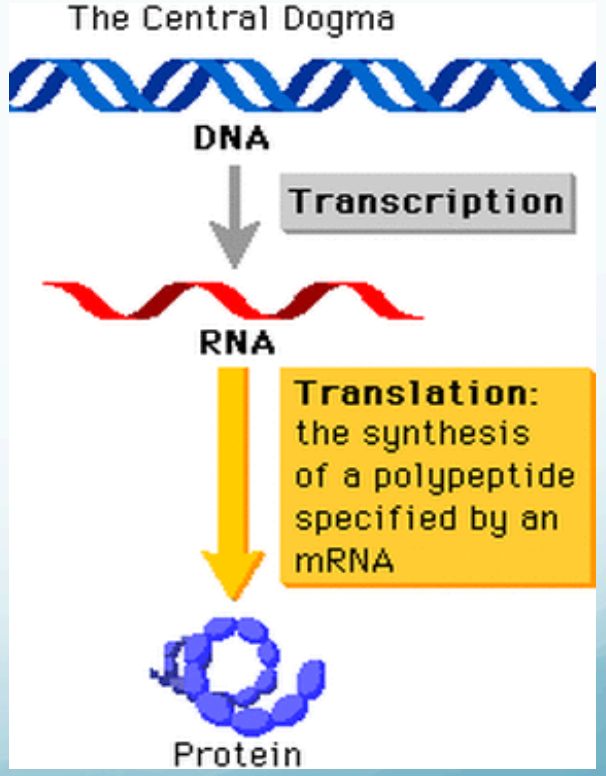
Boolean models of gene regulatory networks

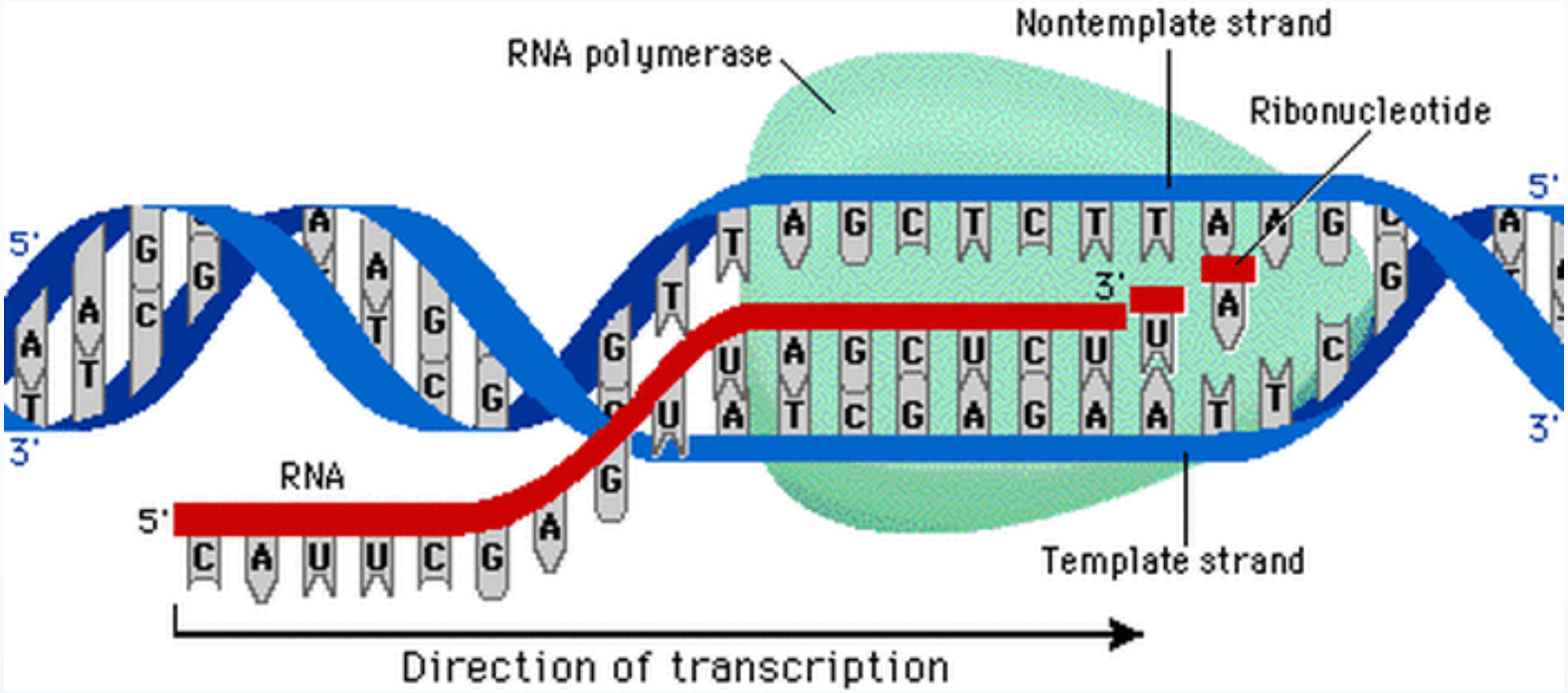
Matthew Macauley
Math 4500: Mathematical Modeling
Clemson University
Spring 2016

Gene expression

- Gene expression is a process that takes gene info and creates a functional gene product (e.g., a protein).
- Some genes code for proteins. Others (e.g., rRNA, tRNA) code for functional RNA.
- Gene Expression is a 2-step process:
  1) transcription of genes (messenger RNA synthesis)
  2) translation of genes (protein synthesis)
- DNA consists of bases A, C, G, T.
- RNA consists of bases A, C, G, U.
- Proteins are long chains of amino acids.
- Gene expression is used by all known life forms.

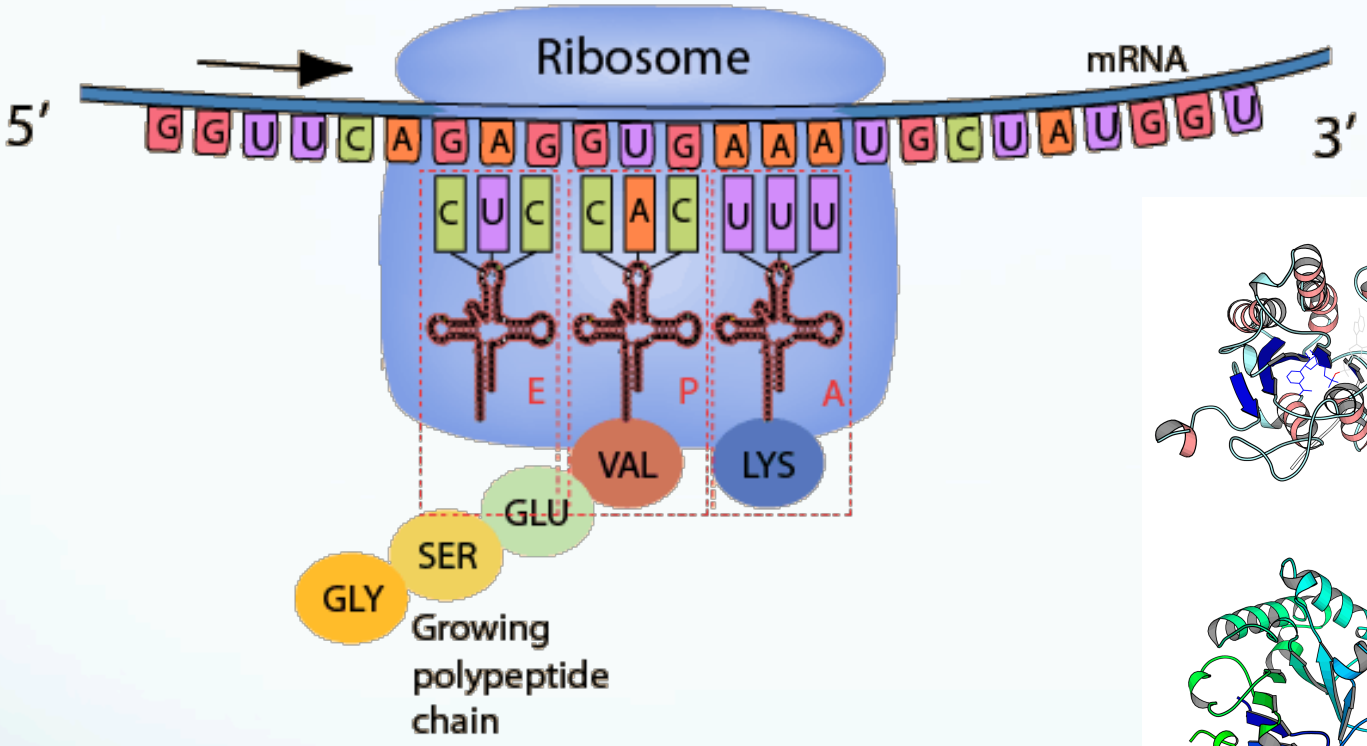


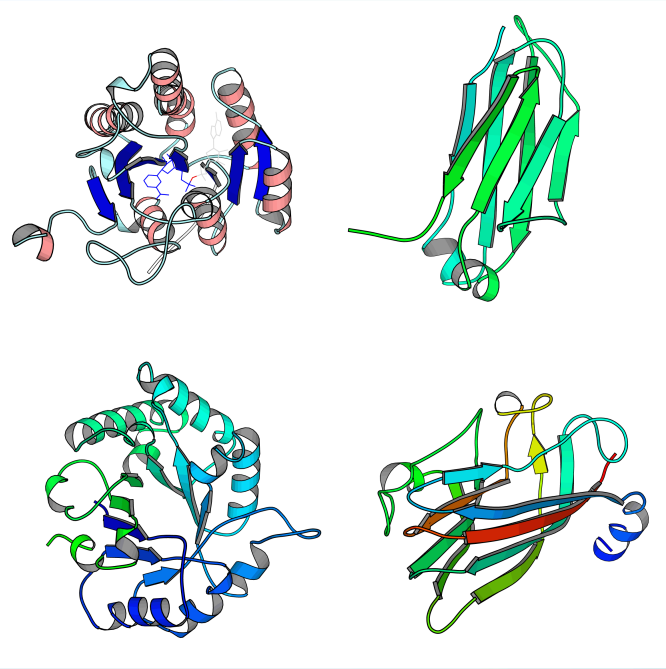
Transcription



- Transcription occurs inside the cell nucleus.
- A helicase enzyme binds to and “unzips” DNA to read it.
- DNA is copied into mRNA.
- Segments of RNA not needed for protein coding are removed.
- The RNA then leaves the cell nucleus.

Translation





- During translation, the mRNA is read by ribosomes.
- Each triple of RNA bases codes for an amino acid.
- The result is a protein: a long chain of amino acids.
- Proteins fold into a 3-D shape which determine their function

Gene expression

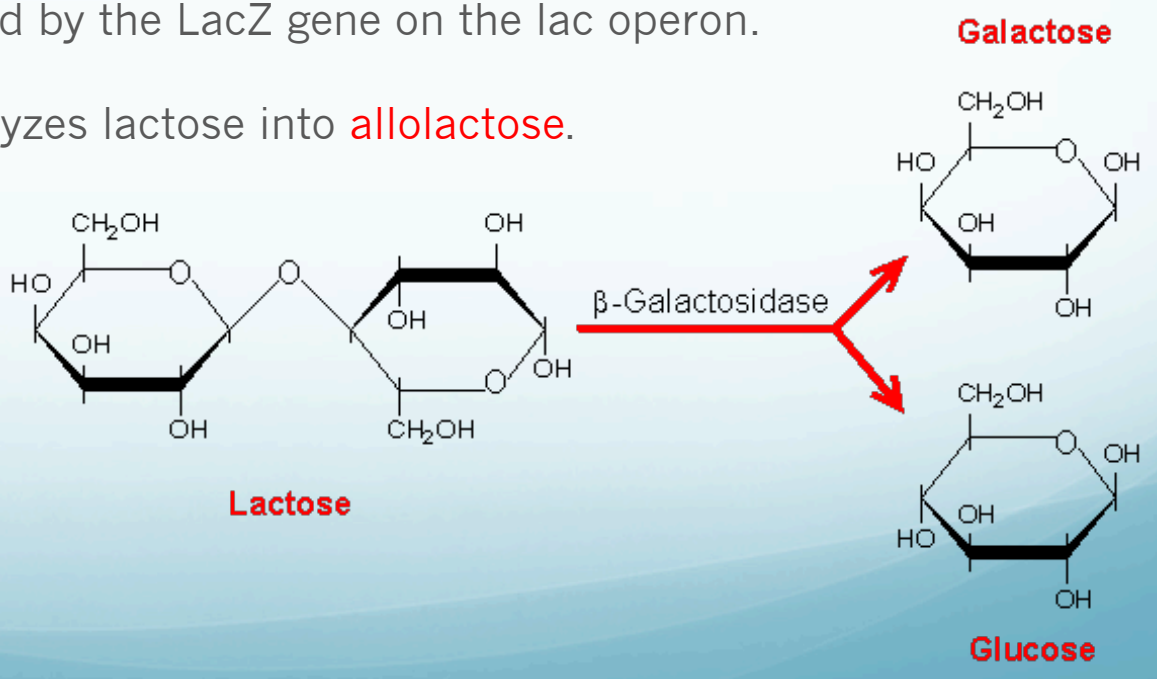
- The expression level is the rate at which a gene is being expressed.
- Housekeeping genes are continuously expressed, as they are essential for basic life processes.
- Regulated genes are expressed only under certain outside factors (environmental, physiological, etc.). Expression is controlled by the cell.
- It is easiest to control gene regulation by affecting transcription.
- Certain repressor proteins bind to sites on DNA or RNA.
- **Goal**: Understand the complex cell behaviors of gene regulation, which is the process of turning on/off certain genes depending on the requirements of the organism.

The *lac* operon in E. coli

- An operon is a region of DNA that contains a cluster of genes that are transcribed together.
- *E. coli* is a bacterium in the gut of mammals and birds. Its genome has been sequenced and its physiology is well-understood.
- The lactose (*lac*) operon controls the transport and metabolism of lactose in *Escherichia coli*.
- The *lac* operon was discovered by Francois Jacob and Jacques Monod in 1961, which earned them the Nobel Prize.
- The *lac* operon was the first operon discovered and is the most widely studied mechanism of gene regulation.
- The *lac* operon is used as a "test system" for models of gene regulation.
- DNA replication and gene expression were all studied in *E. coli* before they were studied in eukaryotic cells.

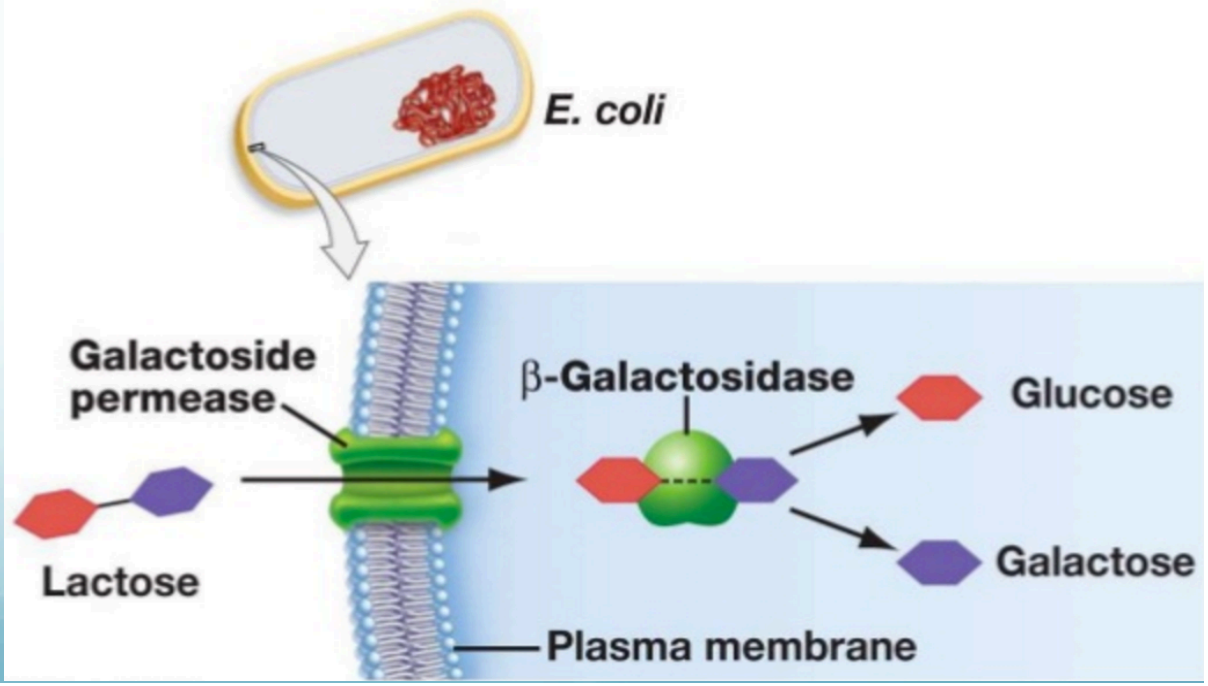
Lactose and β–galactosidase

- When a host consumes milk, *E. coli* is exposed to lactose (milk sugar).
- If both glucose and lactose are available, then glucose is the preferred energy source.
- Lactose consists of one glucose sugar linked to one galactose sugar.
- Before lactose can used as energy, the β–galactosidase enzyme is needed to break it down.
- β–galactosidase is encoded by the LacZ gene on the lac operon.
- β–galactosidase also catalyzes lactose into allolactose.

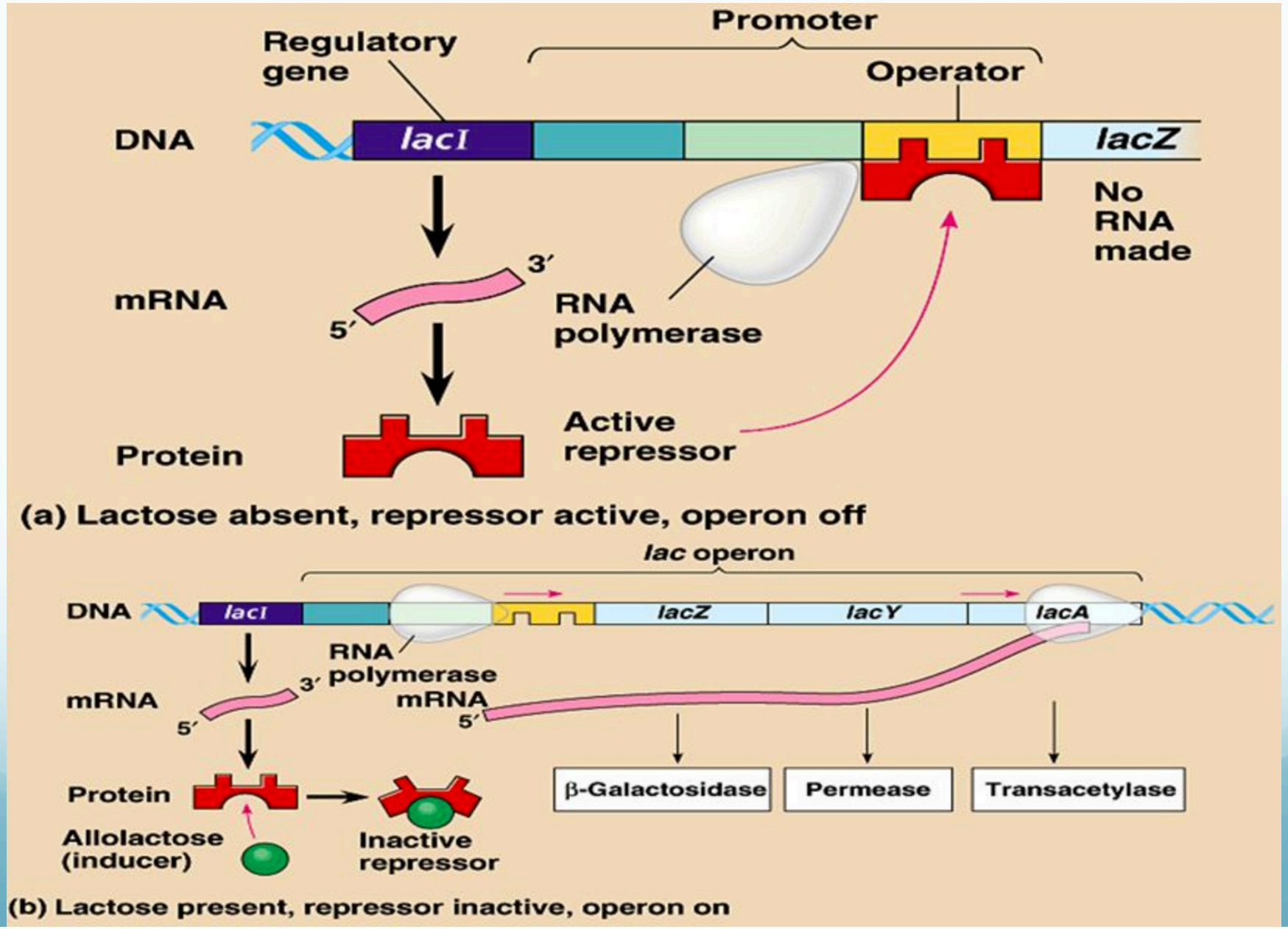


Transporter protein

- To bring lactose into the cell, a transport protein, called *lac* permease, is required.
- This protein is encoded by the LacY gene on the *lac* operon.
- If lactose is not present, then neither of the following are produced:
  1) *β* –galactosidase (LacZ gene)
  2) *lac* permease (LacY gene)
- In this case, the *lac* operon is OFF.



The *lac* operon

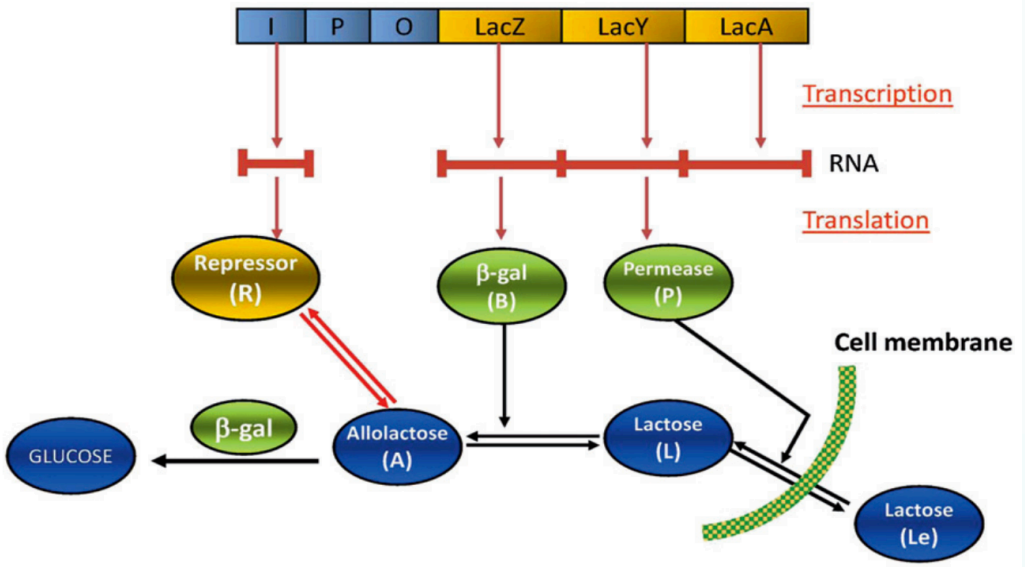


lac operon, with lactose present

- Lactose is brought into the cell by the *lac* permease transporter protein
- *β* –galactosidase breaks up lactose into glucose and galactose..
- *β* –galactosidase also converts lactose into allolactose.
- Allolactose binds to the *lac* repressor protein, preventing it from binding to the operator region of the genome.
- Transcription continues: mRNA encoding the *lac* genes is produced.
- Lac proteins are produced, and more lactose is brought into the cell. (The operon is ON.)
- Eventually, all lactose is used up, so there will be no more allolactose.
- The *lac* repressor can now bind to the operator, so mRNA transcription stops. (The operon has turned itself OFF.)

An ODE *lac* operon model

- M: mRNA
- B: β–galactosidase
- A: allolactose
- P: transporter protein
- L: lactose



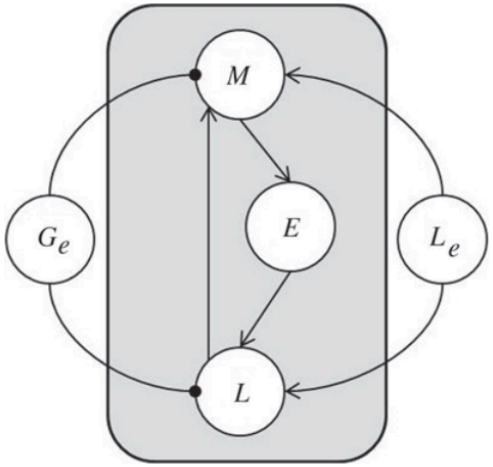
$$
\begin{aligned}
\frac{dM}{dt} &= \alpha_M \frac{1 + K_1 \left(e^{-\mu\tau_M} A_{\tau_M}\right)^n}{K + K_1 \left(e^{-\mu\tau_M} A_{\tau_M}\right)^n} + \Gamma_0 - \widetilde{\gamma}_M M \\
\frac{dB}{dt} &= \alpha_B e^{-\mu\tau_B} M_{\tau_B} - \widetilde{\gamma}_B B \\
\frac{dA}{dt} &= \alpha_A B \frac{L}{K_L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

Downsides of an ODE model

- Very mathematically advanced.
- Too hard to solve explicitly. Numerical methods are needed.
- MANY experimentally determined “rate constants” (I count 18...)
- 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



- What if we instead assumed 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).

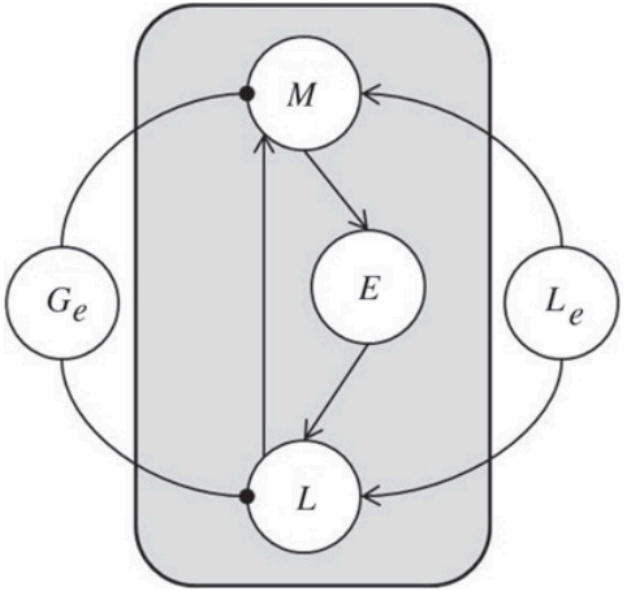
$$x_E(t+1) = f_E(t+1) = M(t)$$

- 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, ...



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

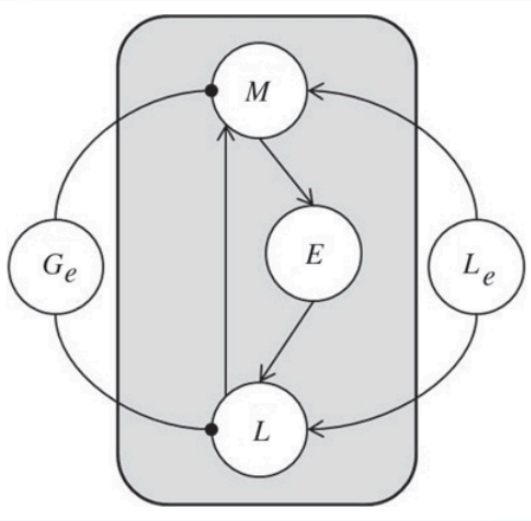
$$x_E(t+1) = f_E(t+1) = M(t)$$

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

- 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 the software: Analysis of Dynamical Algebraic Models (ADAM), at adam.plantsimlab.org.

- First, 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	$z = \overline{x}$	$z = 1 + x$

- Also, everything is done modulo 2, so 1+1=0, and x^2=x, and thus x(x+1)=0.

VIRGINIA BIOINFORMATICS INSTITUTE AT VIRGINIA TECH | VirginiaTech

Analysis of Dynamic Algebraic Models (ADAM) v1.1

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

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

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

Model Input:

Upload a file (.txt) for the polynomials corresponding to the system OR enter them directly into the text area:

SELECT FILES

Enter external parameters directly into the text area; one for each line:

```
f1 = (1+Ge)*(x3*Le+x3+Le)
f2 = x1
f3 = (1+Ge)*(x2*Le+x3*(1+x2))
```

```
Ge = 0
Le = 1
```

4)

What analysis would you like to run?

- (•) Simulation of all trajectories (< 20 nodes)

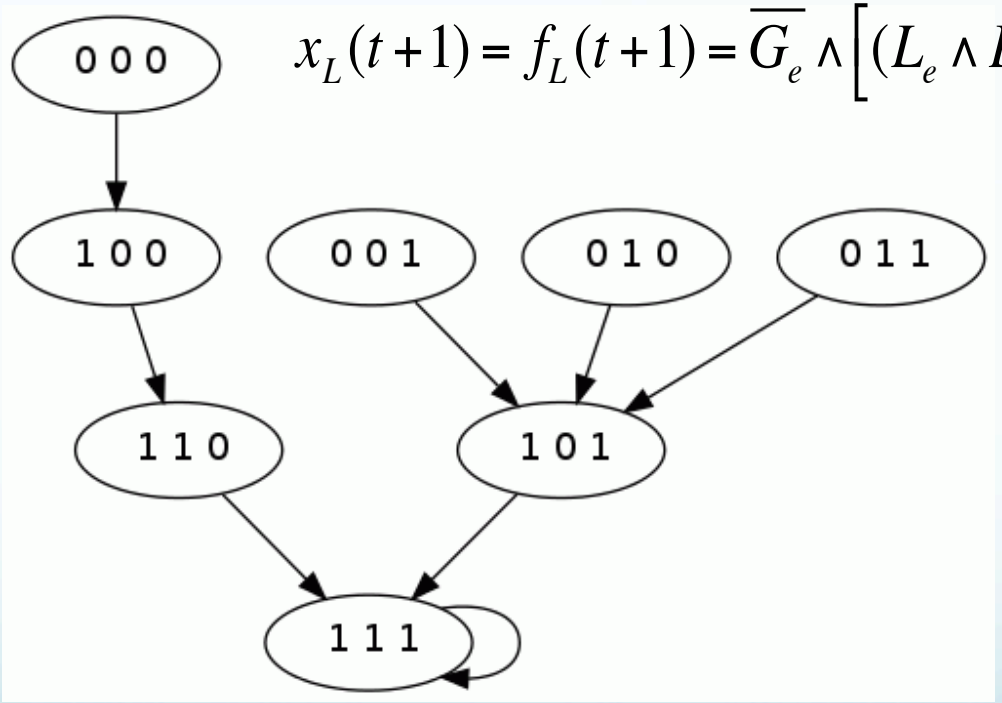
VIRGINIA BIOINFORMATICS INSTITUTE AT VIRGINIA TECH | VirginiaTech

Analysis of Dynamic Algebraic Models (ADAM) v1.1

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

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

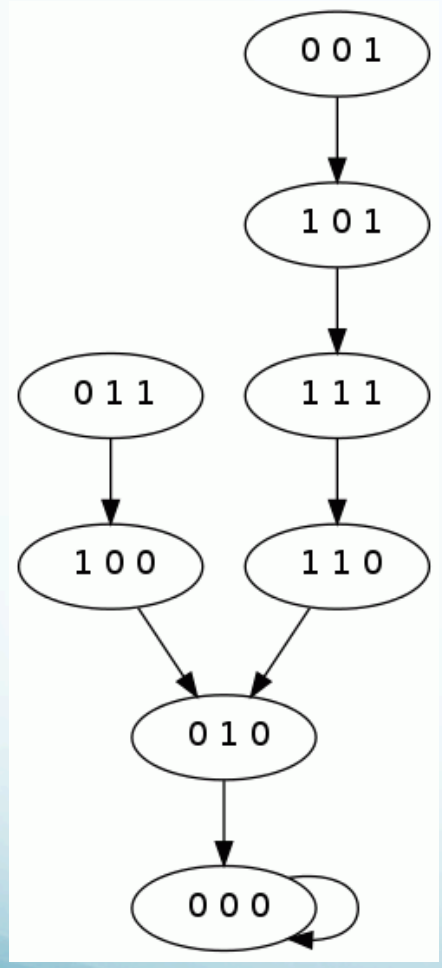
$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge \left[(L_e \wedge E(t)) \vee (L(t) \wedge \overline{E(t)}) \right]$$



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

VIRGINIA BIOINFORMATICS INSTITUTE AT VIRGINIA TECH | VirginiaTech

Analysis of Dynamic Algebraic Models (ADAM) v1.1



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

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

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

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

The operon is OFF.

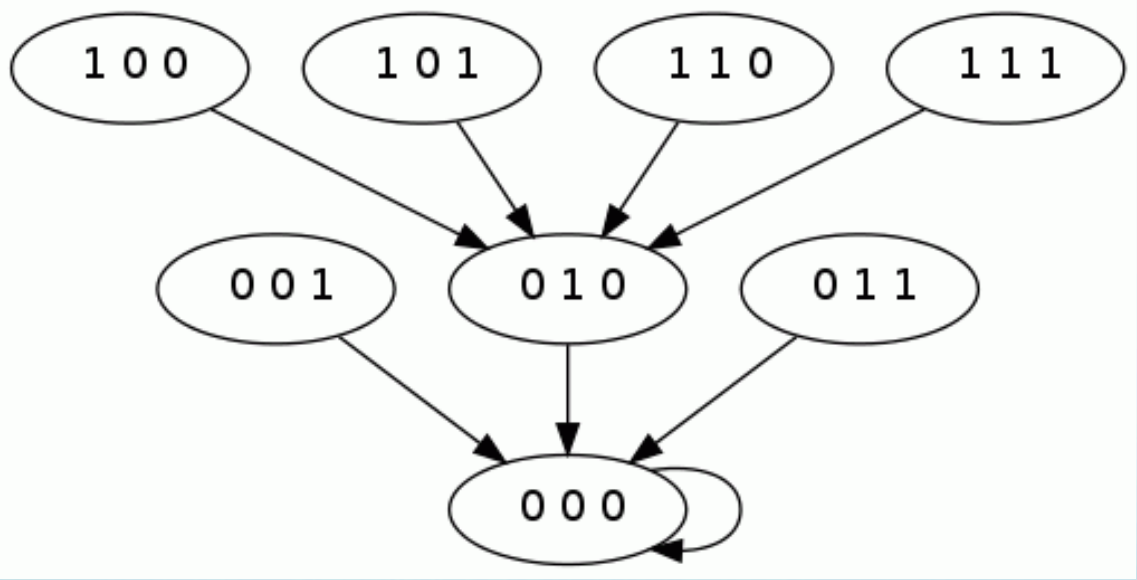
VIRGINIA BIOINFORMATICS INSTITUTE AT VIRGINIA TECH | VirginiaTech

Analysis of Dynamic Algebraic Models (ADAM) v1.1

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

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

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



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

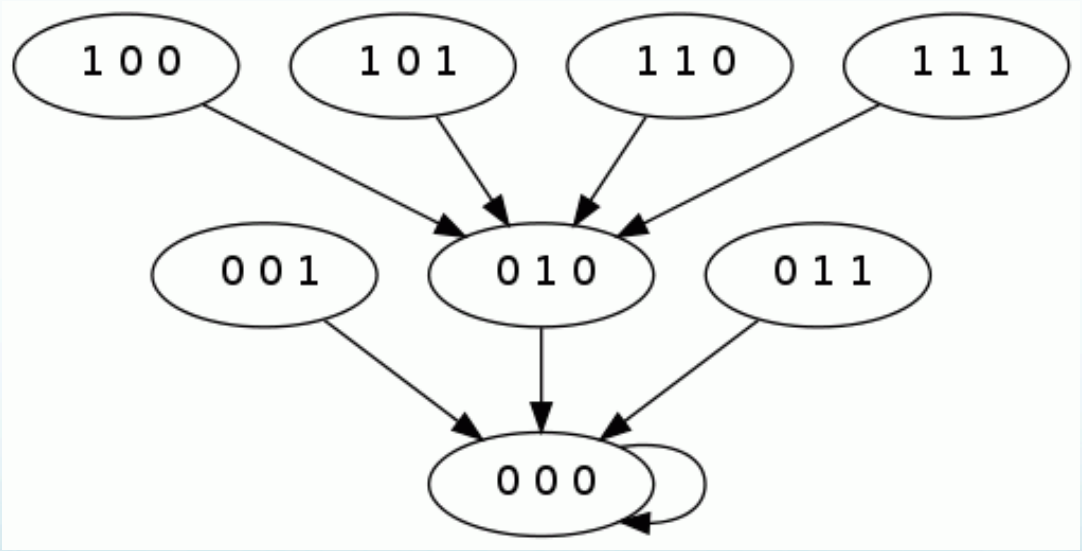
VIRGINIA BIOINFORMATICS INSTITUTE AT VIRGINIA TECH | VirginiaTech

Analysis of Dynamic Algebraic Models (ADAM) v1.1

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

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

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



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

Take-aways

- Gene regulatory networks consist of a collection of gene products that interact each other to control a specific cell function.
- Classically, these have been modeled quantitatively with differential equations (continuous models).
- Boolean networks take a different approach. They are discrete models that are inherently qualitative.
- The state space graph encodes all of 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 was a "toy model". We will study more complicated models of the *lac* operon shortly that captures more of the 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.