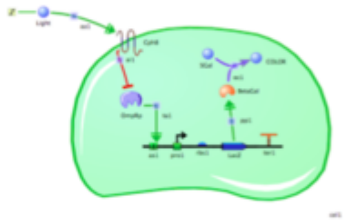
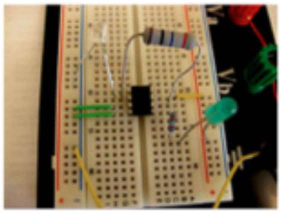
Lab 3: Picture this






TinkerCell Activity

The ultimate goal for this activity is to simulate the operation of the bacterial photography system using the CAD tool. The tool we've chosen (Tinkercell) is one of many, but it is useful in this teaching context because it has an easy-to-use graphical interface, can be an informative "drawing" tool in other contexts when you'll be teaching gene expression, and does not require computer programming skills. During this activity, the students will create a CAD model of the bacterial photography system and then manipulate various parameters, such as promoter strength, concentration of substrate and enzyme activity to examine their effects on the system. In doing this, the students will learn how the parts of a 2 component sensing system (and other genetic and enzyme systems) interrelate. We recommend that the students spend one class period conducting the TinkerCell activity. A second class period can be devoted to altering parameters and examining effects. If students have access to the internet at home or in their dorm, they can download TinkerCell and conduct further studies outside of class.

A correctly built TinkerCell model file can be downloaded here.

Needed Materials

Teacher Provides

Computers for the TinkerCell activity

Kit Provides

USB key with program for loading on student laptops. Alternatively, the program can be downloaded from the Tinkercell homepage, though a slow internet connection will make this a slow download.

Annotated Procedure

Introduction

If you have not had a lesson on how the bacterial photography system works, go read the first half of the design assignment page.

TEACHERS: We recommend you give a lecture describing two-component signaling and the bacterial photography system in more detail.

Part I: TinkerCell

Computer-aided design (CAD) is a hallmark of several mature engineering disciplines, like mechanical engineering or civil engineering. These engineers can rely on computer simulations to reliably predict the behavior of a car or a bridge, rather than run a hundred cars into walls to see how they perform. Biological engineers have fewer good CAD tools at their disposal. More often, they must run laboratory experiments to test a system. But wouldn't it be nice (and quicker and less expensive too!) to try a few things on a computer first? And then, with some good candidate designs in hand, we could turn to the bench with more confidence, having eliminated the clear failures.

TinkerCell

One early effort at a CAD tool for synthetic biology is TinkerCell, developed by engineers at the University of Washington. TinkerCell allows you to visually construct and then simulate/analyze a biological network. Using the following instructions, you can use TinkerCell to build the bacterial photography system (or at least a simplified model of it). For those who would like to read more about the TinkerCell CAD tool, you can find the details in this article from the Journal of Biological Engineering.

Getting Started with TinkerCell

TinkerCell can be downloaded for free from this page. Make sure you download the "current" version, not the "stable" version. The instructions in this tutorial were written for the Mac-based version of the program (1.2.472). If you are running a different version of TinkerCell or if you are running Tinkercell on Windows or Linux, you may see some subtle differences.

After you open the TinkerCell application, begin familiarizing yourself with the basic operation of the program. In particular, try to use

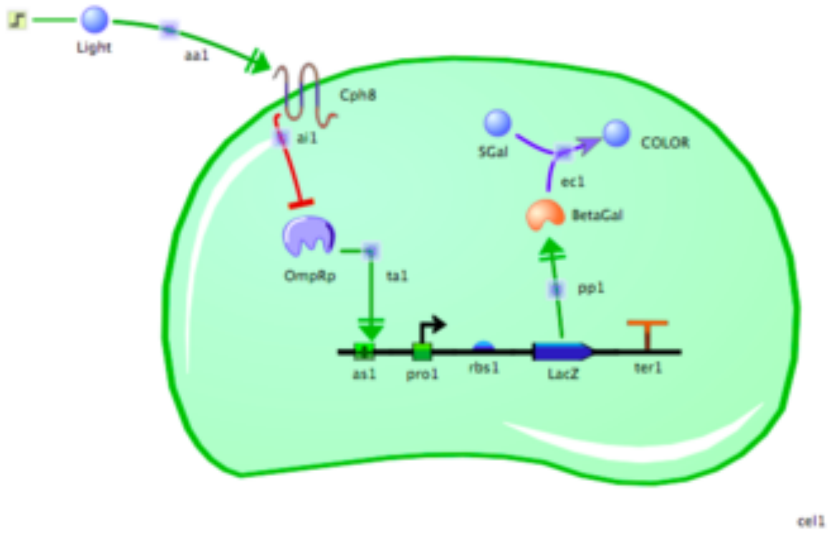
- the **Molecules** and **Reaction** tabs: try to select 2 molecules from the molecules that are available. For example, click the "Enzyme" on the icon strip and then click the network canvas to place an enzyme. Repeat with a second molecule, selecting "Transcription Factor" from the icon strip and placing it on the network canvas. Next, choose the Reaction tab and select either activation or repression. Click on the enzyme first, then the transcription factor. If you chose activation, you'll be asked to choose between two mechanisms. A reaction arrow should appear. Next, if you like, try stamping out two receptor molecules and connect them with a different kind of regulation, or try making an enzyme catalyze a reaction with one or more small molecules.

- the **Parts** and **Regulation** tabs: Choose the Parts tab and try stamping out a gene expression cassette, i.e. an operator (activator or repressor binding site), a promoter, an RBS, a protein coding sequence, and a terminator (optional). The parts do not need to be aligned when you place them on the network canvas, but if you drag them next to one another they should connect. You can then choose "Transcription Regulation" from the Regulation tab, and link the transcription factor you placed earlier to the gene's operator. A reaction arrow should appear. You can move the icons on the canvas, reshape the reaction arrows, and relabel the parts to your liking. Try using "Protein Production" from the Reaction tab to make the coding region of the gene produce a protein.

TEACHERS: At this stage, make sure the parts placed on the canvas have proper rate equations associated with them. TinkerCell automatically writes an appropriate rate equation for each part, but occasionally this fails due to a bug. You can see the rate equations for each part in the summary menu on the right side of the screen, or by hovering your mouse over a component. For example, if you have Protein A regulating Protein B, then if you hover over Protein B, a little graph should pop up showing the relation between Protein B activity and Protein A activity. If these functions sometimes appear and sometimes don't, try deleting and replacing the components without functions. If no functions ever appear for any component, make sure you downloaded the *current* version from the TinkerCell website, not the *stable* version.

TEACHERS: If you can't find the right part or reaction at any step, look among the tabs and choose something that sounds reasonable. One of the main differences between versions of TinkerCell is that reactions, regulatory interactions, etc. get shuffled around between the tabs and renamed.

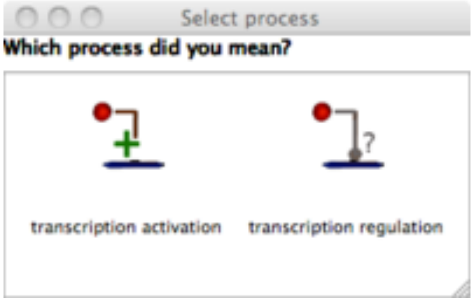
Building the bacterial photography system



Fully built bacterial photography system

Now that you have the basic mechanics in hand, you can build the bacterial photography system. Follow the steps below.

- **Start this project** on a new canvas. Select "New Canvas" from the File Menu, or click the new page icon on the top toolbar.
- **Assemble the reporter gene**: From the "Parts" tab, place an "Activator Binding Site", "Promoter," "RBS," and "Coding" icon on your canvas. Drag the parts next to each other, in that order, so they snap together.
- **Name the reporter gene elements**: Click on the name below each part to rename it. The promoter should be named "PompC." The RBS can be left as is. The coding sequence can be renamed "LacZ".
- **Add the transcription factor**: From the "Molecules" tab, select "Transcription Factor" and place one on the canvas. It will represent the phosphorylated form of OmpR, so rename it "OmpRp".
- **Visual appeal**: Select the OmpRp protein, and then from the "Edit" menu choose "Add decorator." A dialog box will display a choice of icons. From the "Decorators" tab, select "phosphorylation".
- **Activate Transcription of PompC with OmpRp**: From the "Regulation" tab, choose "Transcriptional Activation" and then click on OmpRp and the activator binding site just before PompC. Choose “Transcription Activation” from the pop-up menu.


Select process

Which process did you mean?

Regulating transcription

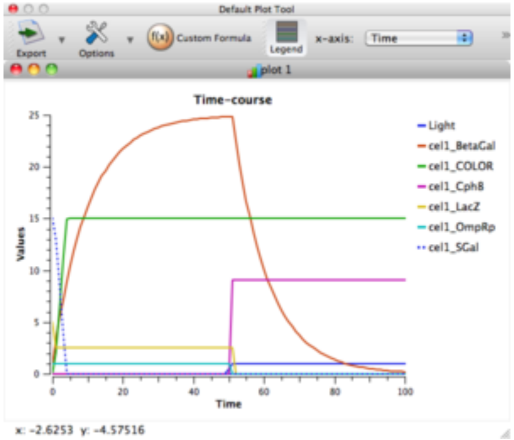
- **Add the Cph8 light receptor**: From the "Molecules" tab, choose "Receptor" and place one on the canvas. Rename it "Cph8."
- **Regulate OmpRp with Cph8**: From the "Regulation" tab, choose "Allosteric Inhibition" and then click on Cph8 and OmpRp. You can reshape the regulatory arrows and move the elements around the canvas as needed for clarity.

TEACHERS: In reality, in its nonphosphorylated form, Cph8 inhibits the activity of OmpRp. Thus, "phosphorylation-dephosphorylation cycle" is a better representation of reality, but "allosteric inhibition" works for our purposes and is simpler.

- **Add the Beta-Gal protein**: From the "Molecules" tab, place an enzyme on the canvas and call it "BetaGal". Link it to the LacZ coding region using the "Protein Production" reaction.
- **Add the colorful small molecule**: Place two more small molecules on the canvas. Connect them to your BetaGal protein using "Enzyme Catalysis" from the "Regulation" tab. Name the input molecule "SGal", and the output molecule "COLOR." Next, double-click on SGal to open a dialog box. Select the "Initial Conditions" tab and increase the concentration to 15.
- **Add a Chassis**: From the "Compartments" tab, choose “Cell” and place one on the canvas. Move and resize the cell so it encloses the transcription factor and the reporter gene. Leave the Cph8 receptor in the cell membrane.
- **Add light**: From the "Molecules" tab, choose "Small Molecule" and print one on the canvas. Rename it "Light" and connect it to Cph8 with an activation arrow from the "Reaction" tab.
- **Turn the light on**: From the "Inputs" tab, choose "Step input" and click on the Light molecule. Next, find the cell compartment in the "Model Summary" list of components on the right. Click the arrow to expand its subcomponents, find the Light molecule, and expand it. Change the value of "step_time" to 50 (this will turn on the light halfway through the simulation, rather than right at the beginning).

Whew! Now we'll go on to simulate the photography system in action...

Simulating the bacterial photography system



Behind the shiny-looking front end of TinkerCell is some serious mathematical capability. We'll use the "Deterministic" simulator (i.e. we will be ignoring the random fluctuations that would occur in a real cell). Click the big green arrow in the top toolbar to run the system.

TEACHERS: The opposite of "deterministic" is "stochastic" -- that is, containing noise and random fluctuations. The workings of a real cell are always affected by the inherent randomness of molecules bouncing around and colliding with each other. Not all cells behave the same way! Not all cells in the dark will produce the exact same level of colored pigment, and cells in the light will produce some small level of pigment despite being nominally "turned off". This random noise is analogous to static in a radio or television. You may want to emphasize this point when you discuss the drawbacks of TinkerCell modeling.

You should see a graph window and a second window with sliders. To make the output a little easier to read, click on "Legend" in the graph window, and uncheck everything except Time, Light, Cph8, OmpRp, LacZ, BetaGal, and COLOR. Take a few minutes to familiarize yourself with the output graph.

You may want to take time now to answer **Questions 1-3** found in the **"Putting it all together"** section (below), instead of waiting until the end of the exercise.

TEACHERS: Note the following --

- All the lines the students unchecked are either constant (e.g. terminator activity) or internal TinkerCell bookkeeping variables (e.g. concentration of inactive Cph8 receptor).
- Although TinkerCell includes nominal units for many of the variables, they are largely arbitrary.
- Most components do not degrade unless you explicitly add a degradation reaction. However, the "Protein Production" module automatically includes degradation. Thus, BetaGal protein degrades but COLOR molecule doesn't -- this may be a point of confusion for the students. In particular, degradation is why the concentration of BetaGal protein increases and decreases logarithmically, rather than going straight to steady state as most of the other lines do.
- There will be a small amount of strange-looking transient behavior at the very beginning of the graph (in the first 5 time units). This can be safely

ignored. You may be able to make it go away by setting the initial concentrations of Cph8, OmpRp, and/or LacZ to zero, but this may also mess up your graphs. (To change the initial concentration/activity of something, double-click on it, select the "Initial Conditions" tab, and type the new value into the box.)

Tuning the system

Turn to the slider window and start changing numbers, making notes of what effects you see. Remember that if the system's behavior gets way out of whack, you can always close the graph/slider windows and re-run the simulation with default values. Many of the variables in the slider window have straightforward names, like "light_step_height", but some are more opaque. In general, things named "Kd" are constants governing the strength of a regulatory interaction. "Kcat" is the catalytic efficiency of the b-gal enzyme.

Suggestions for tweaking the system

You may be asked to run these simulations, to take notes on your findings, or to skip ahead to the next section, called "Let the games begin."

- Change the amount of SGal in the bacteria's environment (the "cel1_SGal" slider). What happens to the COLOR output?
- Change the efficiency of the BetaGal enzyme (the "cel1_ec1_Kcat" slider). What happens to the output?
- Decrease "Light_step_steepness", so that the light turns on gradually rather than all at once. Does this tell you more about how exactly the system responds when its input changes?
- Try changing the light's step input to a sine wave input, and change the frequency. What happens to the output when the frequency is high? When the frequency is low?

Let the games begin

Next, use this model to realize a design specification you have in mind. First, you must decide on some specific changes you would like to make to the system. Would you like the bacterial photographs to develop faster/slower/darker/lighter...? Next, think about how that change would affect the simulation and the graph. Finally, see if you can find the right combination of sliders to make those changes happen. Here are some ideas:

- the bacterial photographs require at least 24 hours to develop in the lab, but they are stable once formed. What elements in the system could be changed to make the COLOR output accumulate more slowly? More quickly?
- the bacterial photographs can sometimes look less intensely colored than the original mask that's used to generate the image. What elements in the system could be varied to change the total amount of color produced?

Putting it all together

Once you've worked through some simulations (or while you work through them), you may be asked by your teacher to answer some of the following questions.

1. What are some approximations/simplifications that you made as you built the model? For example, if you followed these directions then there is no phosphorylated form of Cph8. Do the approximations you made matter?
2. What are the axes of the graph? What are the units of each quantity (concentration, rate, etc.)? (Check the listed units for each quantity in the summary menu on the right.)
3. Consider each line on the graph. Does it go up, down, or stay static in the first half of the simulation (light off)? Does it change at the time the light turns on? Does it go up, down, or stay static in the second half of the simulation (light on)?
4. Choose at least two adjustments that you made to the sliders and describe
    - what change you made
    - how that change affected the shape of the lines on the graph, and
    - why each adjustment made the difference that it did.
5. Describe how you might carry out some of the simulated adjustments in the lab or in a real cell.
6. Why is it valuable for scientists to be able to model their experiments? What about the value of models and simulations for engineers? Finally, did you personally find it valuable (why/why not)?
7. What are some of the drawbacks to this modeling approach? Where does the analogy between a TinkerCell canvas and a biological system break down, so that the model no longer reflects reality?
8. Did you encounter any bugs in TinkerCell? Are there features you would like to see? Describe each bug or feature precisely. Collect all the bug reports / feature requests as a class, and submit them to the TinkerCell development team.
9. Finally, did this exercise suggest any modifications you would like to make to the system, or experiments you would like to try?

Survey

To help us improve the labs, you can

1. send the students here, where they can upload their data.
2. "join a discussion" from the BioBuilder homepage
3. email us: "info AT biobuilder DOT org"

Thanks!

Feedback

We're always looking to hear back from you if you've thought about this unit, tried it, or stumbled across it and want to know more. Please email us through BioBuilder, info AT biobuilder DOT org.