Module 4: Gene Regulation and Stability Using Boolean Logic

Platform: NetLogo

Following this lesson students should be able to:

1) Describe the components that make up a lac operon and how it functions
2) Apply Boolean logic to biological phenomena
3) Identify stability in a biological process and computational system
4) Evaluate model outputs to determine emergent phenomena

Purpose: This module will guide students through a tutorial to code a NetLogo model from scratch that uses Boolean Logic to create a simulation of the lac operon. Biologically, students will be exposed to a **common model of gene regulation**. Computationally, students will **define rules using Boolean logic** and **analyze model outputs to identify a system's emergent phenomena**.

Biological Terms:

1) Lac operon: the set of genes that regulate the creation of enzymes to break down lactose
2) Promotor: the binding site of RNA polymerase
3) Repressor: a protein that inhibits the transcription of the lac operon when lactose is not present
4) Operator: the binding site of the repressor protein
5) Lactose: the backup energy source
6) Glucose: the preferred energy source
7) Catabolite Activator Protein (CAP): a protein that becomes inactive with high levels of glucose
8) RNA polymerase: the protein that transcribes the lac operon
9) CAP Site: the binding site of CAP which speeds up the transcription of the lac operon when CAP is inactive
10) Transcription: the first step of expressing new proteins from genetic material

Computational Terms:

1) Boolean Logic: a problem which reports only a True or False value
2) If: used to carry out a function only when a Boolean problem is True
3) If Else: used to carry out two functions, one when a Boolean problem is True and one when it is False
4) Emergent phenomena: property that arises from the collective behavior of a dynamic system
5) Stability: the ability to converge to an equilibrium across a range of inputs or perturbations (unstable systems do not move toward an equilibrium)
6) Bistability: two states of equilibrium in a dynamic system

Time Estimation:

1) In-Class Activity: 30 minutes
2) Model Tutorial: 1 hour
3) Model Testing and Advancement: 2 hours 30 minutes

Total: 4 hours

Part One: In Class Activity

Materials: none

Rules:

1) Assign each student (or group of students) a role of the lac operon (promotor, repressor, lactose, glucose, CAP, RNA polymerase, enzymes)
2) Have students stand to represent an active state and sit to represent an inactive state
3) Have students decide what must happen for them to reach an active state (i.e. if they are the promotor, the repressor will need to be sitting) and what will need to happen to reach an inactive state (i.e. if they are lactose, enzymes will need to be standing)
4) Start with an inactive lac operon [Standing CAP and repressor] [Sitting: lactose, glucose, RNA polymerase, enzymes]
5) The lactose and glucose students are the inputs. Play a game of Simon Says If and If Else to determine their activation (i.e. If you are wearing a red shirt stand, if else remain seated)
6) After each If/If Else command allow the students to reach stability (i.e. no more changing of sitting vs. standing) before introducing another command
7) Keep introducing commands to the lactose and glucose inputs for multiple rounds. Make sure to give commands that reach all input possibilities at least once [(1) just glucose standing, (2) just lactose standing, and (3) both standing]

Suggested Discussion Questions:

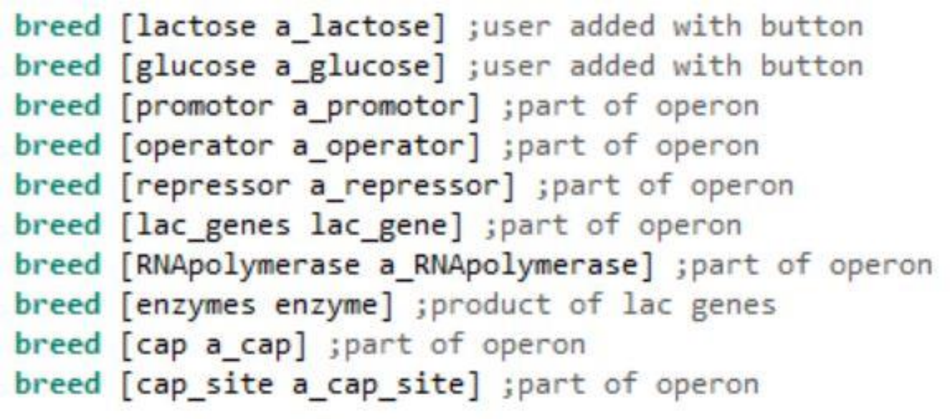
1) How is sitting or standing a good representation of Boolean logic?
2) What were the different points of stability reached throughout the activity?
3) Were there any components of the lac operon that Boolean logic does not fully cover?

Part Two: Model Tutorial

1) Open **NetLogo**

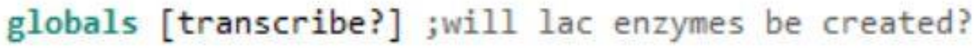
2) Instead of using one of the built-in codes, you will be making one from scratch today. Click the **Code** tab to get started

3) Define the breeds of variables that will be part of the model using **breed [plural_form singular_form]**. The components of this model should include lactose, glucose, promotor, operator, repressor, lac genes, RNA polymerase, enzymes, cap, and cap site



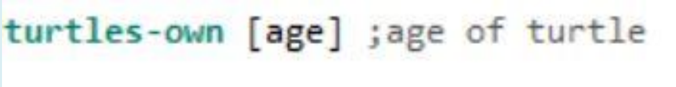
```
breed [lactose a_lactose] ;user added with button
breed [glucose a_glucose] ;user added with button
breed [promotor a_promotor] ;part of operon
breed [operator a_operator] ;part of operon
breed [repressor a_repressor] ;part of operon
breed [lac_genes lac_gene] ;part of operon
breed [RNApolymerase a_RNApolymerase] ;part of operon
breed [enzymes enzyme] ;product of lac genes
breed [cap a_cap] ;part of operon
breed [cap_site a_cap_site] ;part of operon
```

4) Define the global variable **transcribe?** with **globals [transcribe?]**. This variable will use Boolean logic to determine if the conditions are met to turn on the lac operon



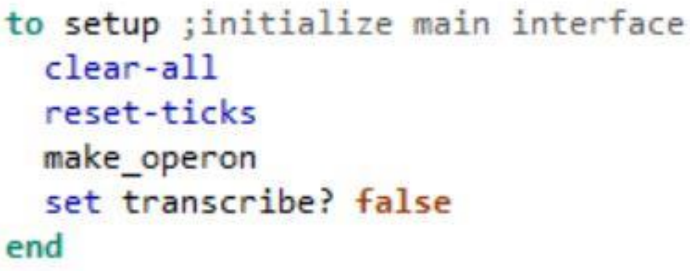
```
globals [transcribe?] ;will lac enzymes be created?
```

5) Define the turtle owned global variable **age** with **turtles-own [age]**. This variable will keep track of the age of the turtles



```
turtles-own [age] ;age of turtle
```

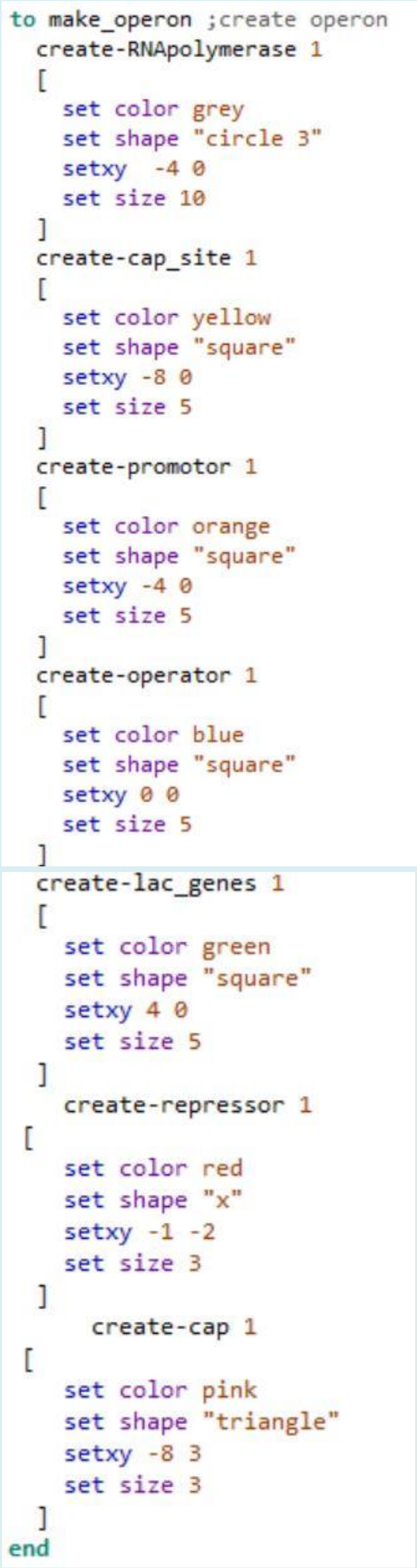
6) To set up the simulation, make a command called **to setup**. Under the command, type 4 actions (1) **clear-all**, which will clear all variables each time the code is run, (2) **reset-ticks**, which will initialize the time to be 0, (3) **make_operon**, a command which will create the lac operon, and (4) **set transcribe? false**, a command that will set the Boolean logic gate as false at beginning of the simulation. Finish the command by typing **end**



```
to setup ;initialize main interface
  clear-all
  reset-ticks
  make_operon
  set transcribe? false
end
```

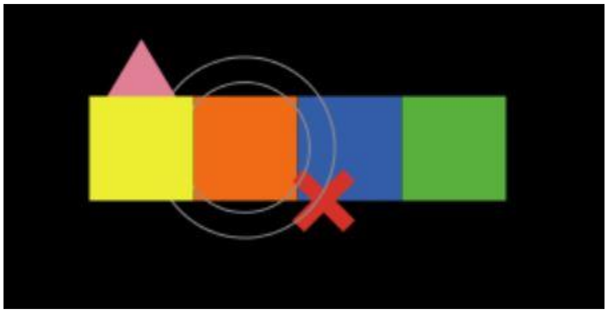
7) To create the lac operon, make a command called **to make_operon**. Under the command, create the 7 turtles that comprise the lac operon: operator, promotor, repressor, cap site, cap, lac genes, and RNA polymerase using the format **create-**

breed 1 [set color ____ set shape "____" setxy ____ ____ set size____]. You can customize with your own preferences in the ____s and finish the command with **end**

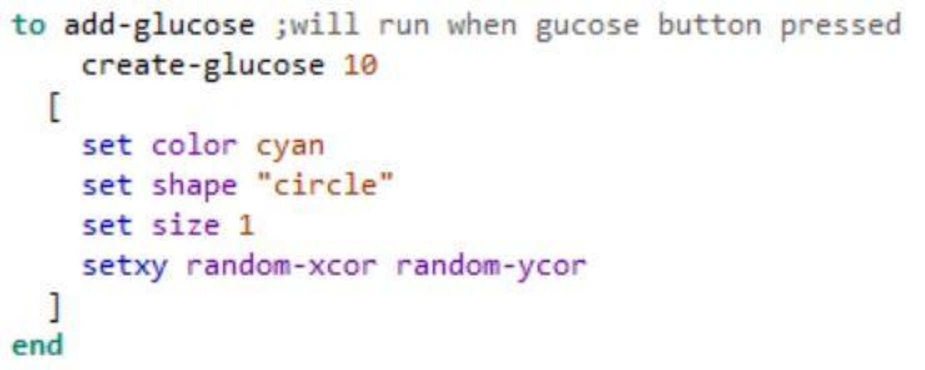


```
to make_operon ;create operon
  create-RNApolymerase 1
  [
    set color grey
    set shape "circle 3"
    setxy  -4 0
    set size 10
  ]
  create-cap_site 1
  [
    set color yellow
    set shape "square"
    setxy -8 0
    set size 5
  ]
  create-promotor 1
  [
    set color orange
    set shape "square"
    setxy -4 0
    set size 5
  ]
  create-operator 1
  [
    set color blue
    set shape "square"
    setxy 0 0
    set size 5
  ]
  create-lac_genes 1
  [
    set color green
    set shape "square"
    setxy 4 0
    set size 5
  ]
    create-repressor 1
 [
    set color red
    set shape "x"
    setxy -1 -2
    set size 3
  ]
      create-cap 1
 [
    set color pink
    set shape "triangle"
    setxy -8 3
    set size 3
  ]
end
```

8) Navigate to the main interface by clicking **Interface** and check to see if your lac operon was built correctly. Under **Button** click **Button**, place it near the top of the white part of the main interface, name it **setup**, and click **Ok**. Click the **setup** button and your lac operon should appear on the screen. If you used the above commands, it will give you the lac operon pictured bellow. [Hint: circle 3 for the RNA polymerase is not a default shape in the NetLogo Turtle Shapes Library and will need to be created]



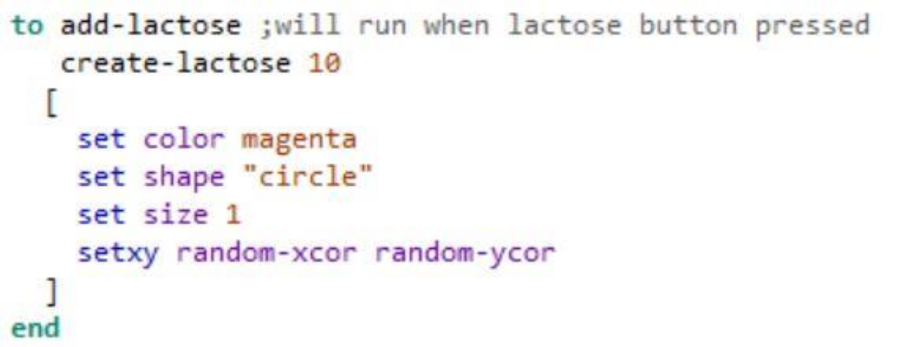
9) To add glucose to the model, make a command called **to add-glucose**. Under the command type **create-glucose 10 [set color cyan set shape "circle" set size 1 setxy random-xcor random y-cor] end**.



```
to add-glucose ;will run when gucose button pressed
    create-glucose 10
  [
    set color cyan
    set shape "circle"
    set size 1
    setxy random-xcor random-ycor
  ]
end
```

10) Navigate to the main interface by clicking **Interface.** Under **Button** click **Button**, place it near the top of the white part of the main interface, name it **add-glucose**, and click **Ok**. When this button is clicked it will add 10 glucose molecules to the simulation

11) Repeat steps 9 & 10 to add lactose to the model in the code and with the button. Instead of the color **cyan**, use **magenta**



```
to add-lactose ;will run when lactose button pressed
   create-lactose 10
  [
    set color magenta
    set shape "circle"
    set size 1
    setxy random-xcor random-ycor
  ]
end
```

12) While on the main interface, also add a plot to the screen. Under **Button** click **Plot** and add it to the bottom of the white part of the main screen. Right click on the plot

and choose **Edit**. Name the plot **Gene Regulation**, the x axis label **Time**, and the y axis label **Amount**

13) Under **pen update commands** change **plot count turtles** to **plot count glucose** and change to default color from black to cyan by clicking on the **black box**, choosing **cyan**, and clicking **Ok**.

14) Click **add pen** and repeat step 13 by naming the pen command, **plot count lactose** and making the color **magenta**

15) Click **add pen** and repeat step 13 by naming the pen command, **plot count enzymes** and making the color **green**. Click **Ok** to exit the plot

16) Navigate back to **Code**. Make a command called **to go** under the command type **tick** and **end**

17) Navigate to the main interface by clicking **Interface.** Under **Button** click **Button**, place it near the top of the white part of the main interface, name it **go**, and click **Ok**. When this button is clicked it will start the simulation. More commands will need to be added under **to go** and the model to simulate the lac operon which you will do in part three

Module 4: Gene Regulation and Stability Using Boolean Logic

1) To define rules for the agents in the simulation you will need to determine what you want the rules to be. Consider how the lac operon works (https://www.khanacademy.org/science/ap-biology/gene-expression-and-regulation/regulation-of-gene-expression-and-cell-specialization/a/the-lac-operon). Specifically think about:
   a. When lactose is present what happens to the repressor?

   b. How does the removal of the repressor effect transcription?

   c. When glucose is present what happens to CAP?

   d. How does the removal of CAP affect transcription?

   e. What is produced as a result of transcription?

   f. What parts of the lac operon including its inputs and products move?

2) Using your considerations from above define the outcomes (rules) for the Boolean Logic problems below. The first two have been completed as examples
   a. If lactose > 0
      True: move repressor and start transcription
      False: nothing

   b. If else lactose >= (greater or equal to) glucose
      True: fast transcription
      False: slow transcription

c. If glucose > 0 [Hint: what happens to CAP?]

True:

False

d. If enzyme > lactose

True:

False:

e. If enzyme on lactose [Hint: what is the purpose of the enzymes produced by the lac operon?]

True:

False:

3) Identify any assumptions you may want to make in order to decrease the complexity of your model.

a.

4) Using these rules and assumptions, determine the code that will need to be added to the model to carry out the rules. Use NetLogo Dictionary (**http://ccl.northwestern.edu/netlogo/docs/index2.html**) for help on syntax and codes that are available. [Hint: the codes **if** and **ifelse** should be very helpful and comprise most of the added components. Also **count**, **ask**, **die**, **any?** and **move** should be helpful]

5) Once you have a working code, it is important to test that it works. Come up with at least 5 tests (one per rule) to determine if your simulation is working properly.

   a.

   b.

   c.

   d.

   e.

6) Perform your tests. If your code is not performing how you anticipated try troubleshooting or writing a new code and retry your test.

   a. Describe the outcome of your tests and the troubleshooting process.

7) Analyzing how a simulation performs when varying input parameters (i.e. changing the amount of lactose or glucose added to the simulation) can uncover emergent phenomena of the model. An emergence of the lac operon biologically and in this simulation is it bistability. Using the Gene Regulation plot from the simulation explain how we can make this conclusion.

   a.

8) This model is extremely basic and has many limitations which do not capture the full workings of a lac operon.

   a. In your opinion, what are the model's three biggest limitations? Especially consider any assumptions that were made when answering this question.

   b. Choose one of these limitations and describe what you think should be added to the model to address this limitation.