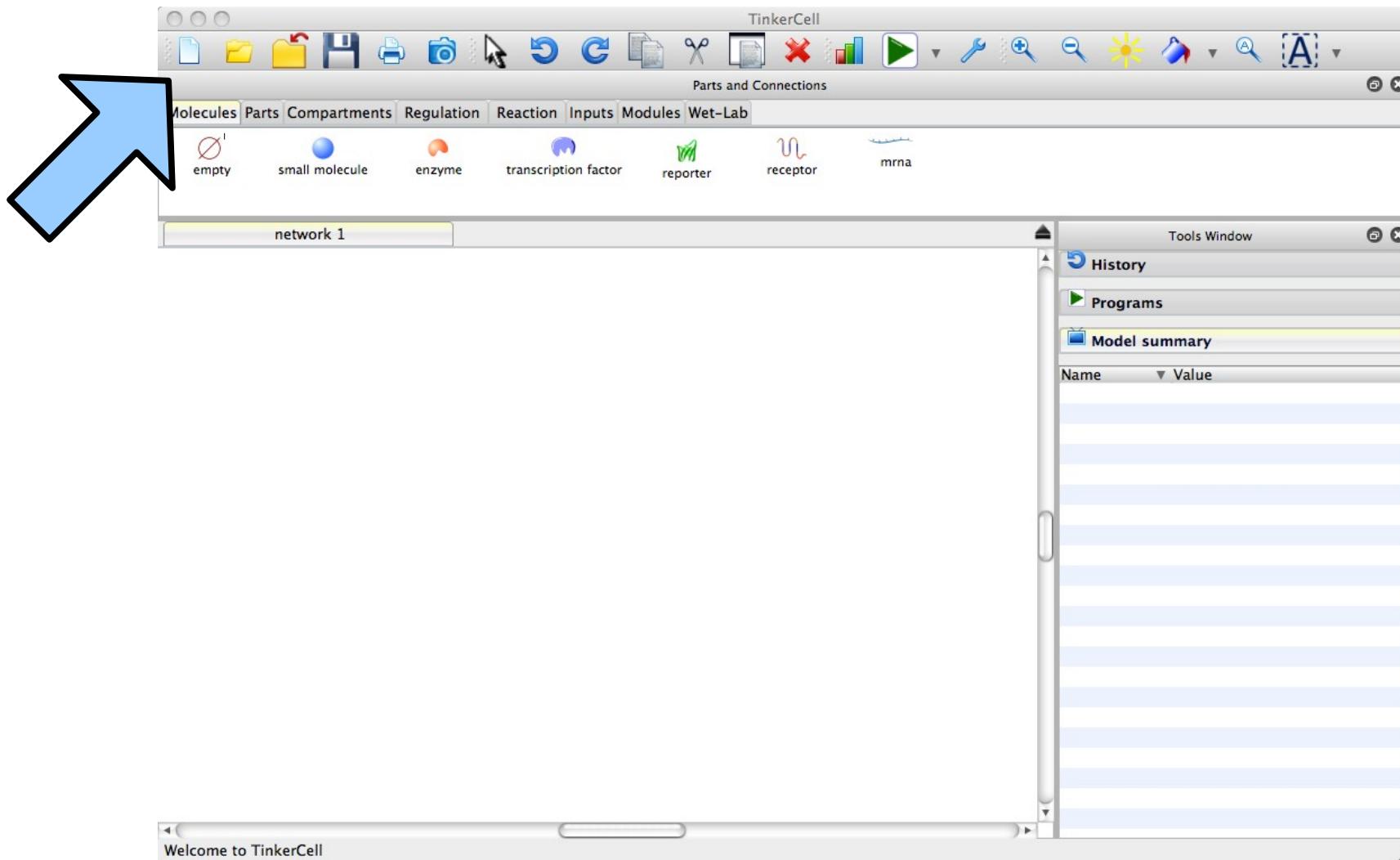


TinkerCell Tutorial: The Bacterial Photography System

Thanks and appreciation to Deepak Chandra from U of W, who developed TinkerCell and modified it to teach the bacterial photography system.
The system itself was made by the 2006 Univ. Texas iGEM team.

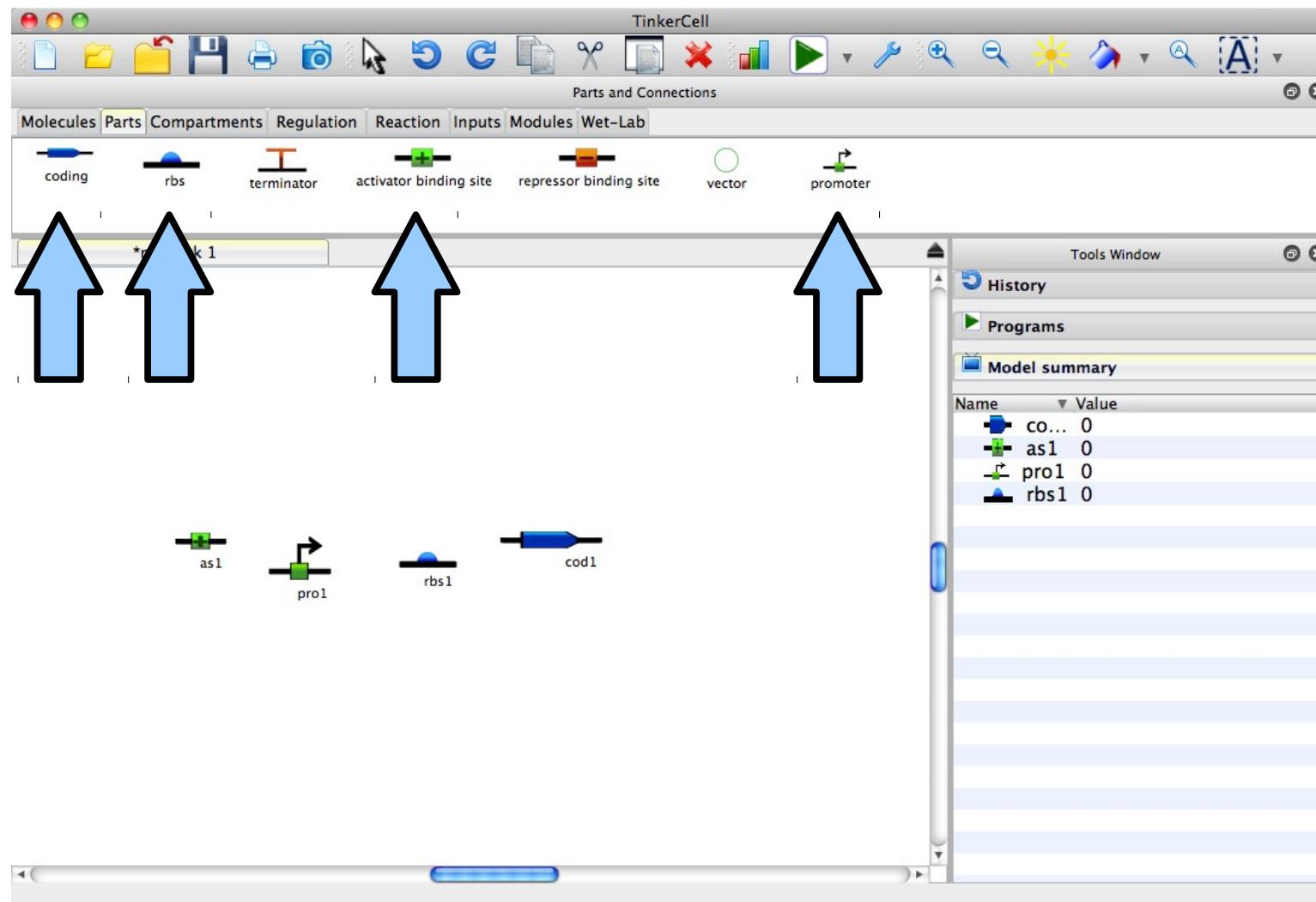
Start a new canvas

Select “New canvas” from the File menu or use the “New” icon at the left of the top toolbar



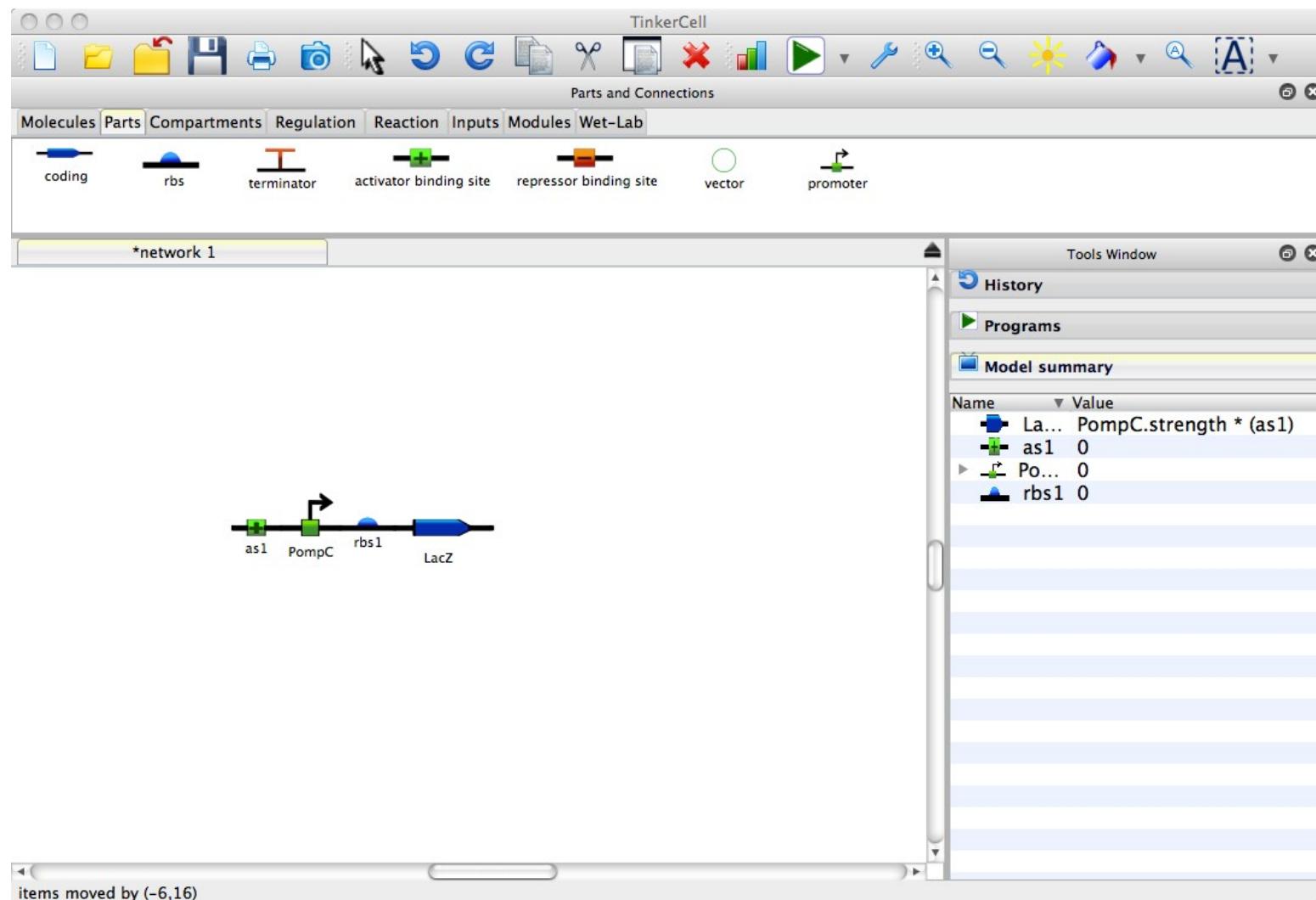
Assemble Reporter

From the “Parts” tab, select and place an “Activator Binding Site”, “Promoter”, “RBS”, and “Coding” icon on the canvas.



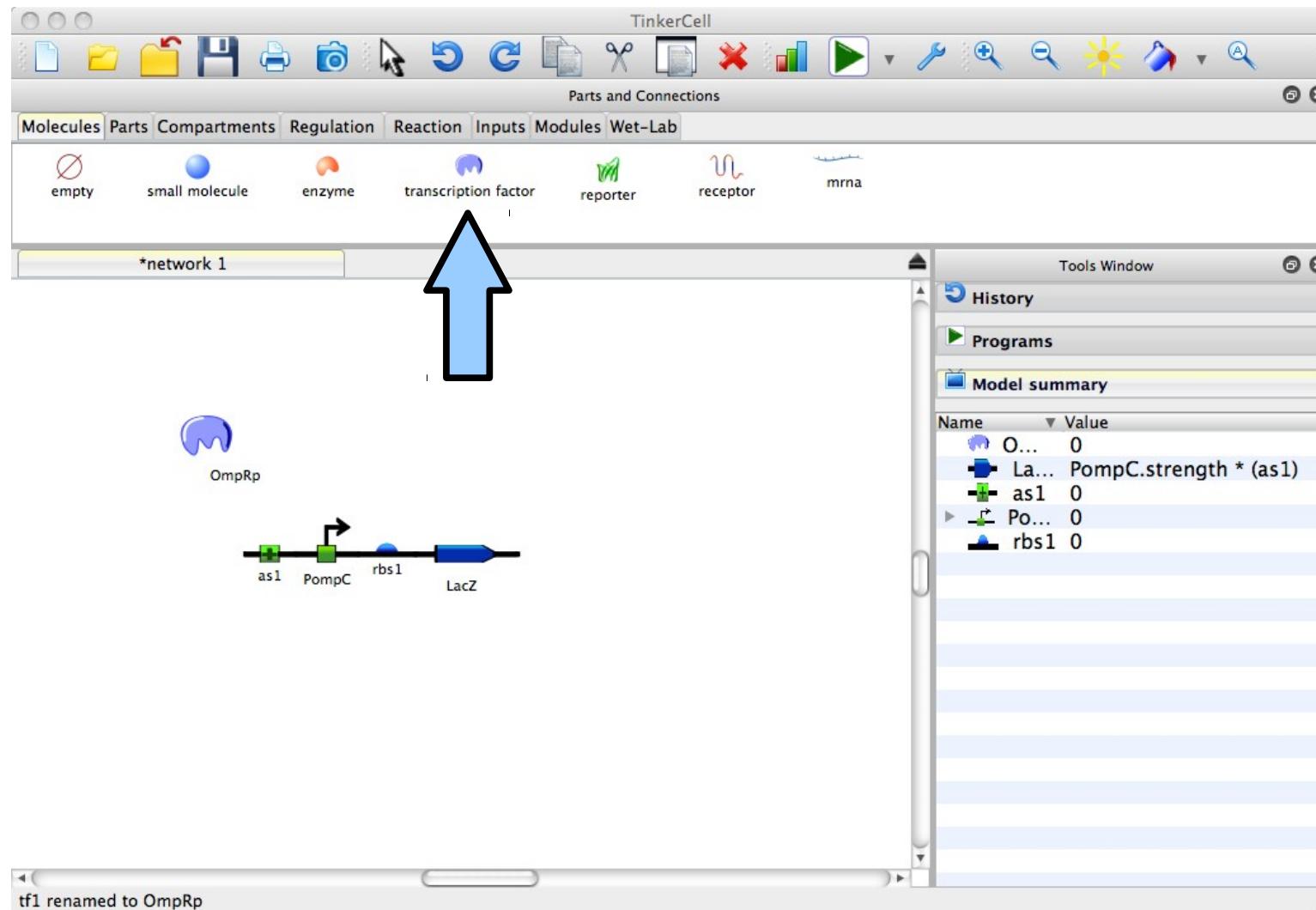
Align and Name

Drag items on the canvas next to one another to align them. Rename the promoter “PompC” and the coding sequence “LacZ”.



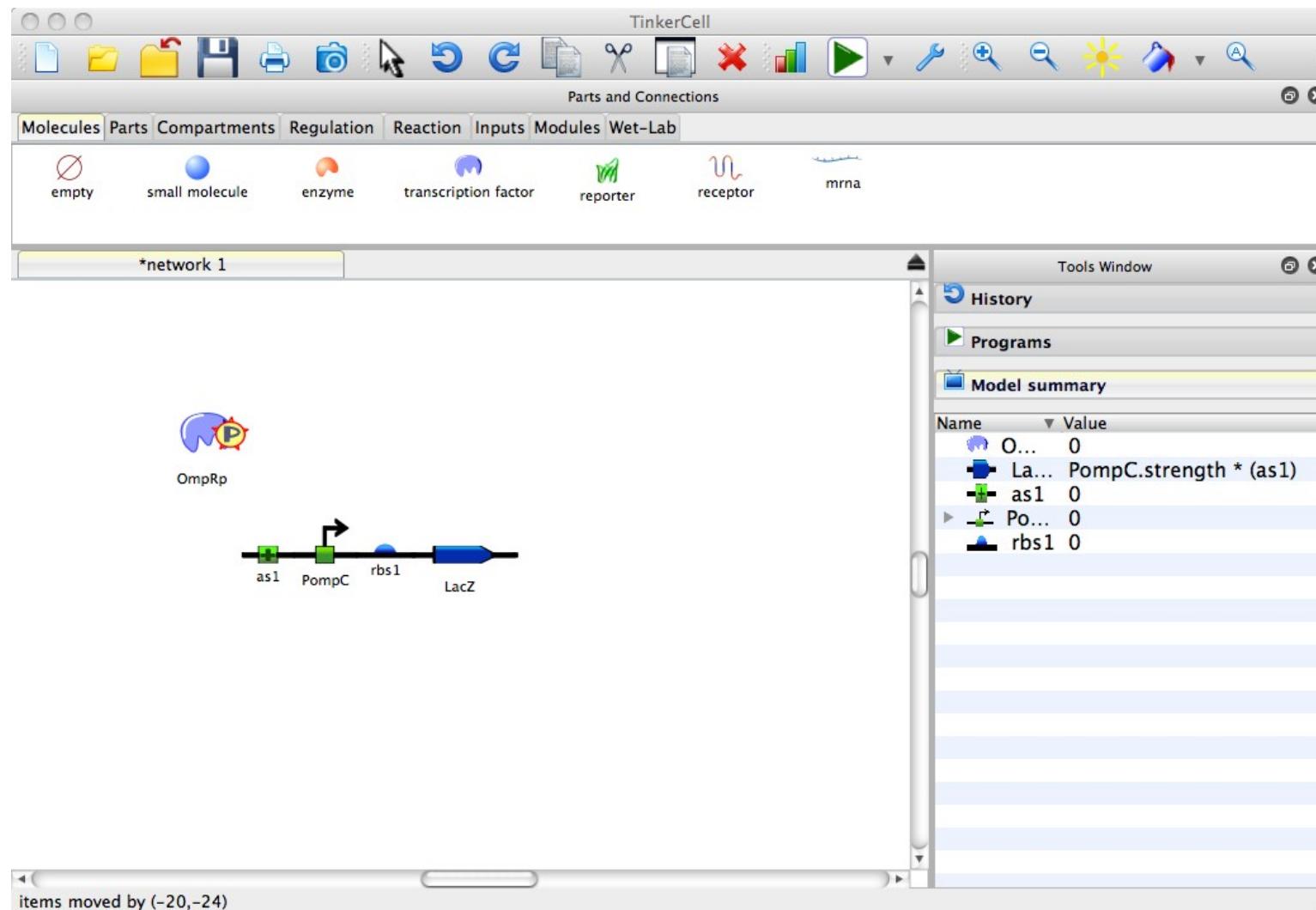
Add a transcription factor

From the “Molecules” tab, add a transcription factor to the canvas and rename it OmpRp



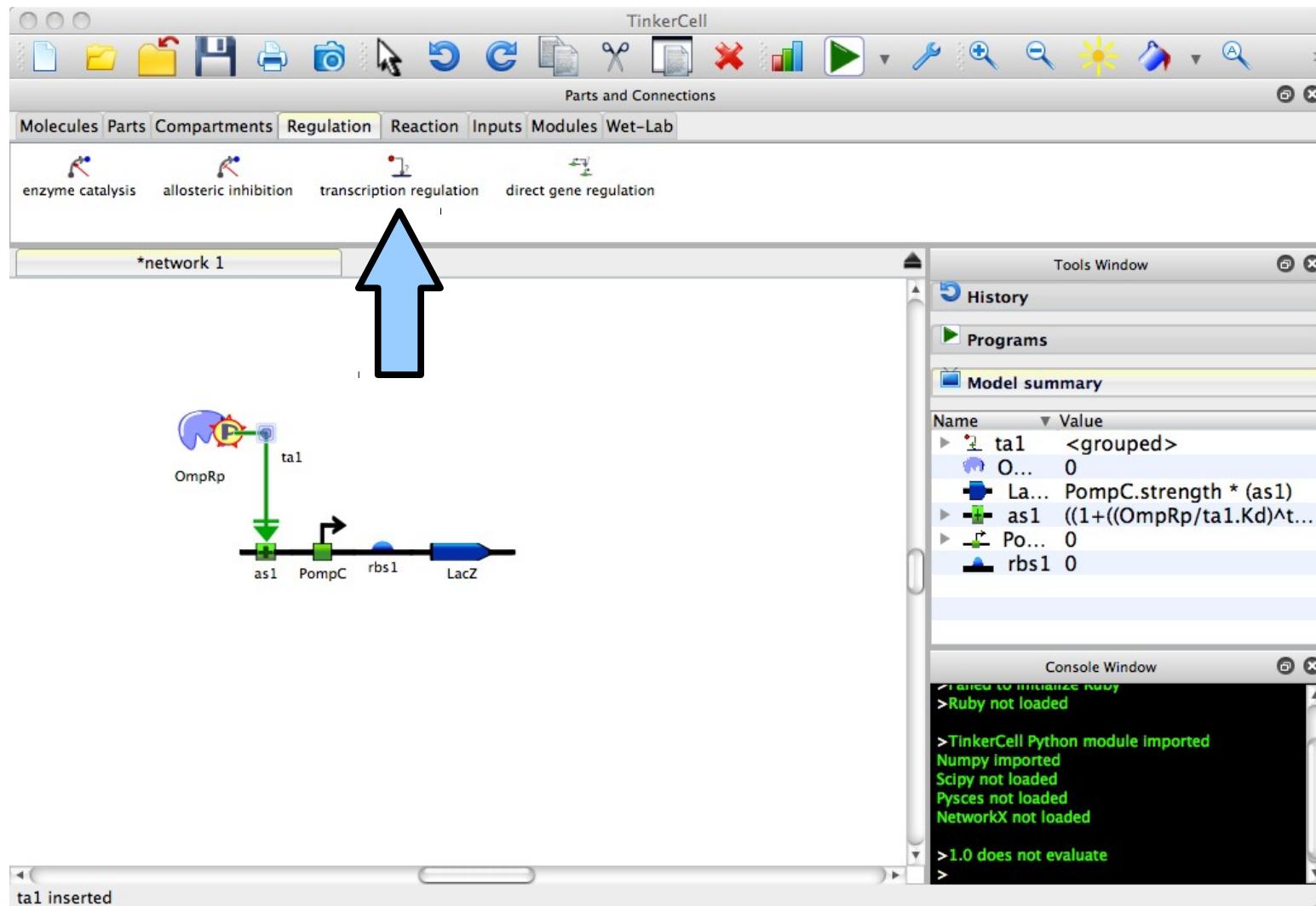
Visual appeal: add a 'P'

Select OmpRp, then from the “Edit” menu, choose “Add Decorator”. From the “Decorator” tab, select “phosphorylation”.



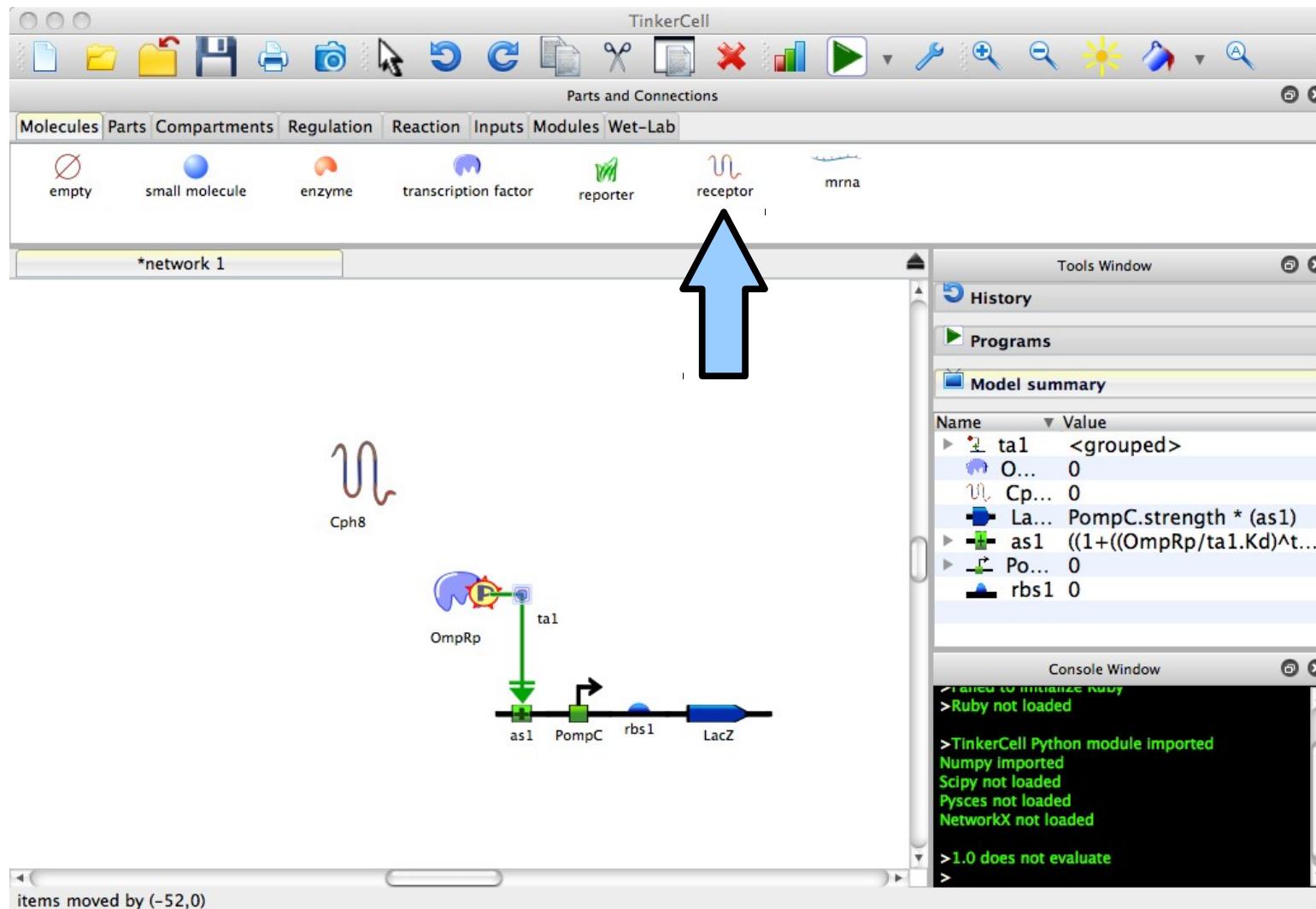
Regulate transcription

From the “Regulation” tab, choose “transcription regulation”, then click OmpRp and as1. Choose “transcriptional activation” from the pop-up.



Add light receptor

From the “Molecules” tab, place a “Receptor” on the canvas and rename it Cph8.



Regulate OmpRp with Cph8

From the “Regulation” tab, choose “Allosteric Inhibition” and then click on Cph8 and OmpRp.

The screenshot shows the TinkerCell software interface. At the top, there is a toolbar with various icons. Below the toolbar, a menu bar has the "Regulation" tab selected. A large blue arrow points upwards from the bottom of the screen towards the "Regulation" tab. The main workspace displays a regulatory network diagram. On the left, there are four categories: "enzyme catalysis", "allosteric inhibition", "transcription regulation", and "direct gene regulation". The "allosteric inhibition" icon is highlighted. The network diagram shows a protein labeled "Cph8" (represented by a red wavy line) activating an allosteric inhibitor "ai1" (a purple square with a red circle). "ai1" inhibits the "OmpRp" protein (a blue oval with a red circle). "OmpRp" is bound to a promoter labeled "PompC" (a green rectangle). The PompC promoter drives the expression of a gene labeled "LacZ" (a blue rectangle). Below the workspace, a status bar says "ai1 inserted". To the right of the workspace are two tool windows: "Tools Window" and "Console Window". The "Tools Window" contains sections for "History", "Programs", and "Model summary". The "Model summary" section lists parameters and variables:

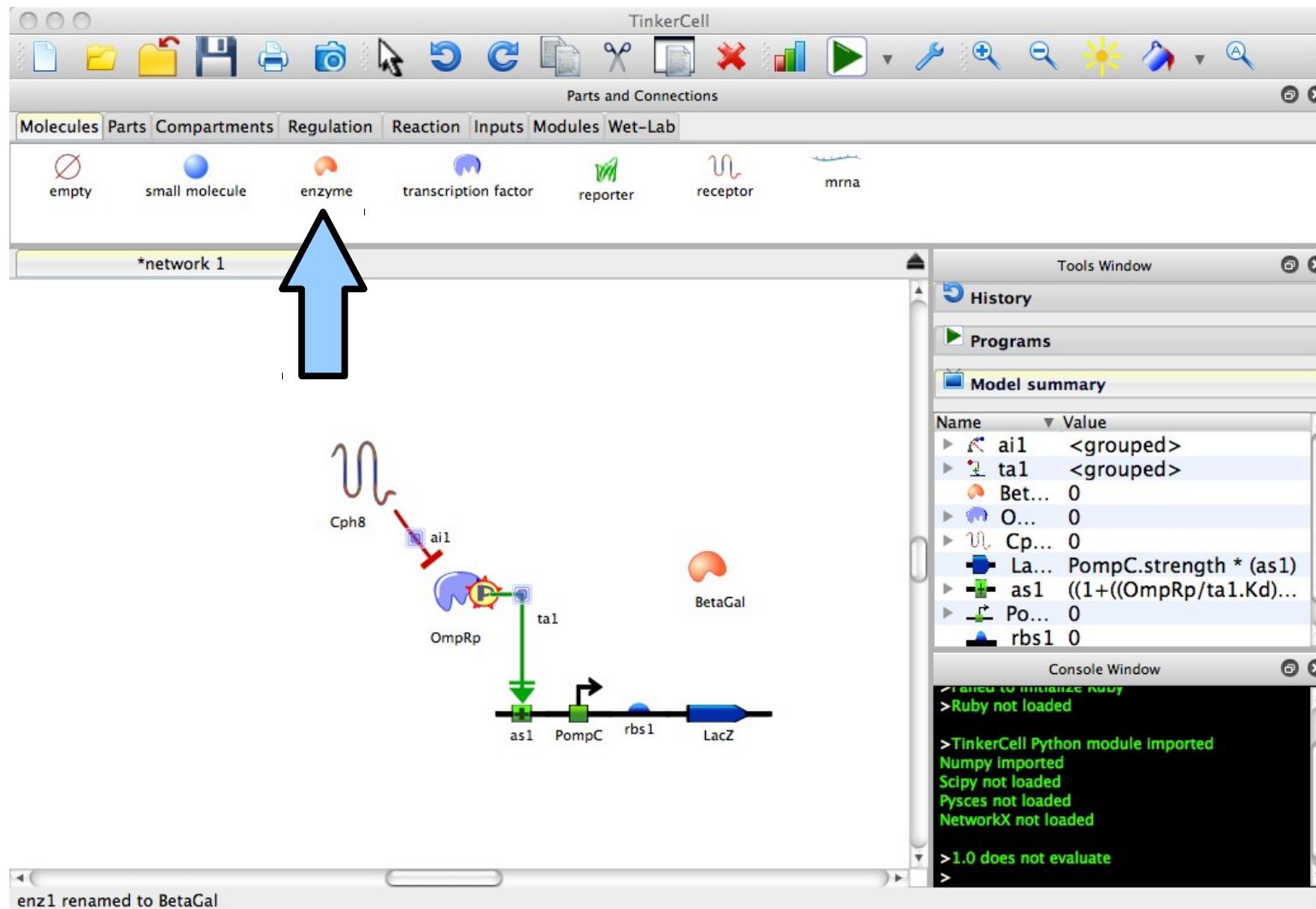
Name	Value
ai1	<grouped>
ta1	<grouped>
O...	0
Cp...	0
La...	PompC.strength * (as1)
as1	((1+((OmpRp/ta1.Kd)^t...
Po...	0
rbs1	0

The "Console Window" shows the following text:

```
Failed to initialize Ruby
>Ruby not loaded
>TinkerCell Python module imported
Numpy imported
Scipy not loaded
Pysces not loaded
NetworkX not loaded
>1.0 does not evaluate
>
```

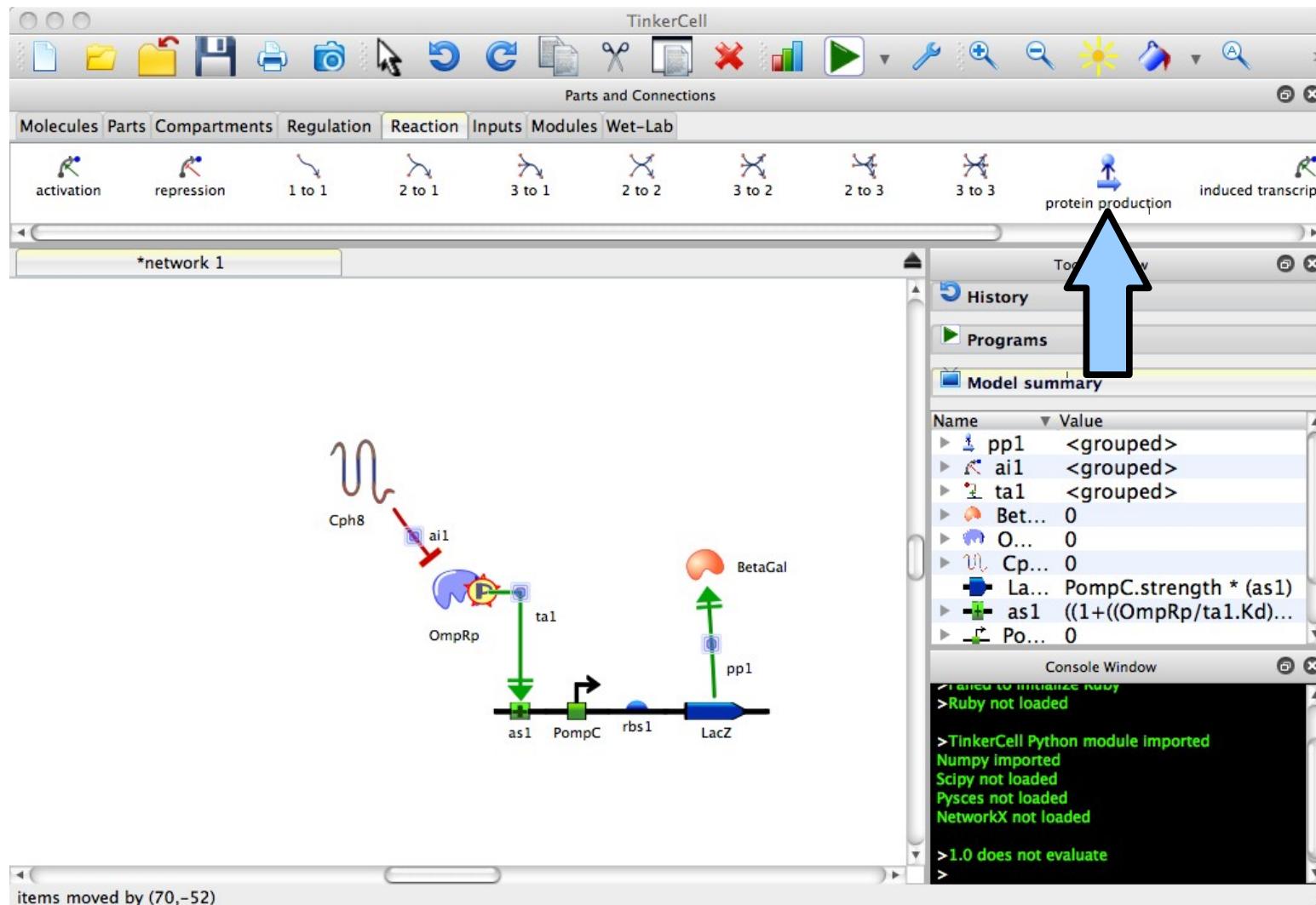
Add the BetaGal protein

From the “Molecules” tab, place an “Enzyme” on the canvas. Name it “BetaGal”.



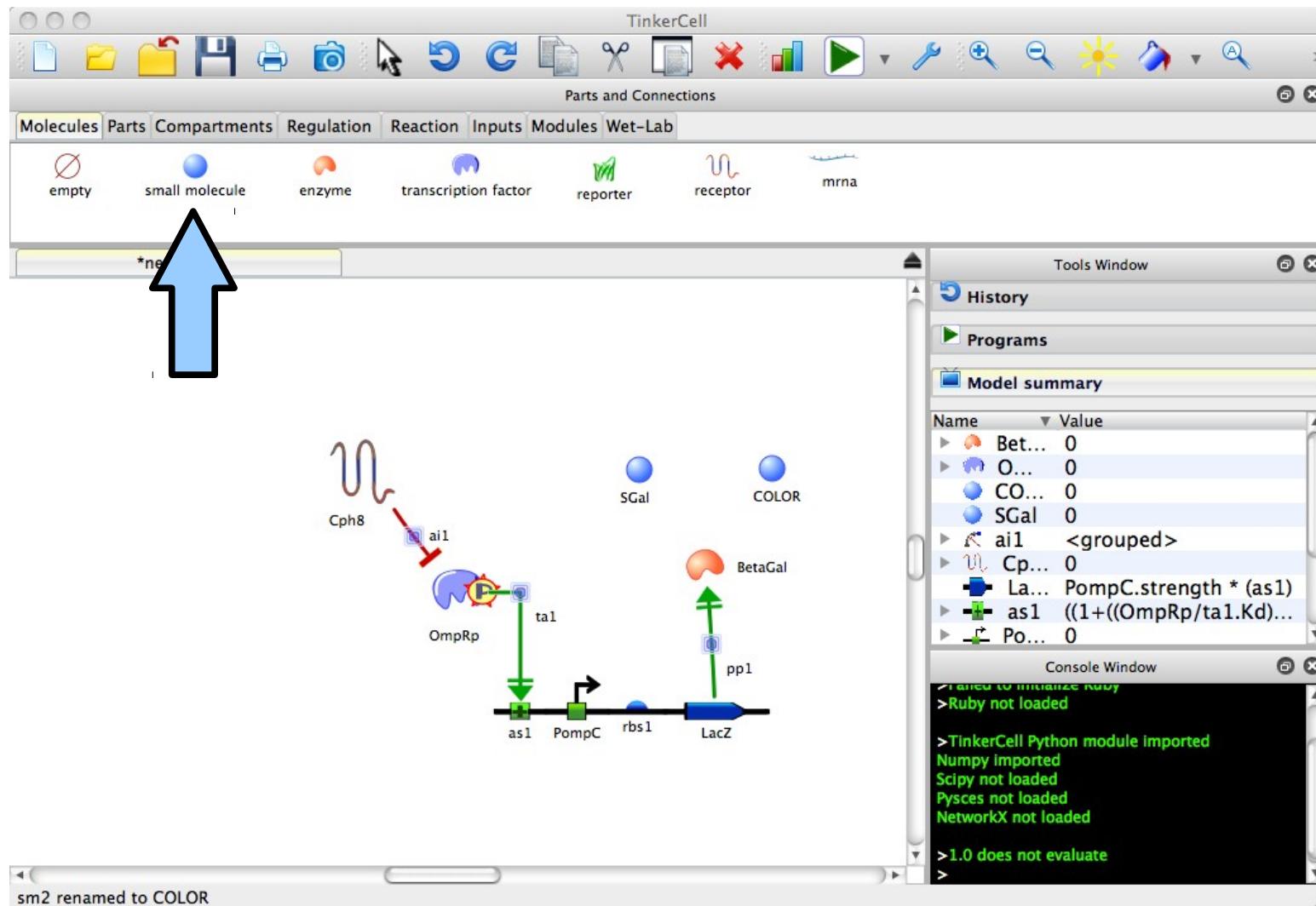
Produce BetaGal protein

From the “Reactions” tab, choose “Protein Production”. Link the LacZ coding region to the BetaGal enzyme.



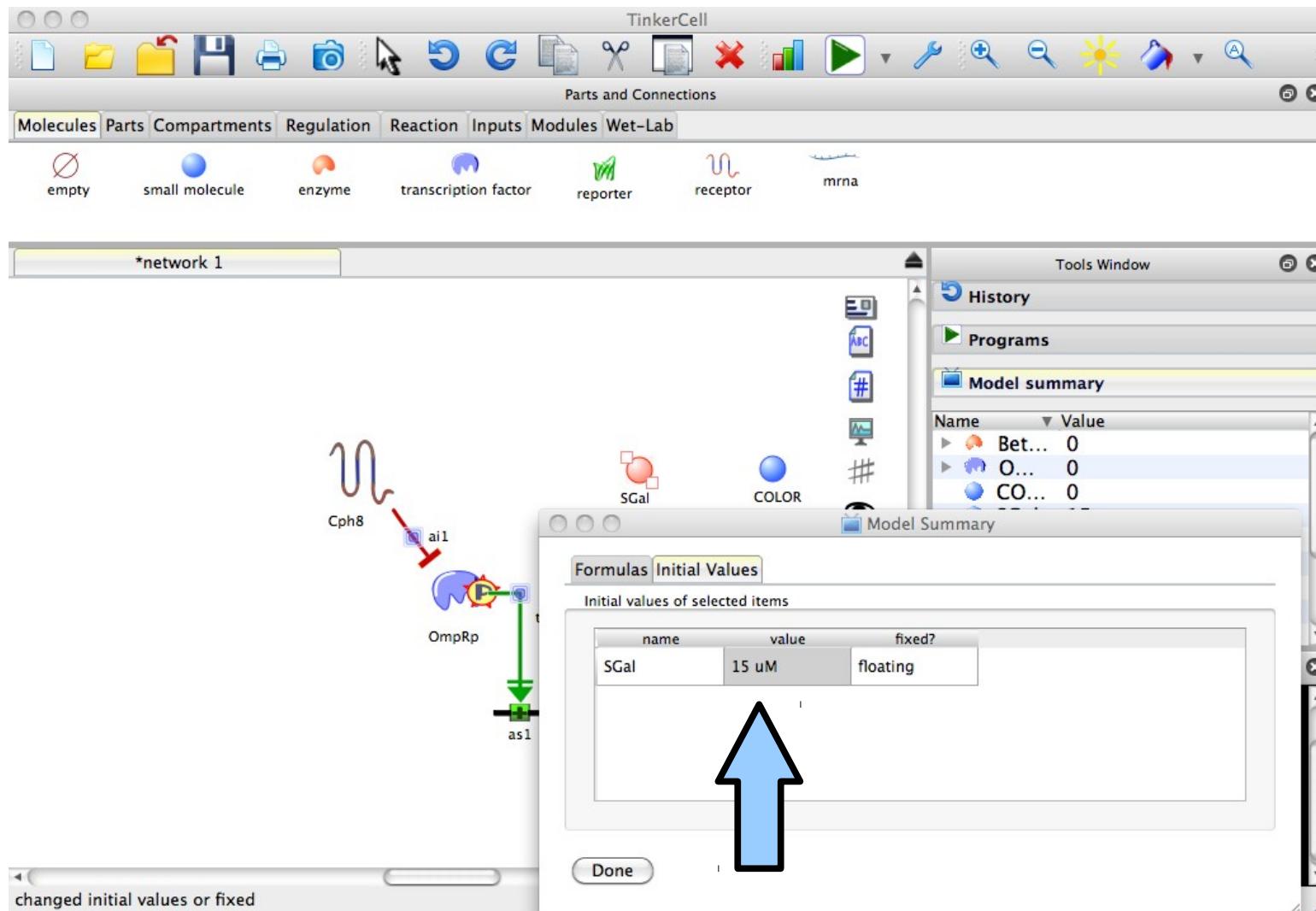
Add the colorful molecule

From the “Molecules” tab, place two “small molecules” on the canvas. Name the first “SGal” and the second “COLOR”.



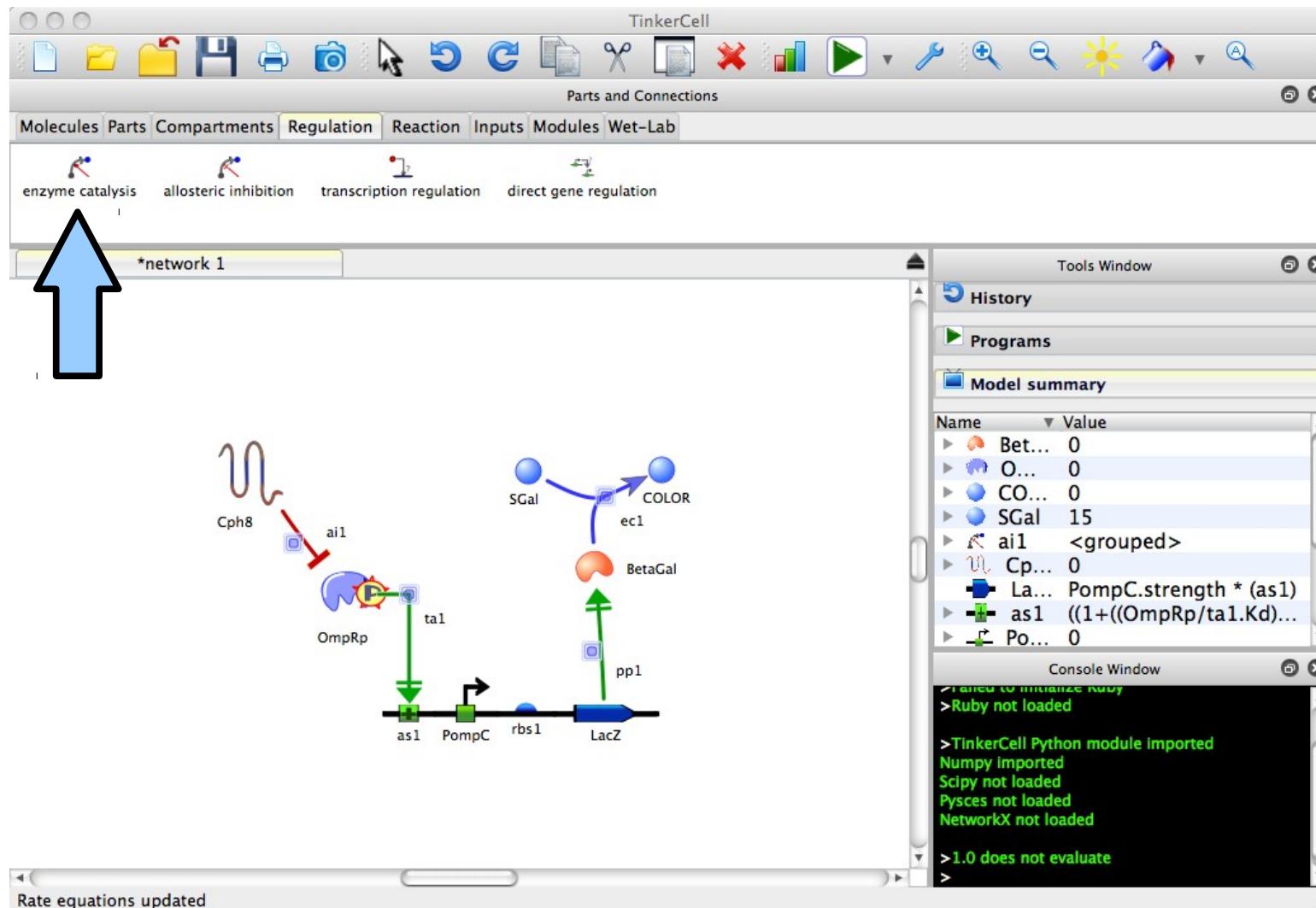
Supply SGal

Double-click on SGal, select the “Initial Conditions” tab, and change its initial concentration from 0 to 15 uM.



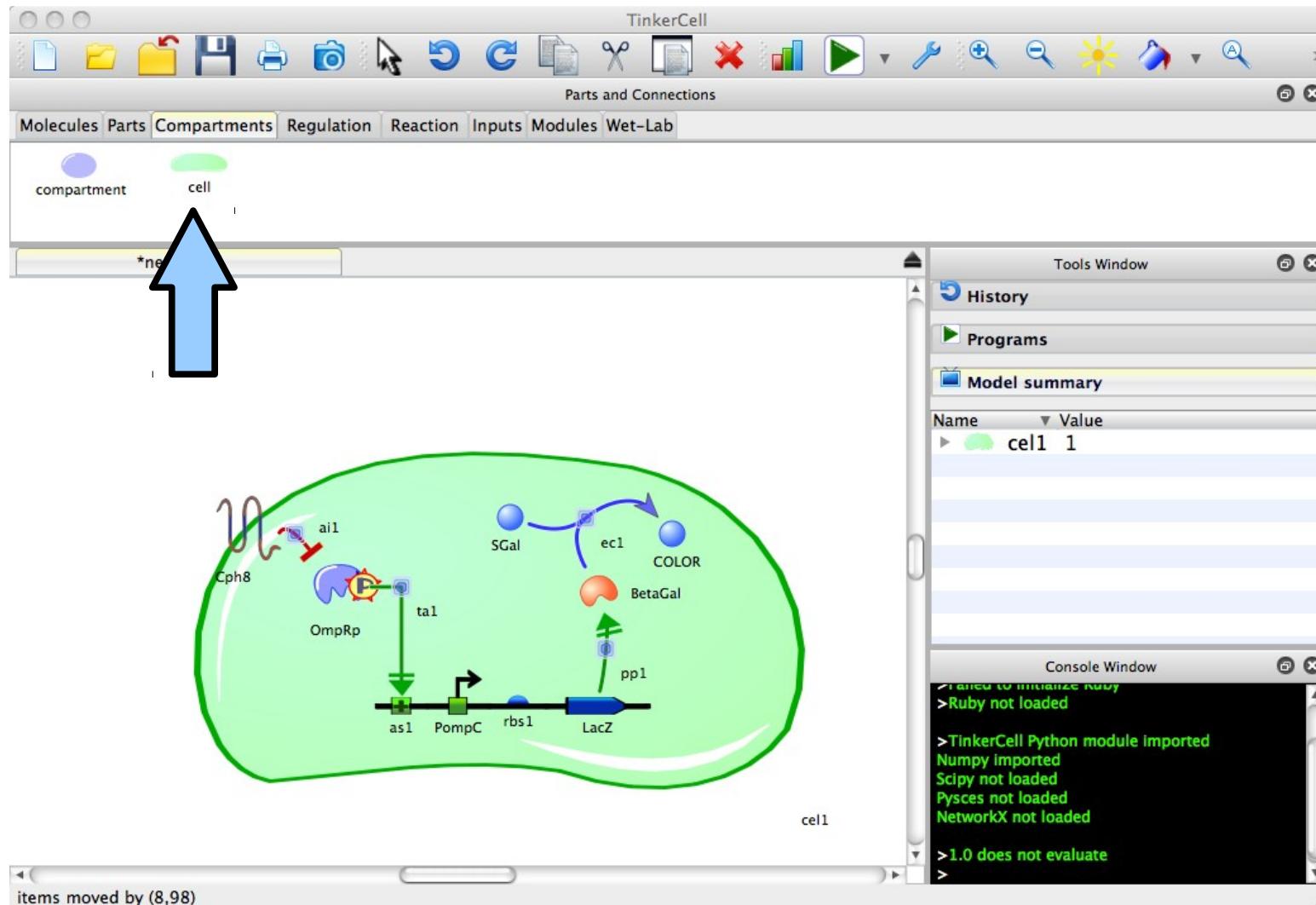
Add the color reaction

From the “Regulation” tab, choose “Enzyme Catalysis”, then click on BetaGal, SGal, and COLOR.



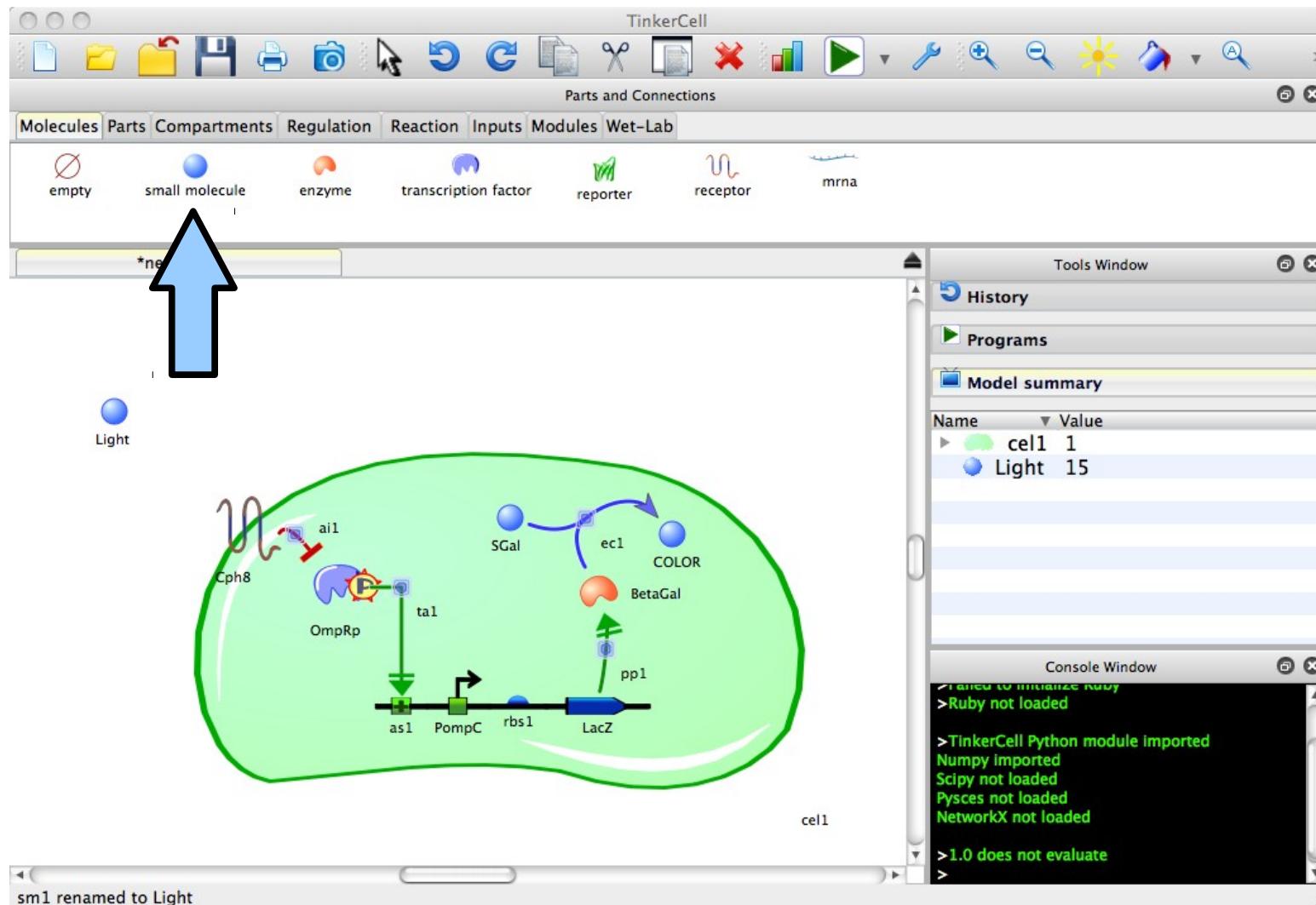
Add a chassis

From the “Compartments” tab, choose “Cell”. Place and resize it on the canvas to include all components (but leave Cph8 in the membrane).



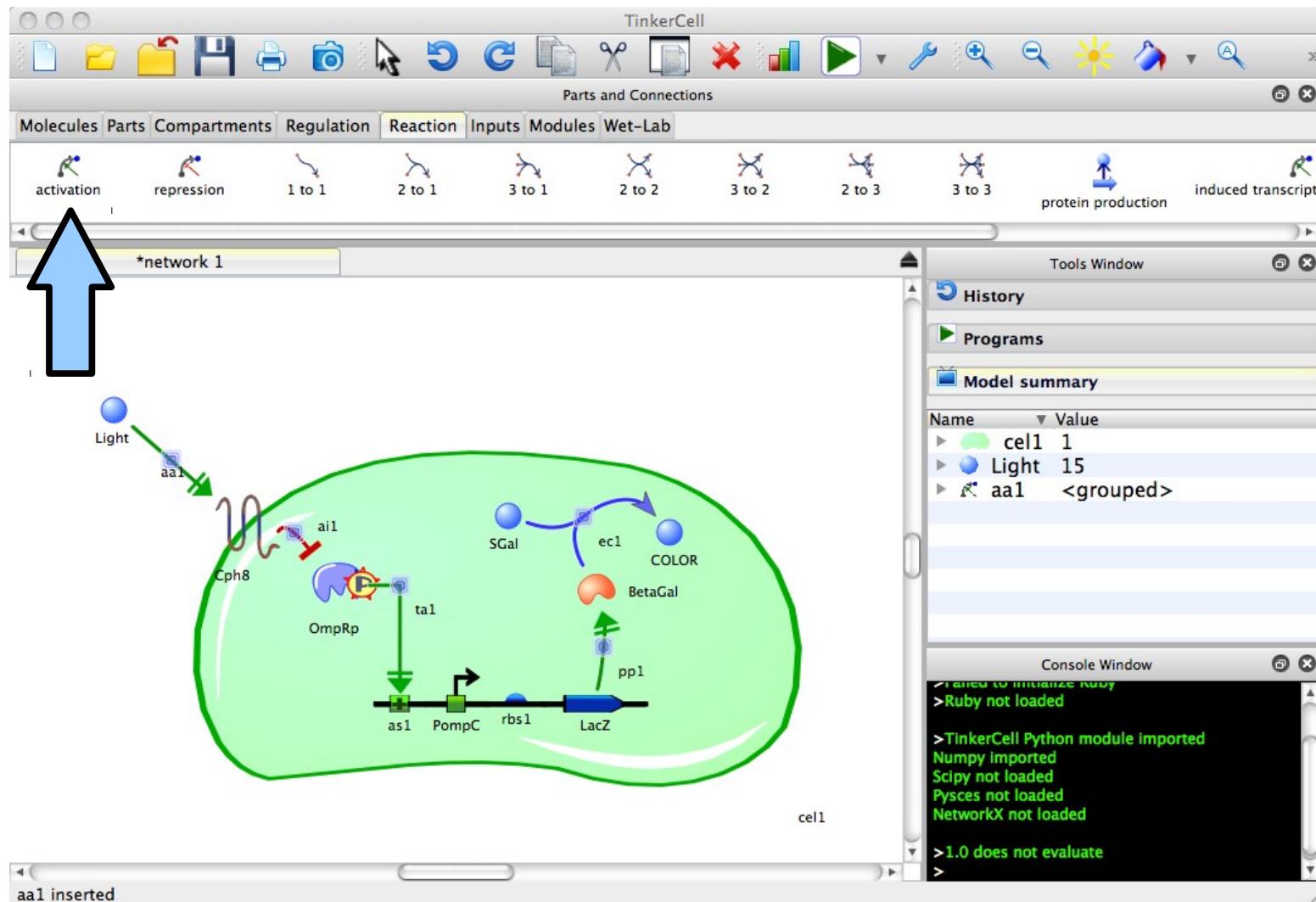
Add light

From the “Molecules” tab, place a small molecule outside the cell and rename it “Light”.



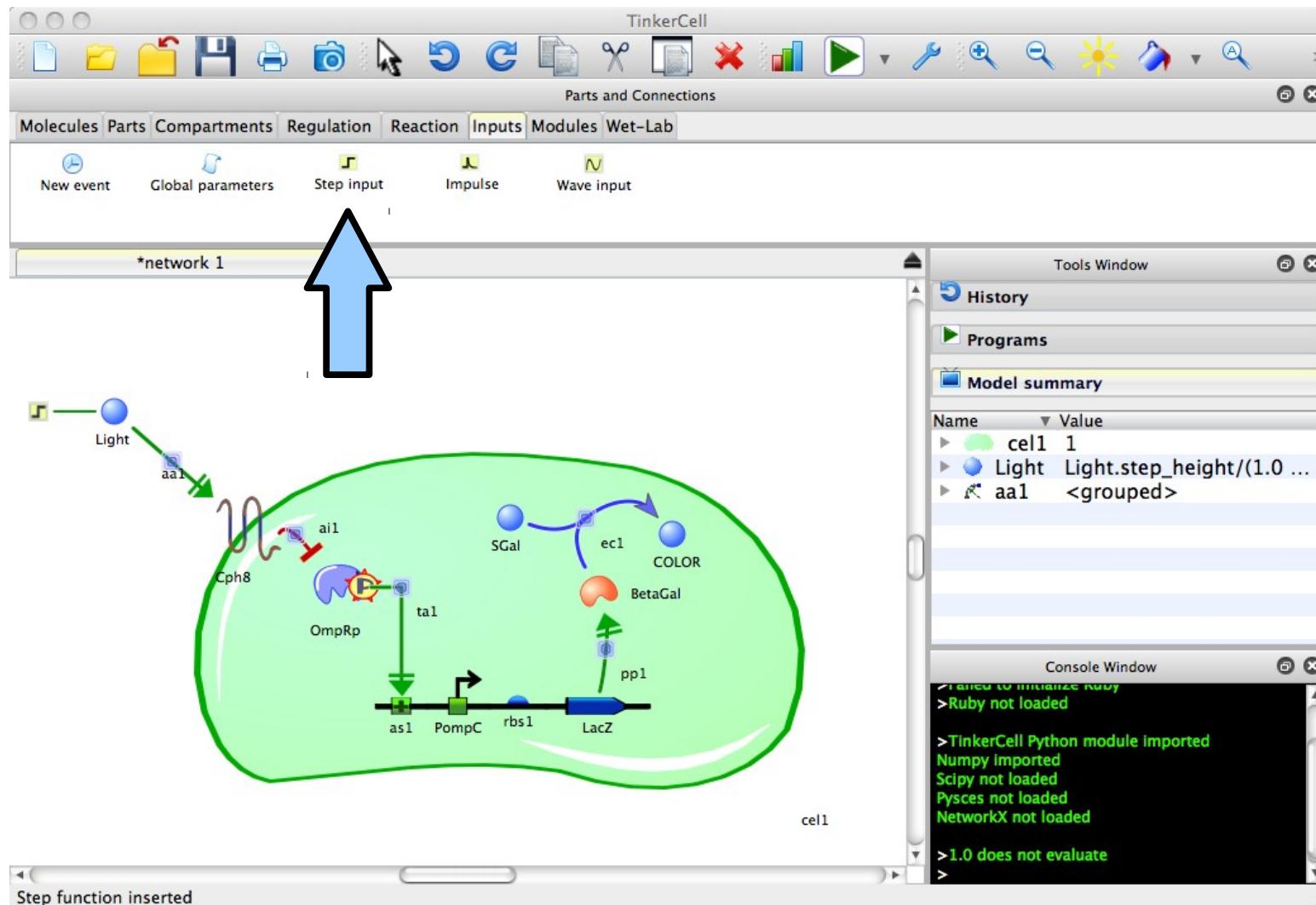
Regulate Cph8 with light

From the “Reaction” tab, choose “Activation” and click on Light and Cph8.



Turn on the light

From the “Inputs” tab, choose “Step input” and click on the Light molecule.



Change the timing

In the “Model summary” menu on the right, find “Light” and click the arrow to expand. Change “Step_time” to 50.

The screenshot shows the TinkerCell interface with a model named "network 1". On the left, a green cell contains a genetic circuit. An input "Light" activates a receptor "aa1", which triggers an "Impulse" signal. This signal activates gene "tal", leading to protein "tal". "tal" activates promoter "PompC", which drives gene "as1". "tal" also activates gene "rbs1", which drives gene "LacZ". "tal" inhibits gene "ail", which is part of a two-component system "OmpRp". "ail" activates gene "ec1", which drives gene "pp1". "pp1" activates gene "ecl", which drives gene "SGal". "SGal" activates gene "BetaGal", which produces the product "COLOR". A blue arrow points to the "Model summary" tab in the Tools Window on the right, which lists parameters and their values. The "Light" parameter is expanded, showing its step function definition. The "step_time" value is highlighted and set to 50.

Name	Value
cell1	1
Light	Light.step_h...
molecularweight	50000
step_height	1
step_time	50
step_steepliness	4
a1	<grouped>

Light.step_time = 50