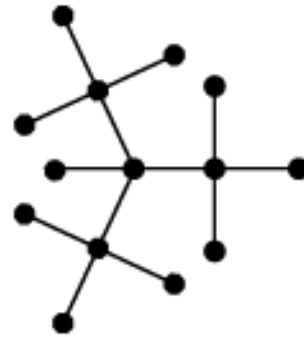
- The **length** or **weight of a path** p is the number of edges in it.
In **weighted** graphs, this is the **sum of weights** along the path.
- A graph is **connected** if there is a path from every node to every other node (in other words every node is reachable from any other node)

Special Graphs

- Tree

An undirected graph that is:

- connected
- acyclic (= contains no cycles)



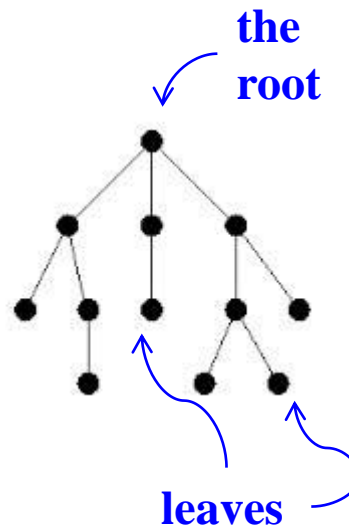
- Rooted tree

A tree with a special node called **root**.

This defines a **hierarchy**:

- **parents** and ancestors
- **children** and descendants

A **leaf** is a node with no children.



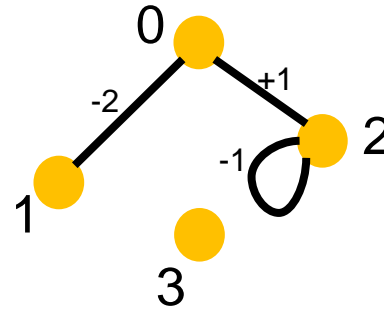
In CS, rooted trees grow downwards...

*(A full binary tree with 16 leaves.
Courtesy of Dr. Shlomit Pinter,
photo taken in Kenya, 2005)*

Graph Representation

- One simple way to represent a graph is a **matrix of adjacencies** (there are additional ways that we will not discuss).

$$G = \begin{bmatrix} [0, -2, 1, 0], \\ [-2, 0, 0, 0], \\ [1, 0, -1, 0], \\ [0, 0, 0, 0] \end{bmatrix}$$



- For example, $G[1][0] == -2$, and $G[3][1] == 0$.

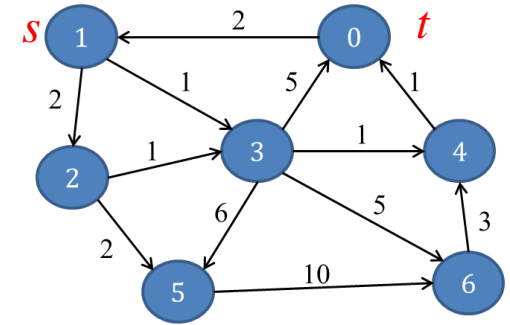
The Bigger Picture

- **Graph theory** and **graph algorithms** are very central within CS.

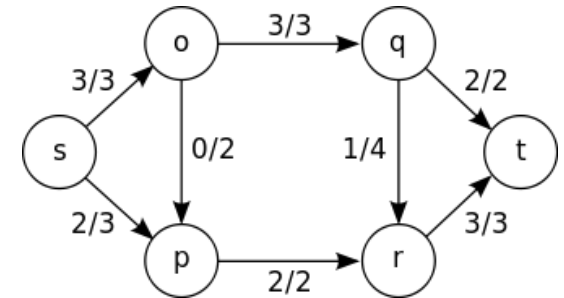
Computational biologists use graph theory to study properties of biological networks, and graph algorithms to solve biological problems (some examples next).

Common Problems in Graph Theory

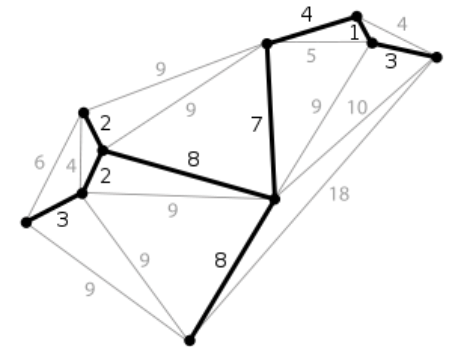
- The **shortest path** problem: find a path from s to t , whose “cost” is minimal.



- The **maximal flow** problem: find a maximum feasible flow from s to t . (weights are flow capacities).



- The **spanning tree** problem: find a subgraph that is a tree and connects all the vertices, with minimal total weight.



- You may also want to check out these two famous topics, related to graph theory: the [7 bridges of Königsberg](#), and the [4 color theorem](#).

Outline

- **Part I** – Basic notions:
 - Modeling and simulation
 - Crash intro to Graph Theory
- **Part II** – a **Boolean** model for **regulatory network** simulation
- **Part III** - **BioNSi** tool hands-on

Boolean Model for Regulatory Networks

- Boolean - 0/1
- A simple case of a discrete model
- Qualitative

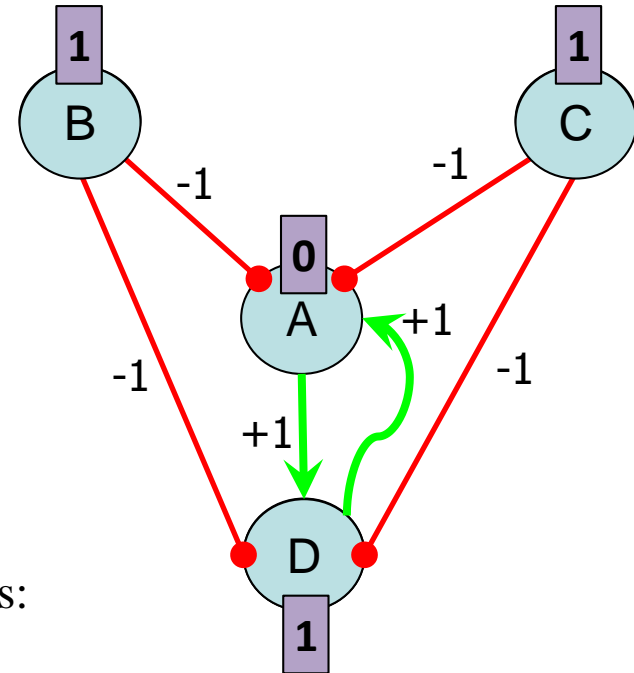
Based on the paper:

[The yeast cell-cycle network is robustly designed,](#)

Li et. al., PNAS 2004

The Boolean Model – User Input

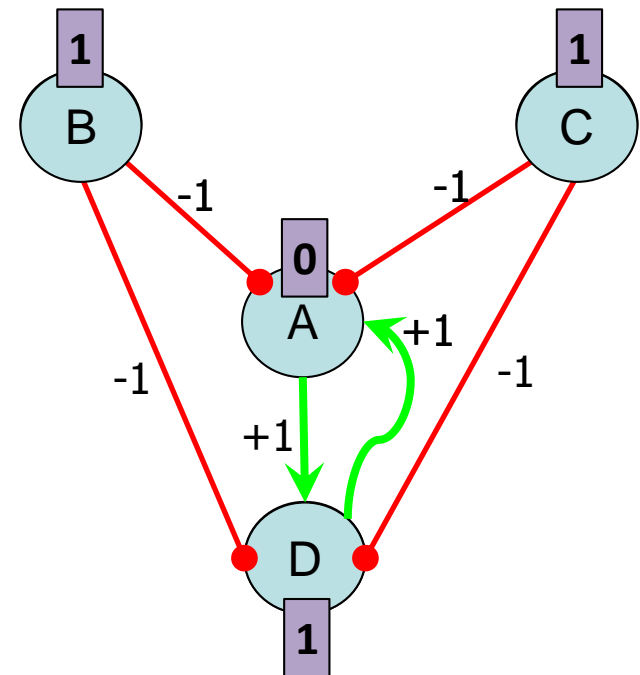
- The model consists of a graph with states.
- Nodes: can represent proteins, mRNA, nutrients, cellular events (*e.g.* mitosis), external signals (*e.g.* light, injected hormone)
- Can assume **state 0** (non active) or **1** (active)
- A **vector** of the network is a sequence of all nodes' states: [0,1,1,1]
- Each node is given an **initial state**. So the network has an **initial vector**.



- Edges: regulation effects
weighted
(+ activation)
(- inhibition)

The Boolean Model - Simulation

- Time is discrete (time steps = 1,2,3,...)
- A transition function determines the states of nodes in the next time step in a synchronous fashion.
It moves the system into the next vector.
- Transition function is applied repeatedly, until one of two options:
 - Steady state (aka "fixed point")
2 consecutive identical vectors
 - Infinite loop
2 **non**-consecutive identical vectors



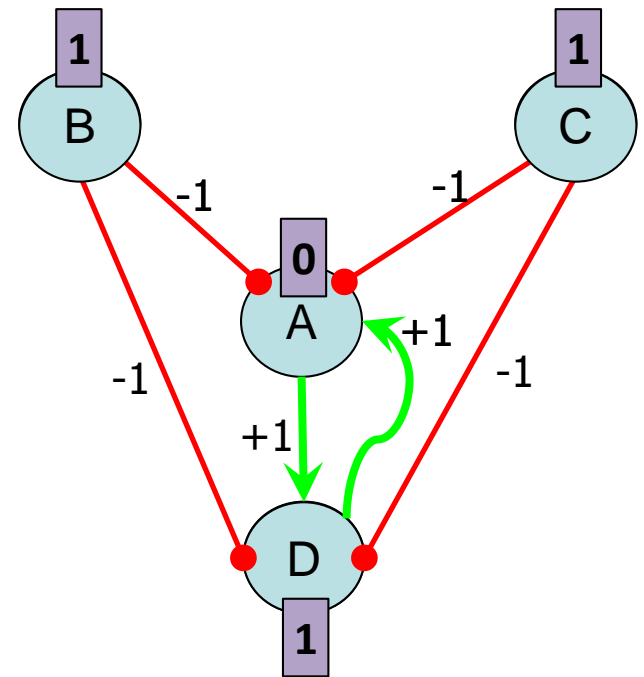
The Boolean Model – Transition Function

- Transition function:

Sums the effects on each node, caused by all its incoming edges.

$$\sigma_i(t) = \sum_j w(j, i) \cdot s_j(t)$$

weight of edge (j, i) state of node j at time step t

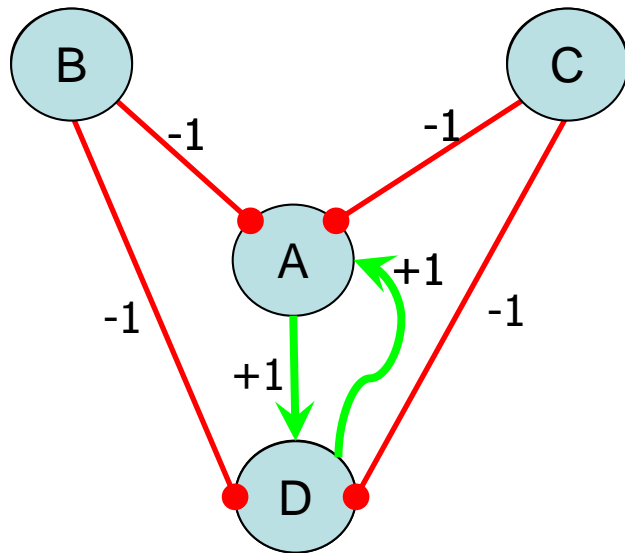


- The state update:

$$s_i(t + 1) = \begin{cases} 1 & \text{if } \sigma_i(t) > 0 \\ 0 & \text{if } \sigma_i(t) < 0 \\ s_i(t) & \text{else} \end{cases}$$

Example 1

- Let's see what happens to node A at t_2 :



	A	B	C	D
t_1 :	1	1	0	0

+

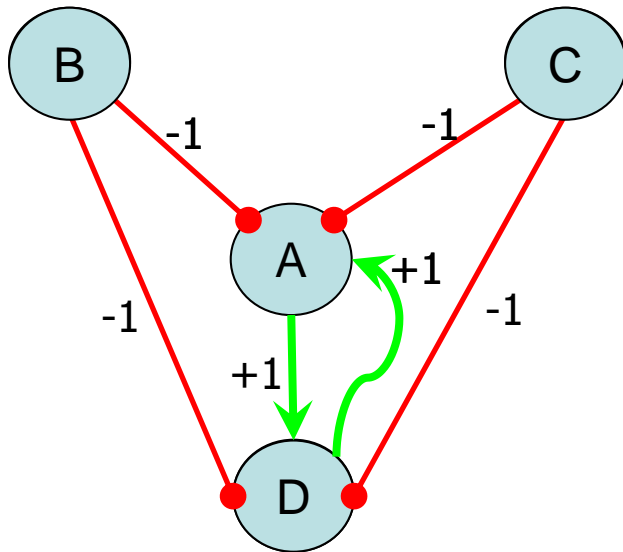
+

= -1

t_2 :	0			
---------	---	--	--	--

Example 2

- Let's see what happens to node A at t_2 :



	A	B	C	D
t_1 :	1	0	0	1

+

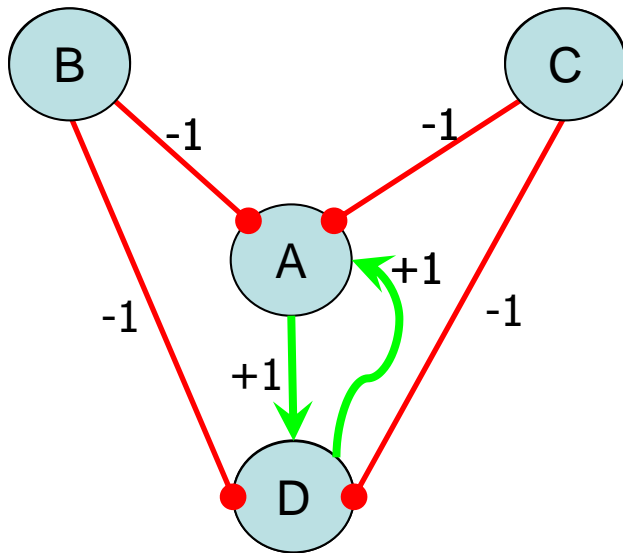
+

= +1

t_2 :	1			
---------	---	--	--	--

Example 3

- Let's see what happens to node A at t_2 :



	A	B	C	D
t_1 :	1	0	1	1

+

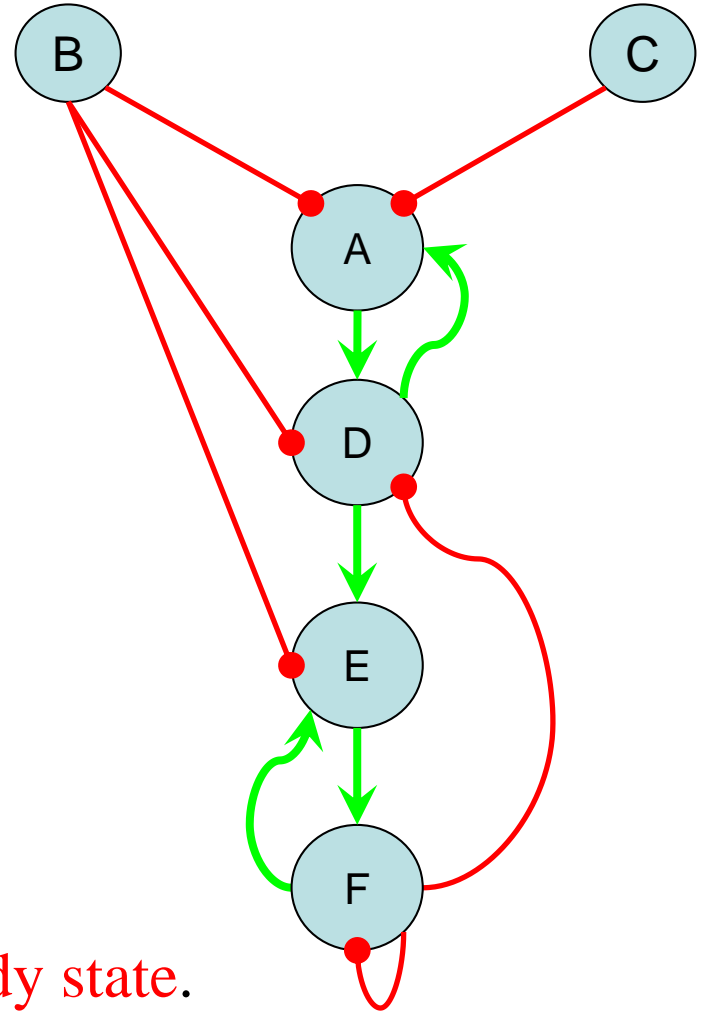
+

= 0

t_2 :	1			
---------	---	--	--	--

A Simulation - Full Example

	A	B	C	D	E	F
t_1 :	1	0	1	0	0	0
t_2 :	0	0	1	1	0	0
t_3 :	0	0	1	1	1	0
t_4 :	0	0	1	1	1	1
t_5 :	0	0	1	0	1	1
t_6 :	0	0	1	0	1	1



Vectors in t_5 and t_6 are identical \rightarrow **steady state**.

Can the system get out of this steady-state in the future?

Exercise - Loops

Give an example for a network and initial vector that yield an **infinite loop**.

Hint: 2 nodes are enough.

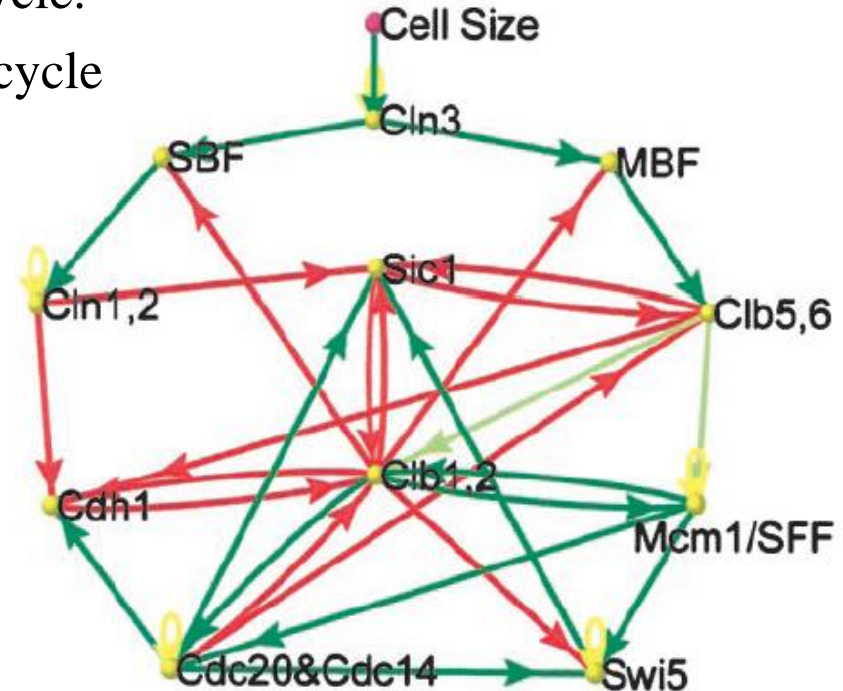
Case Study: The Cell-Cycle in Yeast

"The yeast cell-cycle network is robustly designed", Li et. al., PNAS 2004

- 11 nodes – main regulators of yeast cell-cycle.
- "Cell Size" is the signal for entry into cell-cycle
- Each node can be either 0/1.

Red/yellow edges: weight = -1

Green edges: weight = +1



- Simulation is executed on all possible initial vectors.
How many?
How many potential fixed points?

Yeast Cell-Cycle Simulation **Fixed Points**

- No initial vector yields a loop.
- Out of $2^{11} = 2048$ potential steady states, only 7 are reached !

Table 1. The fixed points of the cell-cycle network

Basin size	Cln3	MBF	SBF	Cln1,2	Cdh1	Swi5	Cdc20	Clb5,6	Sic1	Clb1,2	Mcm1
1,764	0	0	0	0	1	0	0	0	1	0	0
151	0	0	1	1	0	0	0	0	0	0	0
109	0	1	0	0	1	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	1	0	0
7	0	1	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0

The **main "attractor"**:
this steady state attracts
~86% of initial states.

A Complete Cell-Cycle Simulation

- Start with a vector representing **stationary G₁** condition but with Cln3=1 (signal to initiate cell cycle).

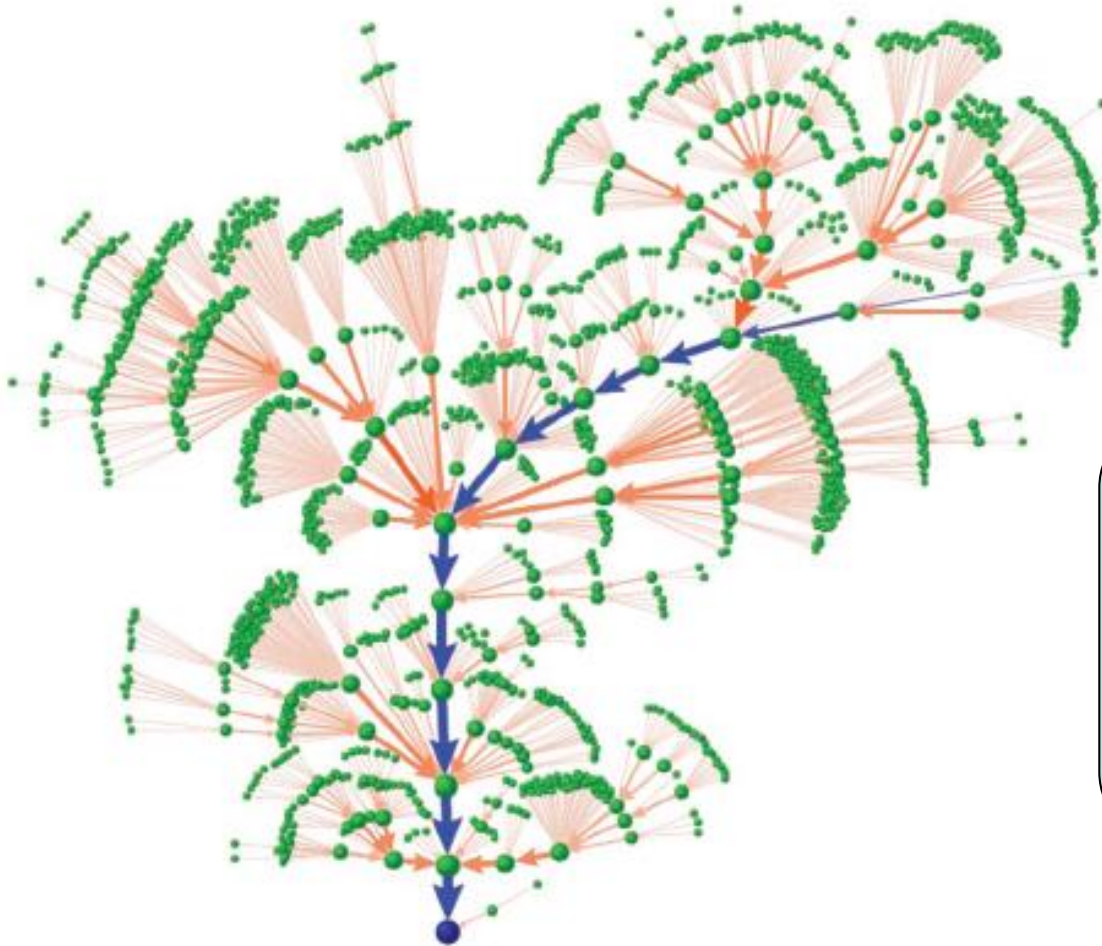
Time	Cln3	MBF	SBF	Cln1,2	Cdh1	Swi5	Cdc20 and Cdc14	Clb5,6	Slc1	Clb1,2	Mcm1/SFF	Phase
1	1	0	0	0	1	0	0	0	1	0	0	START
2	0	1	1	0	1	0	0	0	1	0	0	G ₁
3	0	1	1	1	1	0	0	0	1	0	0	G ₁
4	0	1	1	1	0	0	0	0	0	0	0	G ₁
5	0	1	1	1	0	0	0	1	0	0	0	S
6	0	1	1	1	0	0	0	1	0	1	1	G ₂
7	0	0	0	1	0	0	1	1	0	1	1	M
8	0	0	0	0	0	1	1	0	0	1	1	M
9	0	0	0	0	0	1	1	0	1	1	1	M
10	0	0	0	0	0	1	1	0	1	0	1	M
11	0	0	0	0	1	1	1	0	1	0	0	M
12	0	0	0	0	1	1	0	0	1	0	0	G ₁
13	0	0	0	0	1	0	0	0	1	0	0	Stationary G ₁

The right column indicates the cell-cycle phases. Note that the number of time steps in each phase do not reflect its actual duration.

- As indicated in the table, this simulation is compatible with the **cell cycle stages**: $G_1 \rightarrow S \rightarrow G_2 \rightarrow M \rightarrow G_1$ (stationary)

Cell-Cycle in Yeast - Transitions Tree

- Each **node** in the tree represents a **vector**, **edges** represent **transitions** in the simulation.



Transition tree for the
main "attractor"

1764 = 86% of initial vectors.

(there are 6 other, smaller trees)

Cell-Cycle in Yeast - Paper **Conclusions**

- The yeast cell-cycle is **stable**.

Computational observation: with high probability, **changes to the initial vectors** yield the same fixed point.

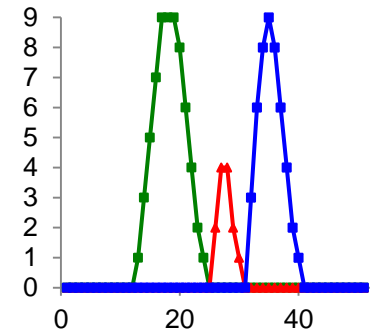
- The yeast cell-cycle is **robust**.

Computational observation: with high probability, **small changes in the network structure** (insert/delete node, change edge) will not harm cell cycle behavior.

Extensions to the Boolean Model

Extension 1: Discrete State Space

- Instead of 0/1, nodes can now assume states between $0, \dots, U$ (e.g. $U=9$)
- $U=1$ is the special case of the Boolean model we saw



- Transition function changes accordingly:

$$s_i(t + 1) = \begin{cases} \min(U, s_i(t) + 1) & \text{if } \sigma_i(t) > 0 \\ \max(0, s_i(t) - 1) & \text{if } \sigma_i(t) < 0 \\ s_i(t) & \text{else} \end{cases}$$

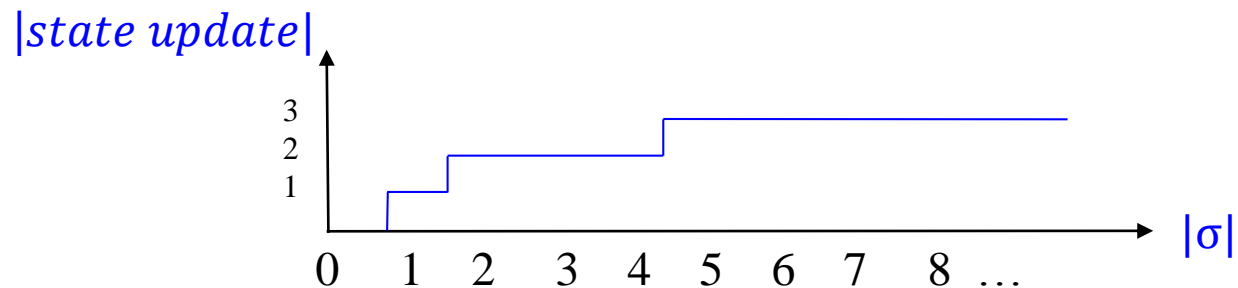
$$\left(\sigma_i(t) = \sum_j w(j, i) \cdot s_j(t) \right)$$

Extension 2: State Update Function

- States change by ± 1 no matter what.
- We may prefer the change to be proportional to $\sigma_i(t)$.

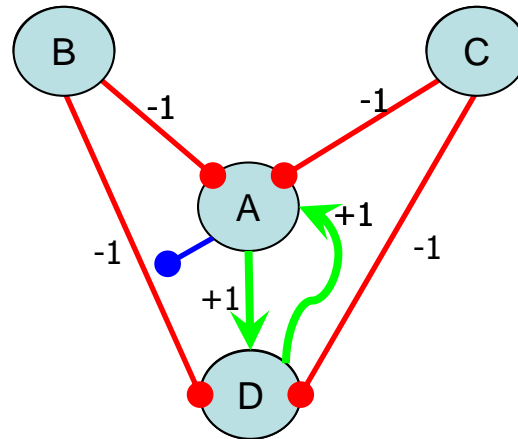
$$s_i(t + 1) = \begin{cases} \min(U, s_i(t) + ?) & \text{if } \sigma_i(t) > 0 \\ \max(0, s_i(t) - ?) & \text{if } \sigma_i(t) < 0 \\ s_i(t) & \text{else} \end{cases}$$

- One reasonable option is a **logarithmic** order update:



Extensions 3, 4, ...

- A new type of interactions called **dependency** edges: **nodes may block or enable other edges**

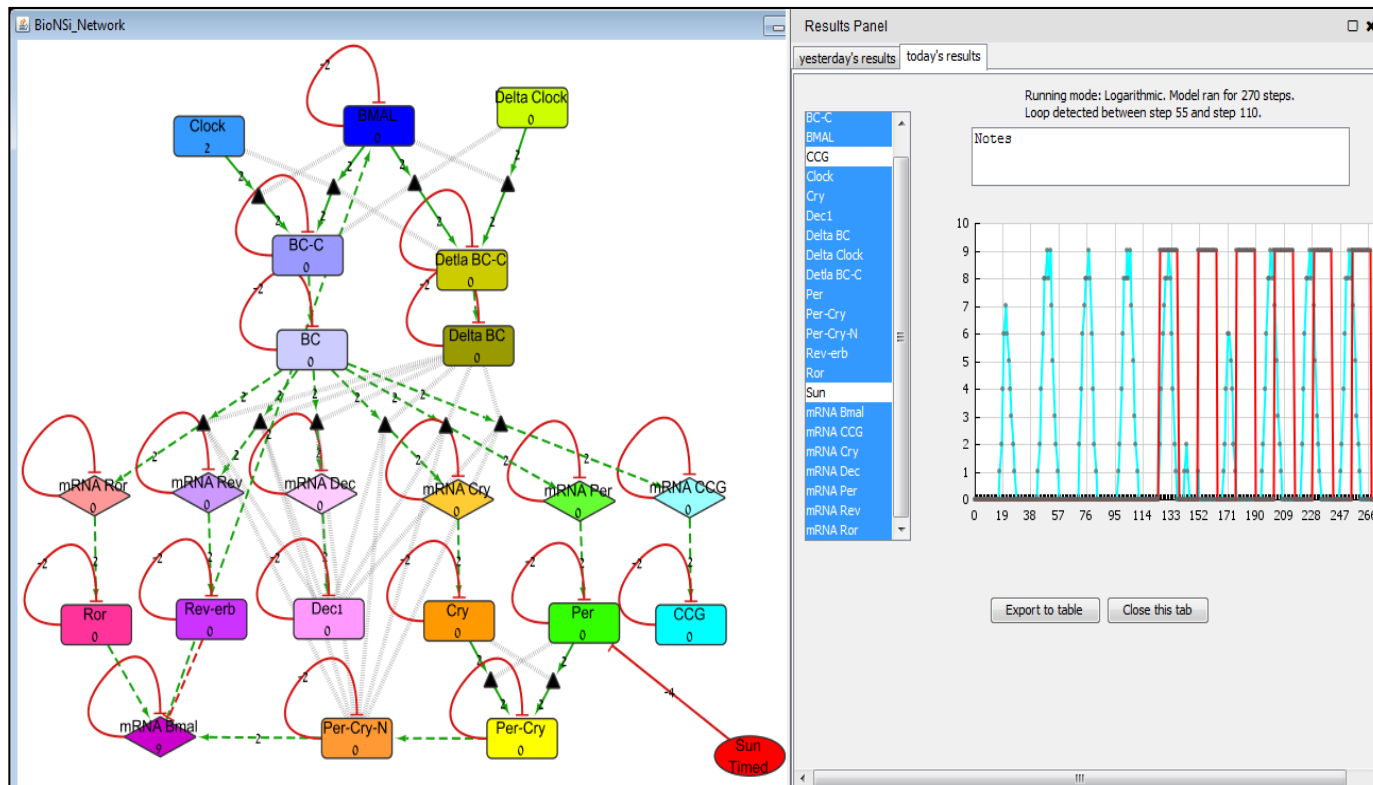


If $A > 0$ edge $B \rightarrow D$ is blocked

- **Delays** on edges
 - The effect of node a on node b will occur in later steps

BioNSi – Biological Network Simulator

- Cytoscape is an open source software for visualization and analysis of networks and pathways (www.cytoscape.org).
- BioNSi is a plugin (app) of Cytoscape (<http://bionsi.wix.com/bionsi>)
It extends the Boolean model in several ways, including those mentioned.



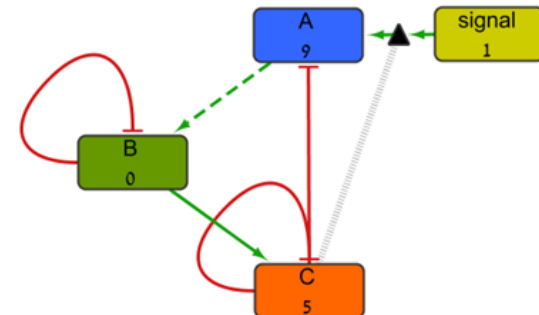
The circadian clock in mammals in a day-night regime, in *BioNSi*

Outline

- **Part I** – Basic notions:
 - Modeling and simulation
 - Crash intro to Graph Theory
- **Part II** – a Boolean model for regulatory network simulation
- **Part III** - **BioNSi** tool hands-on

BioNSi Hands-on Session

- Open **Cytoscape** (installed in this lab's computers)
- Go to bionsi.wix.com/bionsi
 - Click **Download**, save the file (BioNSi.jar) on your computer
- In Cytoscape's main menu:
 - click Apps → App Manager → Install from file
 - Choose BioNsi.jar file
 - BioNSi is now installed.
- In BioNSi's website, go to Examples → Toy example
 - Open the pdf and follow the tutorial



Reflection: Constructing the “Right” Model?

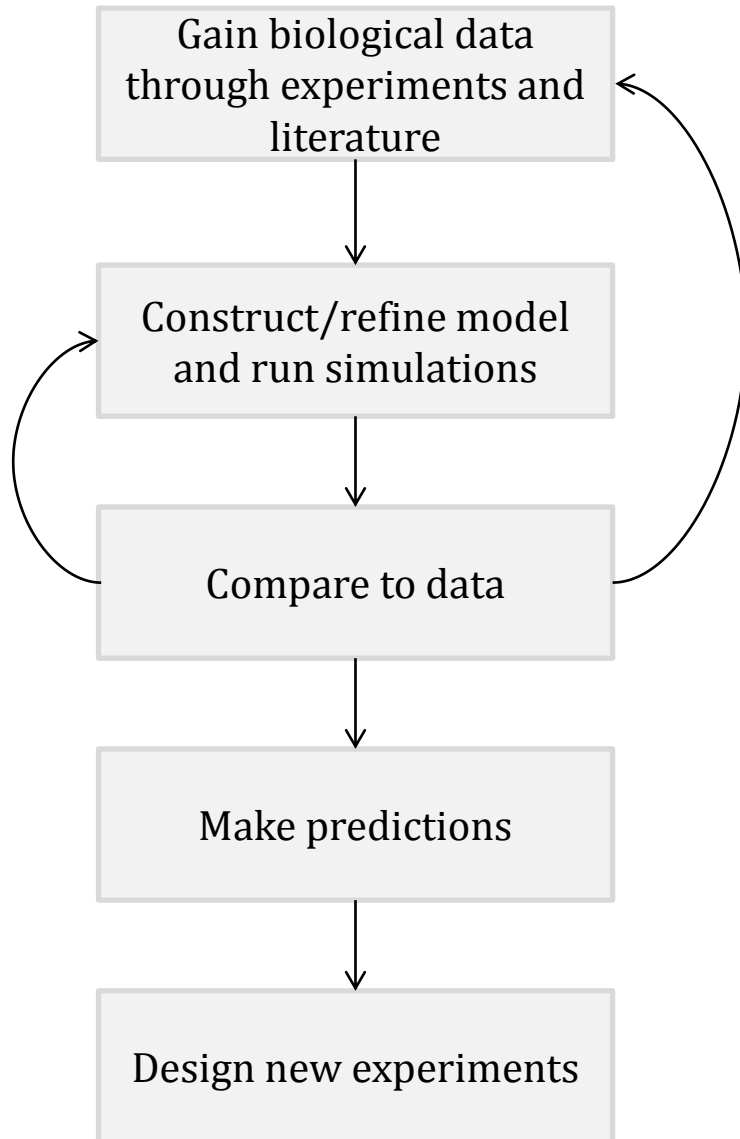


- Main considerations in constructing mathematical models in biology:

Consideration	Meaning	Beware of
Modeling technique	Which model type is best suited for the data and goals of study?	Intractable approaches, too simplified, no sufficient data, etc.
Scope	What elements should be included in the model?	Excluding essential elements or including irrelevant ones
Detail	What level of detail should the model contain?	Too fine- or course-grained models
Parameters	How to set the model parameters appropriately?	Parameter overfitting

- Recall that even “incorrect” models may make correct predictions

Reflection: Iterative Simulation-Experiment Approach

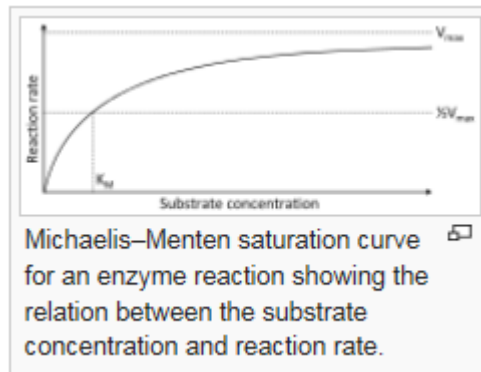


- Using models can point to **gaps in our biological understanding**: a model that fails to recapitulate known biological data reveals where our understanding needs improvement.

Reflection: Discrete Models



- Computer Science is highly biased towards discrete notions, such as graphs (networks), strings (textual sequences), digital images, etc.
- Discrete notions and algorithms are highly underrepresented in **life science curricula**, where **continuous** notions (such as equations over the reals) and probability are taught more widely.



$$v = \frac{d[P]}{dt} = \frac{V_{max}[S]}{K_M + [S]}$$

Advertisement:

Computational Approaches for Life Scientists

Course url: ca4ls.wikidot.com

- A course designed to enrich biologists with computational thinking, and basic **ideas** and **notions** from computer science, **beyond programming and tools**.