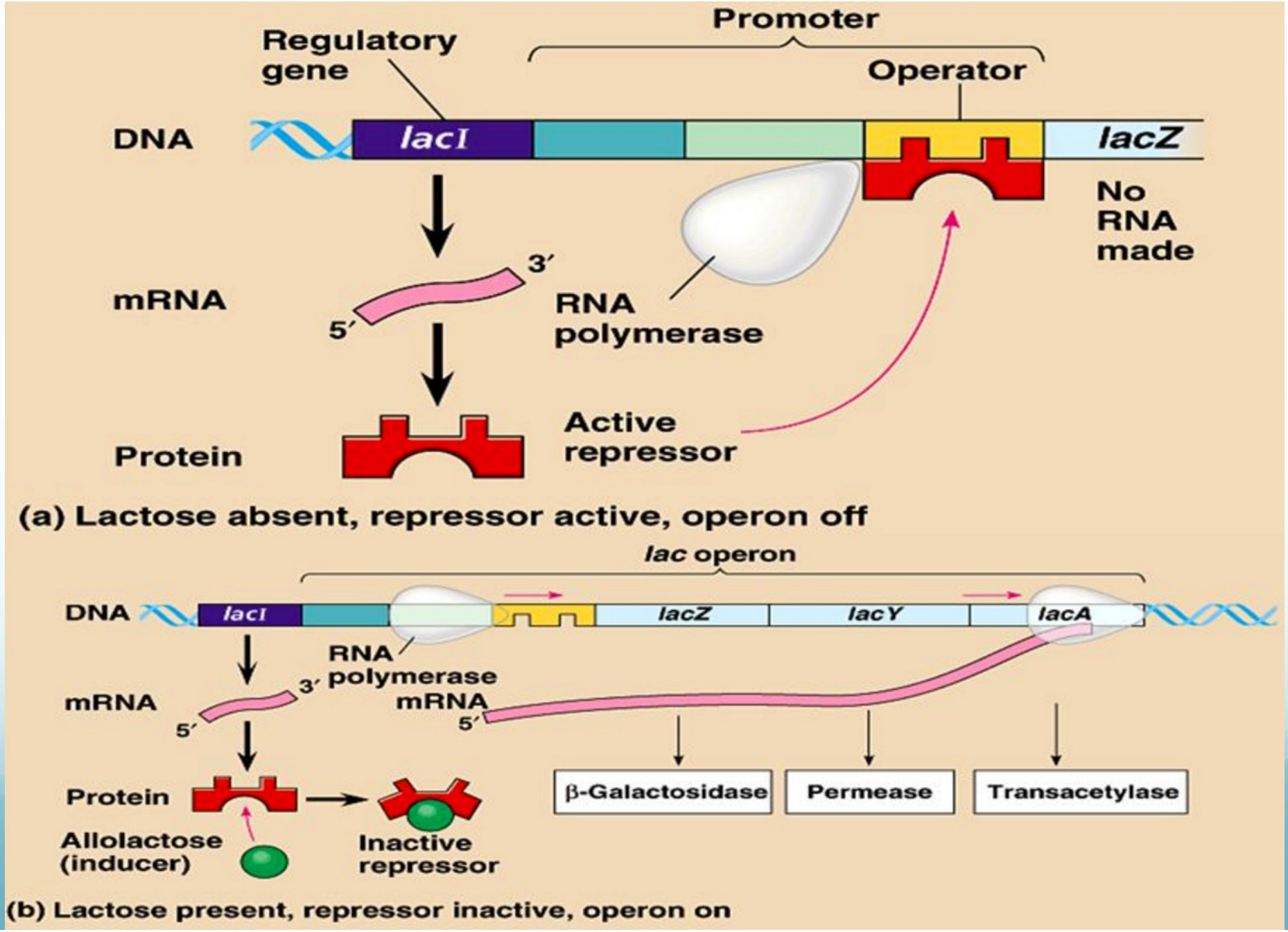
Boolean models of the *lac* operon in *E. coli*

Matthew Macauley
Clemson University

The *lac* operon

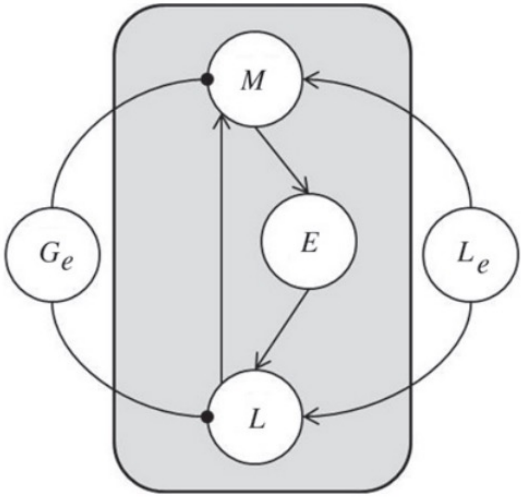


Downsides of an ODE model

- Very mathematically technical.
- Too hard to solve explicitly. Numerical methods are needed.
- MANY experimentally determined “rate constants” (I count 22...)
- Often, these rate constants aren’t known even up to orders of magnitude.

$$
\begin{aligned}
\frac{dM}{dt} &= \alpha_M \frac{1 + K_1 \left(e^{-\mu\tau_M} A_{\tau_M}\right)^n}{K + K_1 \left(e^{-\mu\tau_M} A_{\tau_M}\right)^n} + \Gamma_0 - \widetilde{\gamma_M} M \\
\frac{dB}{dt} &= \alpha_B e^{-\mu\tau_B} M_{\tau_B} - \widetilde{\gamma_B} B \\
\frac{dA}{dt} &= \alpha_A B \frac{L}{K_L + L} - \beta_A B \frac{A}{K_A + A} - \widetilde{\gamma_A} A \\
\frac{dP}{dt} &= \alpha_P e^{-\mu(\tau_B + \tau_P)} M_{\tau_B + \tau_P} - \widetilde{\gamma_P} P \\
\frac{dL}{dt} &= \alpha_L P \frac{L_e}{K_{L_e} + L_e} - \beta_{L_1} P \frac{L}{K_{L_1} + L} - \alpha_A B \frac{L}{K_L + L} - \widetilde{\gamma_L} L
\end{aligned}
$$

A Boolean approach



- Let's assume everything is "Boolean" (0 or 1):
  - Gene products are either present or absent
  - Enzyme concentrations are either high or low.
  - The operon is either ON or OFF.

- mRNA is transcribed (M=1) if there is no external glucose (G=0), *and* either internal lactose (L=1) or external lactose (L_e=1) are present.

$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

- The LacY and LacZ gene products (E=1) will be produced if mRNA is available (M=1).

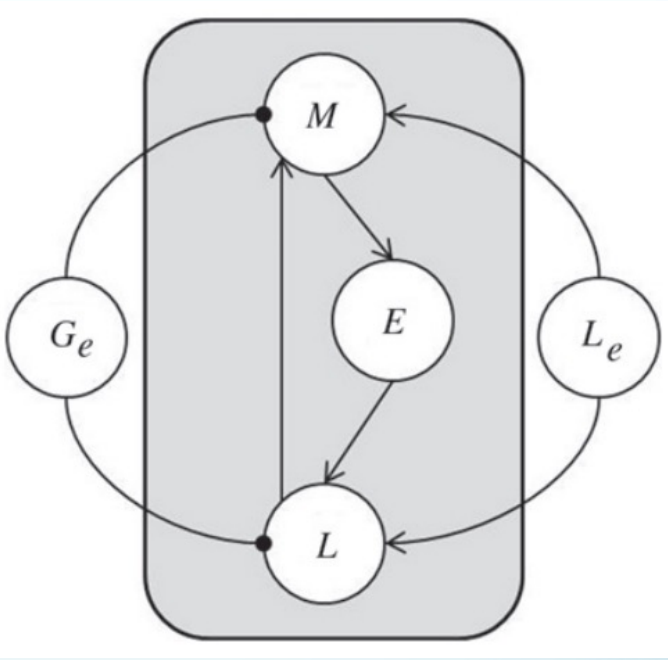
$$x_E(t+1) = f_E(t+1) = M(t)$$

- Lactose will be present in the cell if there is no external glucose (G_e=0), and either of the following holds:
  - ✓ External lactose is present (L_e=1) and *lac* permease (E=1) is available.
  - ✓ Internal lactose is present (L=1), but β–galactosidase is absent (E=0).

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$

Comments on the Boolean model

- We have two "types" of Boolean quantities:
  - mRNA (M), lac gene products (E), and internal lactose (L) are variables.
  - External glucose (G_e) and lactose (L_e) are parameters (constants).
- Variables and parameters are drawn as nodes.
- Interactions can be drawn as signed edges.
- A signed graph called the wiring diagram describes the dependencies of the variables.
- Time is discrete: t = 0, 1, 2, ...



$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

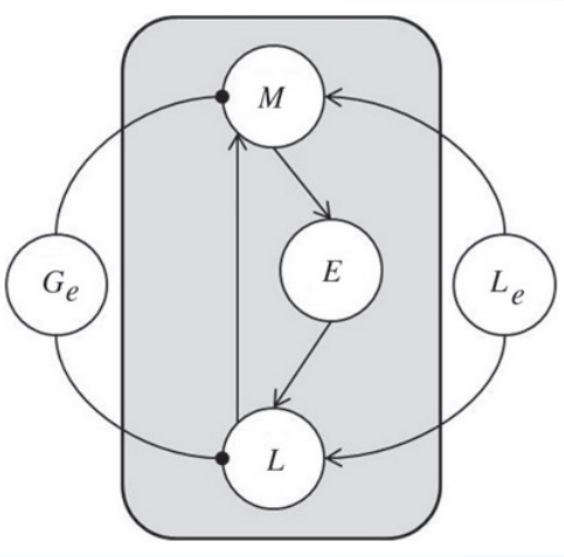
$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$

- Assume that the variables are updated synchronously.

How to analyze a Boolean model

- At the bare minimum, we should expect:
  - Lactose absent => operon OFF.
  - Lactose present, glucose absent => operon ON.
  - Lactose and glucose present => operon OFF.



$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$

- The state space (or phase space) is the directed graph (V, T), where

$$V = \{(x_M, x_E, x_L) : x_i \in \{0,1\}\} \qquad T = \{(x, f(x)) : x \in V\}$$

- We'll draw the state space for all four choices of the parameters:
  - (L_e, G_e) = (0, 0). We hope to end up in a fixed point (0,0,0).
  - (L_e, G_e) = (0, 1). We hope to end up in a fixed point (0,0,0).
  - (L_e, G_e) = (1, 0). We hope to end up in a fixed point (1,1,1).
  - (L_e, G_e) = (1, 1). We hope to end up in a fixed point (0,0,0).

How to analyze a Boolean model

- We can plot the state space using several software packages:
  - The "Cyclone" software package: at http://cyclone.algorun.org/.
  - The BoolNet library in R.

- For Cyclone, we need to convert our logical functions into polynomials.

$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$

- Here is the relationship between Boolean logic and polynomial algebra:

Boolean operations	logical form	polynomial form
AND	$z = x \wedge y$	$z = xy$
OR	$z = x \vee y$	$z = x + y + xy$
NOT	$z = \overline{x}$	$z = 1 + x$

- Everything is modulo 2, so 1+1=0, and 1=-1, and $x^2=x$, and thus x(x+1)=0.

ALGORUN

Cyclone | Run | About | Web API | Help

Cyclone

Calculate Dynamics of a discrete dynamical system using exhaustive search

More Information: https://github.com/PlantSimLab/cyclone

Packaged by: Abdelrahman Hosny

Authors

Reinhard Laubenbacher (PI)
Center for Quantitative Medicine

$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$

input

```
1  MODEL NAME: 3-variable lac operon model
2  SIMULATION NAME: Lactose but not glucose
3  NUMBER OF VARIABLES: 3
4  VARIABLE NAMES: x1 x2 x3
5  NUMBER OF STATES: 2 2 2
6  SPEED OF VARIABLES: 1 1 1
7
8  f1 = 1
9  f2 = x1
10 f3 = x2 + x3 + x2*x3
```

output | visualization

```
1  digraph test {
2  node0 [label=" 0 0 0"];
3  node1 [label=" 0 0 1"];
4  node2 [label=" 0 1 0"];
5  node3 [label=" 0 1 1"];
6  node4 [label=" 1 0 0"];
7  node5 [label=" 1 0 1"];
8  node6 [label=" 1 1 0"];
9  node7 [label=" 1 1 1"];
10 node0 -> node0
11 node1 -> node5
12 node2 -> node1
13 node3 -> node5
14 node4 -> node2
15 node5 -> node7
16 node6 -> node3
17 node7 -> node7
18 }
19
```

Load sample data | Change parameters | Reset computation | RUN COMPUTATION

ALGORUN | Cyclone | Run | About | Web API | Help

Cyclone

Calculate Dynamics of a discrete dynamical system using exhaustive search

More Information: https://github.com/PlantSimLab/cyclone

Packaged by: Abdelrahman Hosny

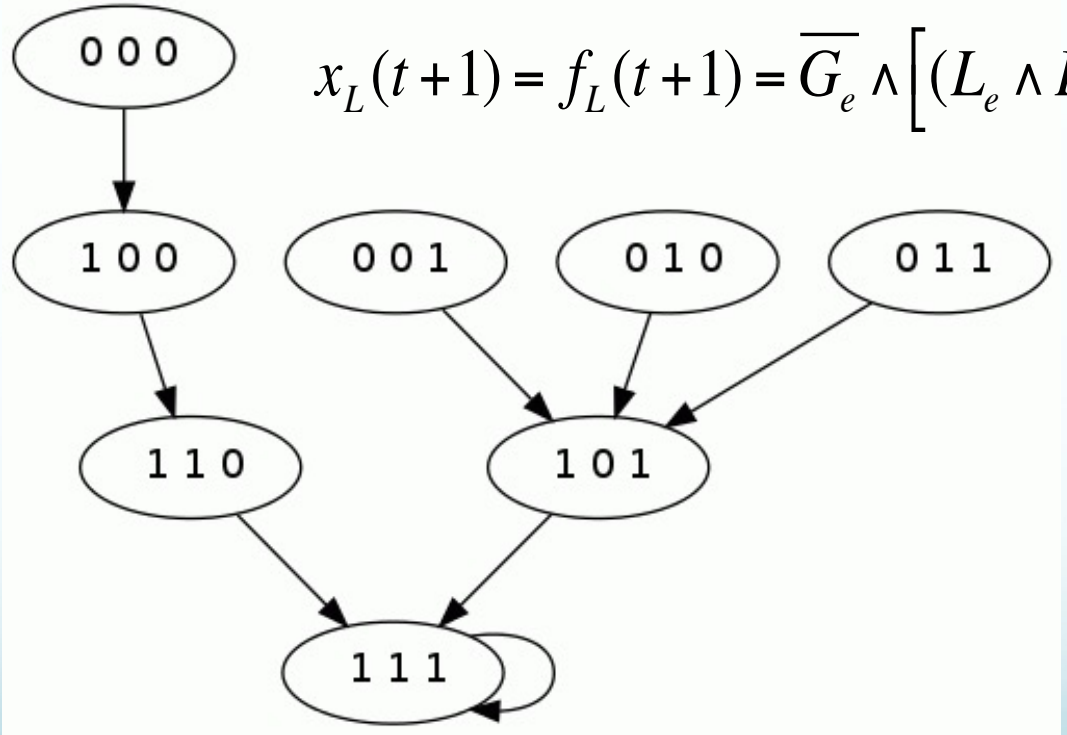
Authors

Reinhard Laubenbacher (PI)
Center for Quantitative Medicine

$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$



State space when (G_e, L_e) = (0, 1). The operon is ON.

ALGORUN | Cyclone | Run | About | Web API | Help

Cyclone

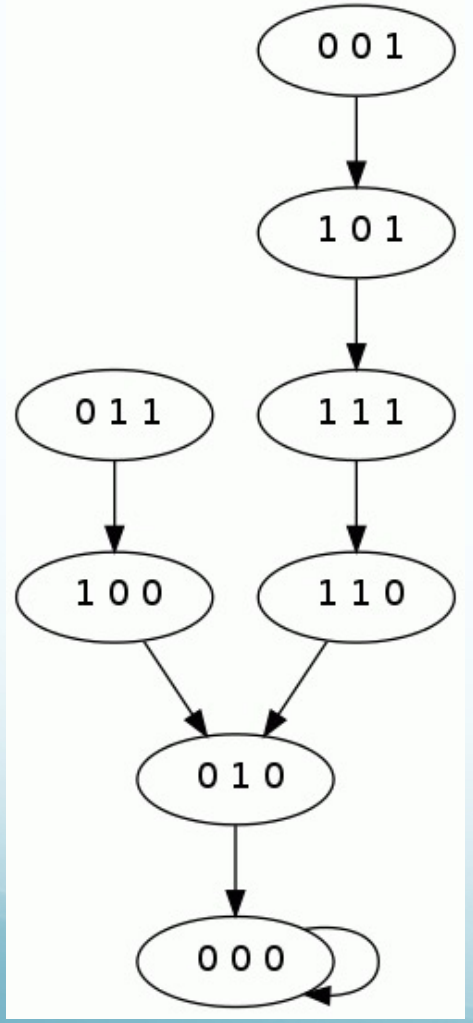
Calculate Dynamics of a discrete dynamical system using exhaustive search

More Information: https://github.com/PlantSimLab/cyclone

Packaged by: Abdelrahman Hosny

Authors

Reinhard Laubenbacher (PI)
Center for Quantitative Medicine



$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$

State space when (G_e, L_e) = (0, 0).

The operon is OFF.

ALGORUN | Cyclone | Run | About | Web API | Help

Cyclone

Calculate Dynamics of a discrete dynamical system using exhaustive search

More Information: https://github.com/PlantSimLab/cyclone

Packaged by: Abdelrahman Hosny

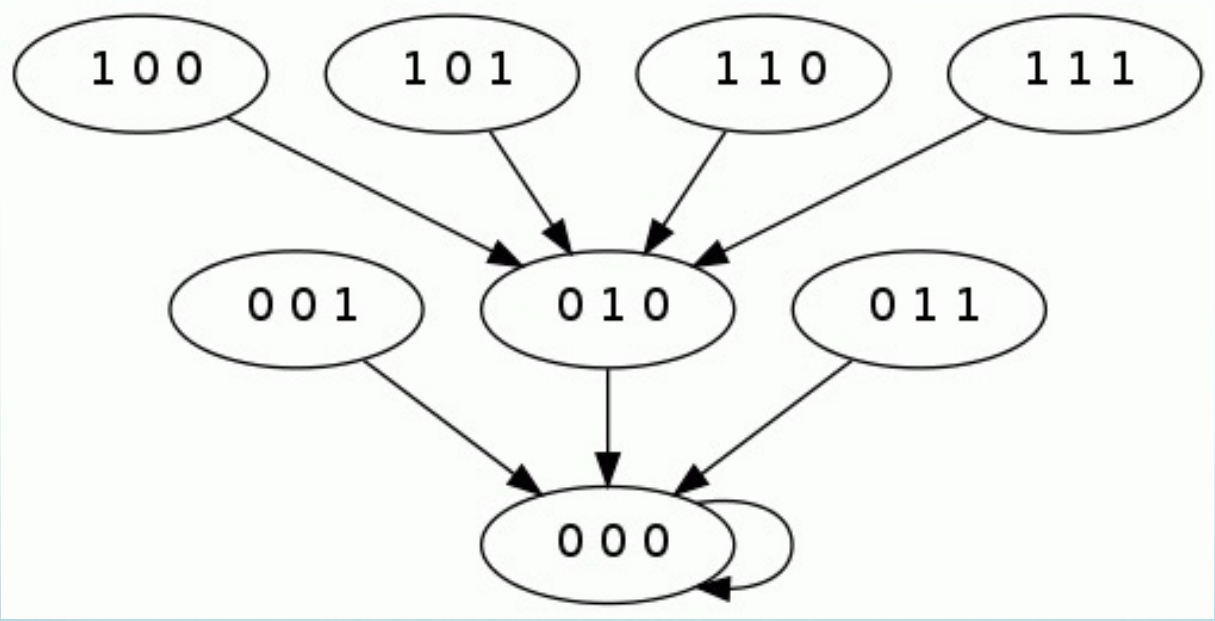
Authors

Reinhard Laubenbacher (PI)
Center for Quantitative Medicine

$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$



State space when (G_e, L_e) = (1, 0). The operon is OFF.

ALGORUN | Cyclone | Run | About | Web API | Help

Cyclone

Calculate Dynamics of a discrete dynamical system using exhaustive search

More Information: https://github.com/PlantSimLab/cyclone

Packaged by: Abdelrahman Hosny

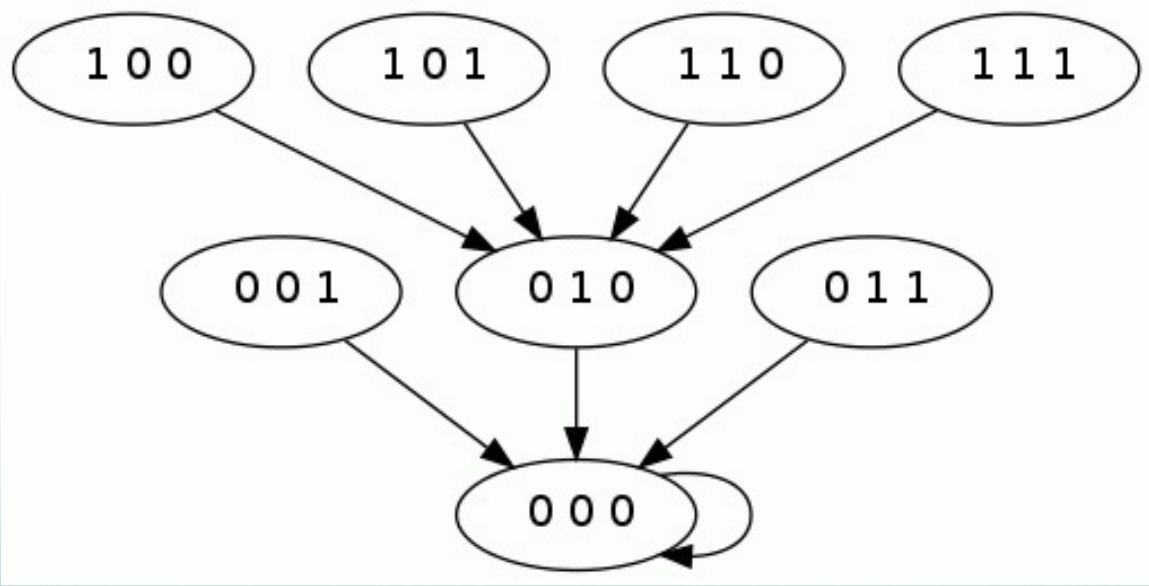
Authors

Reinhard Laubenbacher (PI)
Center for Quantitative Medicine

$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e)$$

$$x_E(t+1) = f_E(t+1) = M(t)$$

$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$



State space when (G_e, L_e) = (1, 1). The operon is OFF.

Summary so far

- Gene regulatory networks consist of a collection of gene products that interact with each other to control a specific cell function.
- Classically, these have been modeled quantitatively with differential equations (continuous models).
- Boolean models take a different approach. They are discrete models that are inherently qualitative.
- The state space graph encodes all the dynamics. The most important features are the fixed points, and a necessary step in model validation is to check that they are biologically meaningful.
- The model of the *lac* operon shown here is a "toy model". Next, we will see more complicated models of the *lac* operon that capture intricate biological features of these systems.
- Modeling with Boolean logic is a relatively new concept, first done in the 1970s. It is a popular research topic in the field of systems biology.

A more refined model

- Our first model only used 3 variables: mRNA (M), enzymes (E), and lactose (L).
- Let's propose a new model with 5 variables:
  - M: mRNA
  - B: β–galactosidase
  - A: allolactose
  - L: intracellular lactose
  - P: *lac* permease (transporter protein)

$$f_M = A$$
$$f_B = M$$
$$f_A = A \vee (L \wedge B)$$
$$f_L = P \vee (L \wedge \overline{B})$$
$$f_P = M$$

- Assumptions
  - Extracellular lactose is always available.
  - Extracellular glucose is always unavailable.
  - Translation and transcription require one unit of time.
  - Protein and mRNA degradation require one unit of time
  - Lactose metabolism require one unit of time

Using Macaulay2 to generate polynomials

- Let's see how to analyze this model two ways:
  - Using "Cyclone" in AlgoRun (polynomial algebra)
  - Using BoolNet in R
- For Cyclone, we need to convert these into polynomials.
- It will be convenient to do this using a software package.
- We'll use Macaulay2, an open-source computational algebra package that can be run online: http://www2.macaulay2.com/Macaulay2/
- Here's how to use Macaulay2 to turn these functions into polynomials:

$$f_M = A$$
$$f_B = M$$
$$f_A = A \vee (L \wedge B)$$
$$f_L = P \vee (L \wedge \overline{B})$$
$$f_P = M$$

```
Q = ZZ/2[M,B,A,L,P] / ideal(M^2-M,B^2-B,A^2-A,L^2-L,P^2-P)

RingElement | RingElement :=(x,y)->x+y+x*y;
RingElement & RingElement :=(x,y)->x*y;

fM = A;
fB = M;
fA = A | (L & B);
fL = P | (L & (1+B));
fP = M;

(fM, fB, fA, fL, fP)
```

Using Cyclone to compute the state space

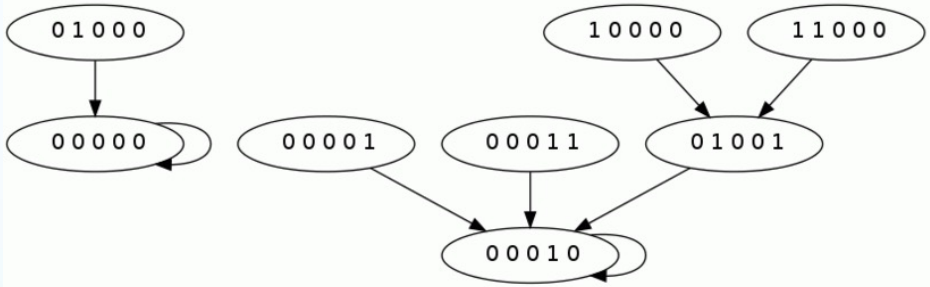
$$f_M = A$$
$$f_B = M$$
$$f_A = A \vee (L \wedge B)$$
$$f_L = P \vee (L \wedge \overline{B})$$
$$f_P = M$$

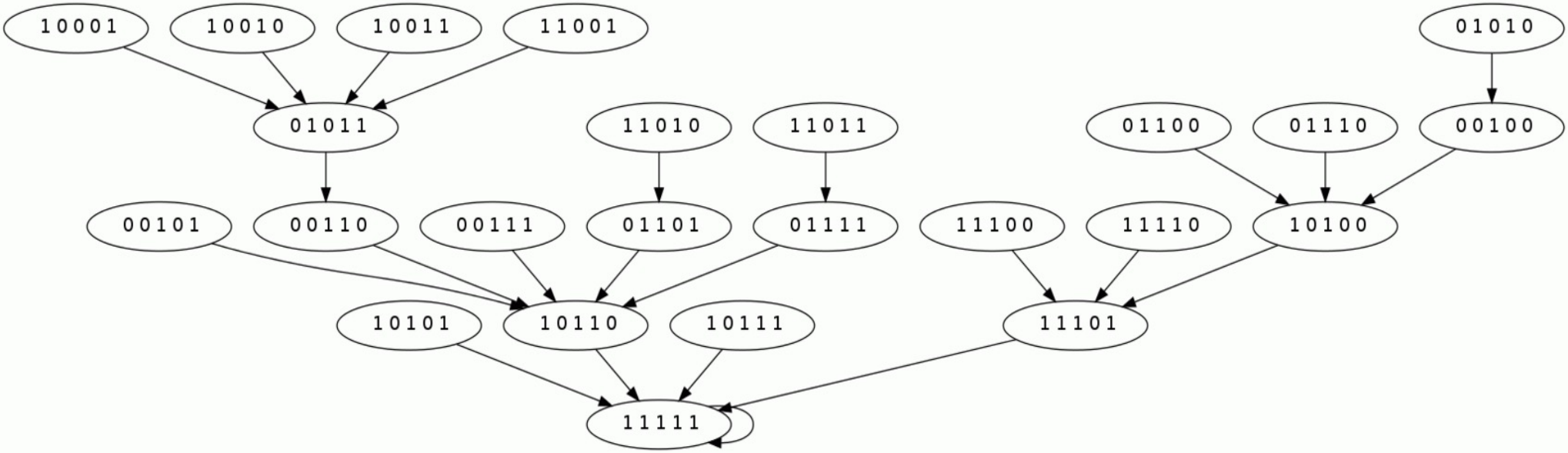
ALGORUN | Cyclone | Run | About

input

```
1  # Example Boolean Model
2  NUMBER OF VARIABLES: 5
3  NUMBER OF STATES :   2
4
5  M = A
6  B = M
7  A = B*A*L+B*L+A
8  L = B*L*P+B*L+L*P+L+P
9  P = M
```

http://cyclone.algorun.org/





Using BoolNet in R to compute the state space

lac-operon

```
1  targets, factors
2  M, A
3  B, M
4  A, A | (L & B)
5  L, P | (L & !B)
6  P, M
7
```

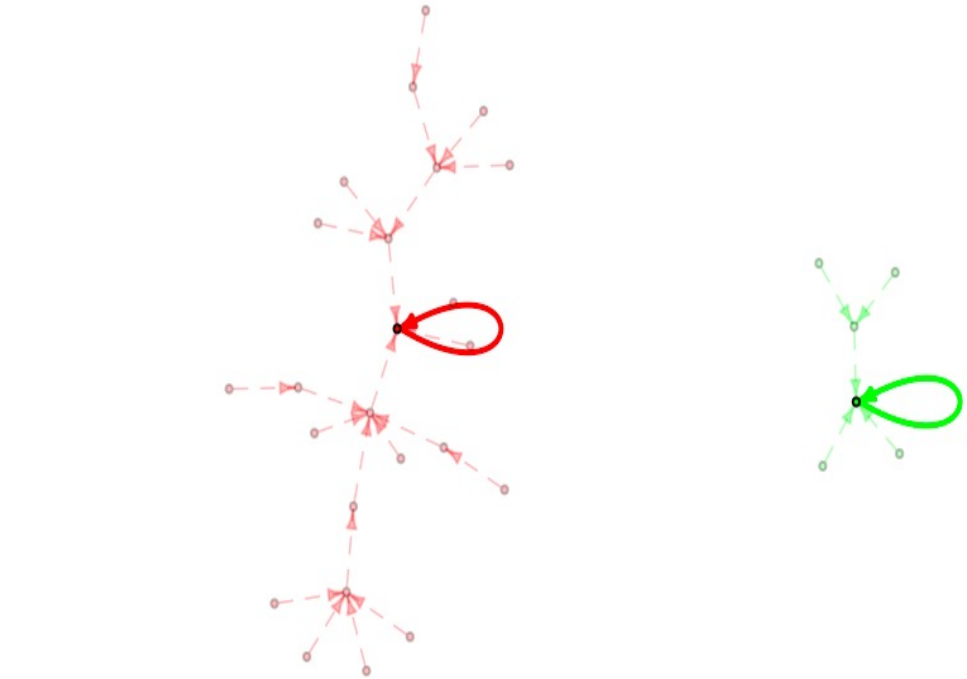
7:1 Text File

Console Terminal

~/code/BoolNet/

```
> lacModel <- loadNetwork("lac-operon")
> getAttractors(lacModel)
Attractor 1 is a simple attractor consisting of 1 state(s) and has a basin of 2 state(s):

 |--<----|
 V       |
 00000   |
 V       |
 |-->----|


Genes are encoded in the following order: M B A L P

Attractor 2 is a simple attractor consisting of 1 state(s) and has a basin of 6 state(s):

 |--<----|
 V       |
 00010   |
 V       |
 |-->----|


Genes are encoded in the following order: M B A L P

Attractor 3 is a simple attractor consisting of 1 state(s) and has a basin of 24 state(s):

 |--<----|
 V       |
 11111   |
 V       |
 |-->----|


Genes are encoded in the following order: M B A L P
```

```
> lacModel <- loadNetwork("lac-operon")
> lacAttractors <- getAttractors(lacModel)
> plotStateGraph(lacAttractors)
>
```



Problems with our refined model

- Model variables:
  - M: mRNA
  - B: β–galactosidase
  - A: allolactose
  - L: intracellular lactose
  - P: *lac* permease (transporter protein)

$$f_M = A$$
$$f_B = M$$
$$f_A = A \vee (L \wedge B)$$
$$f_L = P \vee (L \wedge \overline{B})$$
$$f_P = M$$

- Problems:
  - The fixed point (M,B,A,L,P) = (0,0,0,0,0) should not happen with lactose present but not glucose. [though let's try to justify this…]
  - The fixed point (M,B,A,L,P) = (0,0,0,1,0) is not biologically feasible: it would describe a scenario where the bacterium does not metabolize intracellular lactose.

- Conclusion: *The model fails the initial testing and validation, and is in need of modification*. (Homework!)