

# CPIB SUMMER SCHOOL 2011: INTRODUCTION TO BIOLOGICAL MODELLING

## Building and simulating multi-variable models

Markus Owen

### 1 Outline

In the previous practical you began to use CellDesigner (<http://celldesigner.org/>) to build and simulate simple models, combined with graphical analysis to predict the behaviour of those models.

Here, we will build on that experience by constructing multi-variable models. With two dynamic variables we can use the *phase-plane* to predict a lot about the system dynamics. We'll practice that with a model for a genetic toggle switch (Section 2) and for a negative feedback loop (Section 3). Then we will construct an extended negative feedback model with three variables, and show that it can oscillate.

In University of Nottingham computer labs, launch CellDesigner via the "Start" menu: Start > All Programs > UoN Software > School of Mathematics > CellDesigner

### 2 Toggle switch

In this section you will construct a simple model for a genetic toggle switch, as discussed in the lectures. The structure of the network is shown in Figure 1, and the model equations are:

$$\frac{du}{dt} = \frac{\alpha_1}{1 + v^\beta} - u, \quad \frac{dv}{dt} = \frac{\alpha_2}{1 + u^\gamma} - v. \quad (1)$$

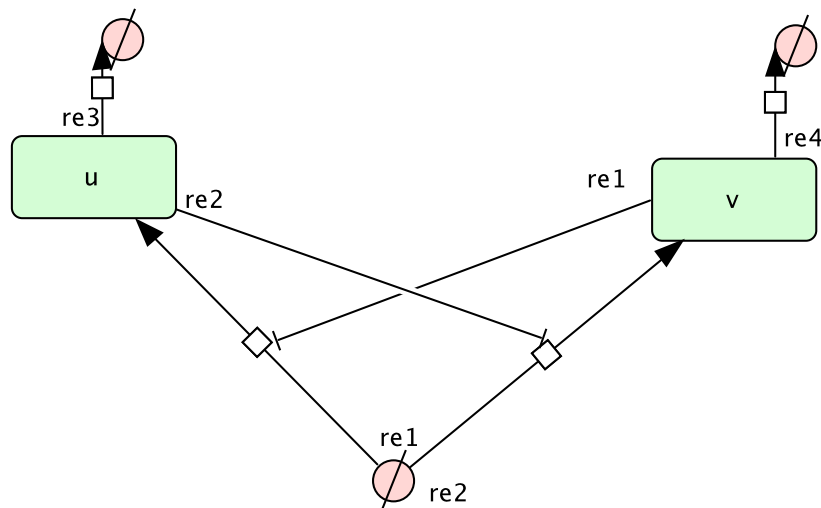


Figure 1: Schematic of the genetic toggle switch.  $u$  and  $v$  repress each other's production (reactions "re1" and "re2"), and both degrade (reactions "re3" and "re4").

## 2.1 Creating the model in CellDesigner

Follow the instructions below to create the model in CellDesigner. If you have difficulty, you can download `toggle_mro.xml` from the course web-page.

1. Start a new model.

Click 'File', 'New' and "OK". Then 'File', 'Save' and save it as e.g. `toggle`. Make sure you save the file periodically during the steps below.

2. Create the source from which proteins are produced.

Click the last icon on the Species toolbar (the 'Degraded' species):



then click on the canvas where you want to place it. A dialog should pop up, asking for a name for the species (e.g. 'source').

3. Create the proteins u and v.

Click protein (the first entry on the Species toolbar), click where you want it on the diagram, and then name the protein u; repeat for v.

4. Create transitions from the source to u and v, representing production of u and v.

Click 'state transition', the first arrow on the Reaction toolbar:



then click on the canvas at the starting point, 'source', and target, u; repeat for v. You can name these reactions by clicking the 'Reactions' tab and the double-clicking the 'name' column to enter text (e.g. 'u-production').

5. Make these production terms repressed.

Click 'Inhibition', third icon on the Modifier section of the Reaction toolbar:



then click the u icon on the canvas, followed by the v production reaction (to implement u repressing v). Repeat for v inhibiting u production.

6. Now implement degradation of u and v.

Click "Degradation" in the toolbar () , then click the u icon on the canvas; repeat for v.

At this stage, your canvas should look something like Figure 1. This includes the qualitative structure of the network, but not the details of the reaction rates.

7. Right click on reaction 1 (this should be production of u, inhibited by v), and click "Edit KineticLaw". This opens a window showing the reaction of interest. In the 'math' textbox, set the Kinetic Law to  $\alpha_1 / (1 + s_3^\beta)$ . Note that  $s_3$  here refers to species 3, which should be v (assuming you have created species in the same order as in `toggle_mro.xml`). Remember that, when entering expressions for rate laws, you can click on the relevant species to find its id. Click 'update', 'close', and 'close' in the reaction window.

Now do the same for reaction 2 (this should be production of v, inhibited by u). The KineticLaw should be set to  $\alpha_2 / (1 + s_2^\gamma)$ .

The parameters also need adding. Click the "Parameters" tab and then the "New" button. In the Parameter dialog, specify id, name, value and units for each new parameter. You can use the same label for id and name. Do this for  $\alpha_1$ ,  $\alpha_2$ ,  $\beta$  and  $\gamma$  (setting  $\alpha_1 = \alpha_2 = 3$ ,  $\beta = \gamma = 2$ ).

For degradation, edit the KineticLaw and click the species that is degraded (so the rate is the same as its amount, e.g. the rate expression in the KineticLaw for u-degradation should be simply “s2”).

Save the file, “File->Save”, or “Ctrl-S”.

## 2.2 Simulations using the “ControlPanel”

Click “Simulation”, then “ControlPanel” in the main menu bar. Change “End Time” to 40, then click “Execute”. You should get a plot of the result with three curves, even though, at this stage, your model has five variables:  $u$  and  $v$  are identical since they have identical initial values and dynamics (their corresponding parameter values are the same), so you only see one curve (the other is underneath). The same is true for the two degraded species. If you click on a species in the panel to the right of the plot, the relevant curve flashes.

As for the Positive Feedback model, we need to set the “boundaryCondition” for the source and degraded species to “true”. Double click the “b.c.” entries in the “Species” tab, or right click on the species and select “Edit Species”, and then set “boundaryCondition” to “true”. We also need different initial conditions for  $u$  and  $v$  to allow their values to diverge. Double click the “InitialQuantity” entry for  $u$  in the “Species” tab, and set this to be 1 rather than the default value of zero.

Save the file, “File->Save”, or “Ctrl-S”.

Now click “Execute” again. You should now see just two curves, for  $u$  and  $v$ , but this time  $u$  and  $v$  go to different values ( $u$  high and  $v$  low).

From the lectures, we expect that different initial conditions will give different outcomes (so the switch can have  $u$  high, or  $v$  high). We can check this by changing the initial value of  $v$  to be higher than that for  $u$ . In the ControlPanel, click the  $v$  “InitialValue”, and change it to 1.1. Then click “Execute” - this time you should see  $u$  go down, and  $v$  up (see Figure 2).

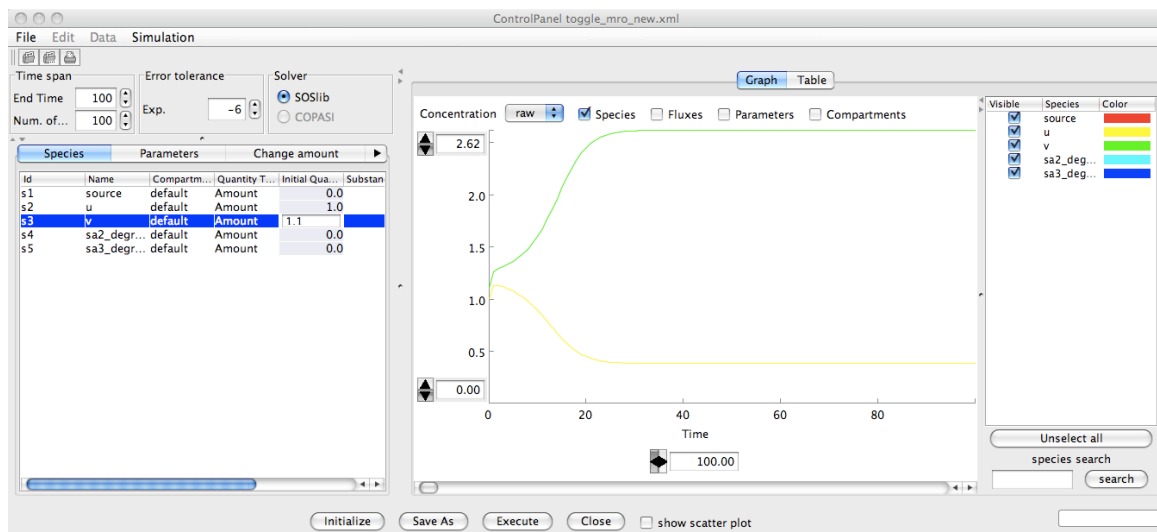


Figure 2: Simulation of the genetic toggle switch model given by equation (1) and illustrated in Figure 1. With  $u(t = 0) = 1$  and  $v(t = 0) = 1.1$ , so that  $v$  goes to the high steady state. Parameter values:  $\alpha_1 = \alpha_2 = 3$ ,  $\beta = \gamma = 2$ .

Now lets see the result for a range of initial values for  $v$ , using the “Parameter Scan” tab. Click the “Scan parameter” checkbox, make sure “Initial value” is selected, change the “Name” to ‘v’, change “To” to 2, then click “Execute” and click the “Results” tab. Click on ‘Overlay Setting’ and select ‘v’ only, then “OK”, then “Show Graph”. You should see a set of curves, one for each initial

value of  $v$ . Initial values with  $v < u$  (which has an initial value of 1) end up at the low steady state, and those with  $v > u$  go to the high steady state (see Figure 3).

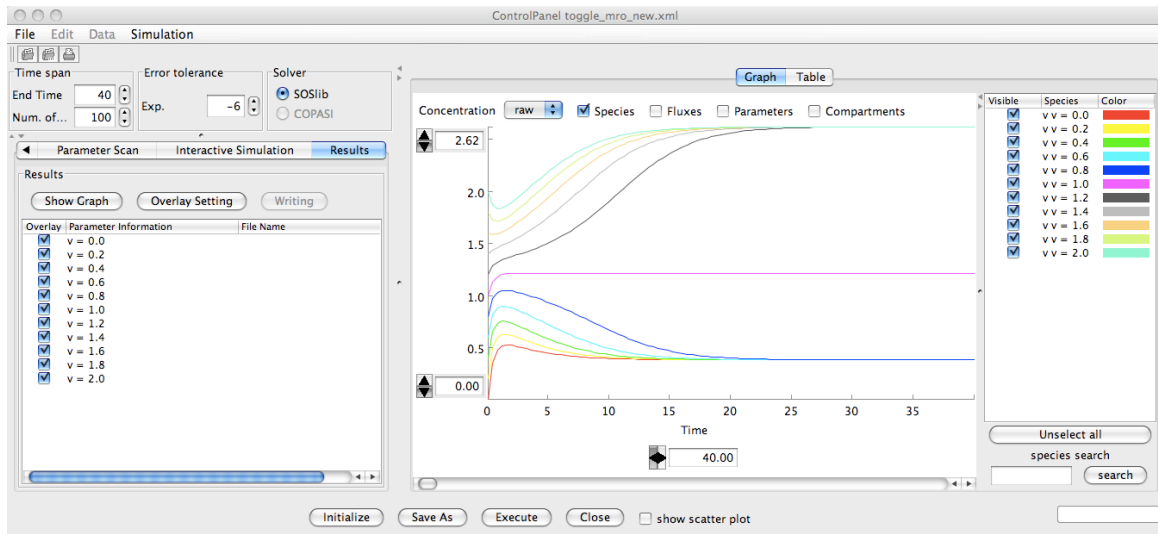


Figure 3: Simulation of genetic toggle switch model, with  $u(t = 0) = 1$  and  $v(t = 0)$  varying from 0 to 2, so that  $v$  goes to a high or low steady state.

Alternatively, we could consider what happens as a parameter varies. Click the “Scan parameter” checkbox, select “Parameter value”, change the “Name” to  $\alpha_1$ , change “To” to 4 and “Interval” to 0.2, then click “Execute” and click the “Results” tab. Click on “Show Graph”. You should see a set of curves, one for each value of  $\alpha_1$ . Values of  $\alpha_1$  less than or equal to 3 give high  $v$ , and values greater than 3 end up at the low  $v$  steady state (see Figure 4).

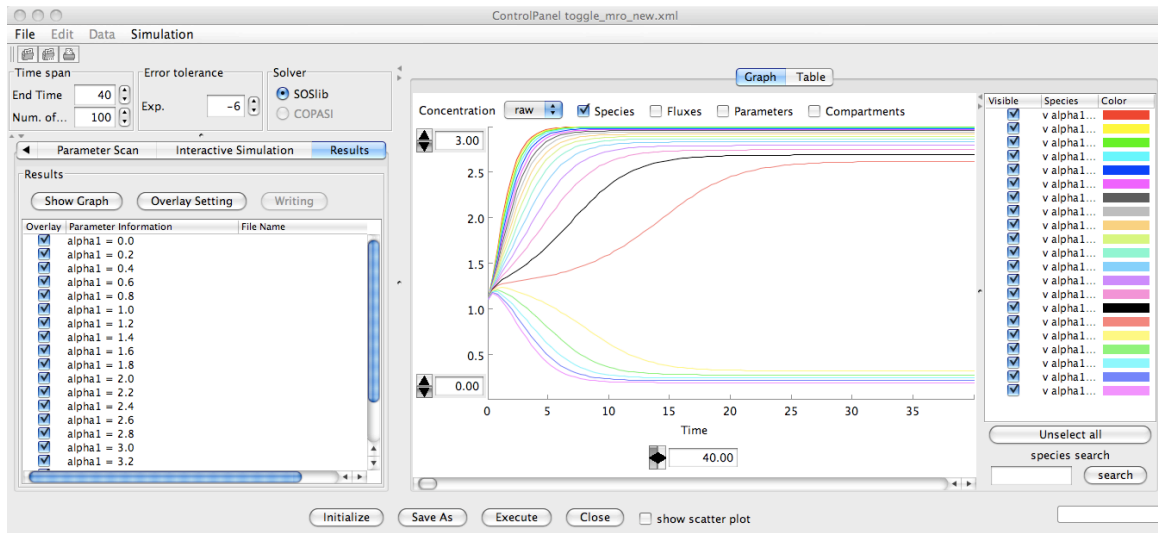


Figure 4: Simulation of genetic toggle switch model, with “ $\alpha_1$ ” varying from 0 to 4, so that  $v$  goes to a high or low steady state.

### 2.3 Phase plane analysis

In Lecture 2.2 we considered the *phase-plane* for this system, and showed how the dynamics you have seen in these simulations can be predicted using the *nullclines*. Review the lecture material, in the light of your simulations. Why does sufficiently high  $\alpha_1$  guarantee that  $v$  must go to a low value?

### 3 Single negative feedback

#### 3.1 Direct repression of mRNA transcription by protein monomers

Many transcription factors negatively regulate their own production (by transcriptional repression of their own gene). A simple way of representing the state of this system considers two variables, mRNA concentration and protein concentration:

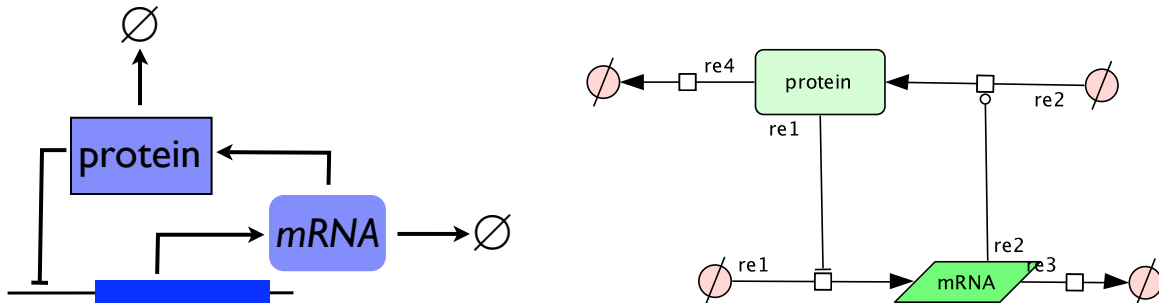


Figure 5: Negative feedback loop for a single gene: mRNA transcription is repressed by translated protein. The right-hand figure shows the implementation in CellDesigner.

A simple model for this system represents four processes:

- i) production of mRNA by transcription
- ii) degradation of mRNA
- iii) production of protein by translation
- iv) degradation of protein.

Representing the concentration of mRNA and protein by  $m$  and  $p$  respectively, an ODE model for this system is:

$$\frac{dm}{dt} = \frac{k_1 p_0^n}{p_0^n + p^n} - d_1 m, \quad \frac{dp}{dt} = k_2 m - d_2 p. \quad (2)$$

1. Give interpretations for the parameters  $k_1$ ,  $p_0$ ,  $n$ ,  $d_1$ ,  $k_2$  and  $d_2$ .
2. Write down and sketch the nullclines for this system. It is easiest to consider  $p$  (protein) on the horizontal axis, and  $m$  (mRNA) on the vertical axis, as in Lecture 2.2, slides 9-10. Can you see from this that there can only be a single steady state? Try to work out the direction of trajectories on the nullclines and the phase-plane axes. You should be able to see that solutions can overshoot the steady state.
3. What does increasing  $p_0$  do to the shape of the  $m$ -nullcline? Recall that  $p_0$  is the value of  $p$  at which the “repression function”  $\frac{k_1 p_0^n}{p_0^n + p^n}$  is half its maximum. Use this to explain how increasing  $p_0$  changes the steady state.
4. Try to construct this model in CellDesigner.  
When adding species, you can use the specific “Species” icon for mRNA, the 6<sup>th</sup> entry on the species toolbar. Create a source for mRNA, the mRNA itself, a source for protein, and the protein (you should then have four species).

Then add “transition” reactions from the mRNA source to mRNA, and from the protein source to protein.

Then make mRNA production inhibited by protein, using the third icon on the Modifier section of the Reaction toolbar:



Make protein production *catalysed* by mRNA, using the first icon on the Modifier section of the Reaction toolbar.

Add degradation of mRNA and protein, using  $\rightarrow\phi$ . After this you should have 6 species, and your network should look something like that in Figure 5.

You should have four reactions, which need associated kinetic laws (from the model equations): one for production (transcription) of mRNA ( $m$ , whose KineticLaw depends on  $p$ ), one for production of protein ( $p$ ) via translation (whose kinetic law depends on  $m$ ), and two degradation reactions. For example, the KineticLaw for transcription should be of the form  $k_1 * p_0^n / (p_0^n + s_3^n)$ , where “s3” is the id of the protein species.

With the parameter values  $k_1 = 50$ ,  $p_0 = 10$ ,  $n = 10$ ,  $d_1 = 0.1$ ,  $k_2 = 1$ ,  $d_2 = 0.1$ , and initial conditions with  $m = 1$ ,  $p = 5$ , simulate the model up to an end time of 100. What do you notice about the system behaviour for these parameters? Show how different initial conditions end up at the same steady state.

If you cannot construct the model, you can download it as `NegativeFeedback.xml`.

**Note:** In the CellDesigner “ControlPanel” you can plot trajectories in the phase-plane by selecting just the two variables of interest in the list of variables, and clicking the “show scatter plot” checkbox.

5. Simulate for increasing  $p_0$  from 5 to 30 in steps of 5. For each value, keep a note of the steady state values of  $p$  and  $m$  (or you could use a “Parameter Scan”). How do the steady state concentrations depend on  $p_0$ . Does this agree with your phase-plane analysis?
6. Explore the effects of varying the other parameters in the model. For example, change  $d_1$  and  $d_2$  to 0.75 and note the more prominent overshoot of the steady state.

### 3.2 Repression of mRNA transcription by protein dimers

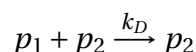
Many transcription factors have to dimerise in order to bind DNA (e.g. basic helix-loop-helix proteins). We can include this in the model by adding another variable to the model to represent protein dimers:

$$\frac{dm}{dt} = \frac{k_1 p_0^n}{p_0^n + p_2^n} - d_1 m \quad (3)$$

$$\frac{dp_1}{dt} = k_2 m - d_2 p_1 - 2k_D p_1^2 \quad (4)$$

$$\frac{dp_2}{dt} = k_D p_1^2 - d_3 p_2 \quad (5)$$

where  $m$ ,  $p_1$  and  $p_2$  represent mRNA, monomeric protein, and dimeric protein, respectively. In this model, mass action kinetics are used to represent the dimerisation reaction, and dimerisation is assumed to be irreversible with rate  $k_D$ :



Finally,  $d_3$  is the degradation rate of protein dimers.

- Using your implementation of the previous model (or “NegativeFeedback.xml”) as a basis, construct and simulate this dimerisation model in CellDesigner. You will need to add a “Homodimer formation” reaction, which is in the toolbar (just to the left of “Degradation”). Then delete the repression of transcription by protein, and replace it with repression by the dimer. Also add degradation of the dimer and the new parameters. You will need to add KineticLaws for dimerisation ( $k_D * s_3 * s_3$ ), repression by dimers ( $k_1 * p_0^n / (p_0^n + s_9^n)$ ), and dimer degradation ( $d_3 * s_9$ ). Your network should look something like that in Figure 6.

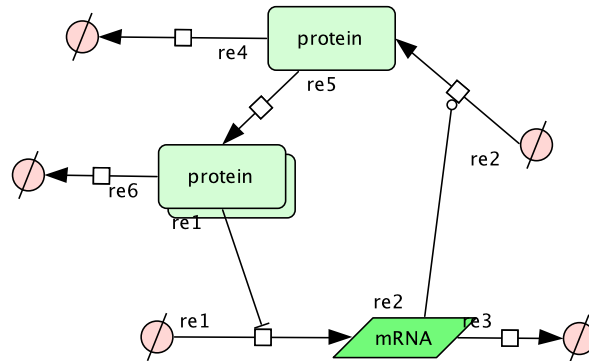


Figure 6: Negative feedback loop with mRNA transcription repressed by protein dimers.

For the parameters, set  $k_D = 1$ ,  $d_3 = 0.05$ , and keep the other parameters the same as listed in Section 3.1 (so  $k_1 = 50$ ,  $p_0 = 10$ ,  $n = 10$ ,  $d_1 = 0.1$ ,  $k_2 = 1$ ,  $d_2 = 0.1$ ). Use an end time of 100, and set the initial concentrations to:  $m = 1$ ,  $p_1 = 5$ ,  $p_2 = 5$ . What do you find with this relatively rapid dimerisation rate?

- Decrease the dimerisation rate to  $k_D = 0.1, 0.01, 0.001$ . What happens to the magnitude of the overshoot of the steady state? For  $k_D = 0.001$  try increasing the time-span of simulation to 1000. What do you notice? What happens as you decrease the dimerisation rate further? What happens to the frequency of any oscillations? Is this what you expected?
- With  $k_D = 0.001$ , reduce the cooperativity  $n$  of the Hill function from 10 to 2 in steps of 1. Describe what happens to the time course of mRNA and protein. Is this what you expected? Can you make biological sense of this?
- The form of transcription rate used assumes that the transcription factor can reduce the rate of transcription to zero (for high concentrations). In reality, some residual transcription is still possible even when the concentration of repressor is high. This can be incorporated in the model by adding a basal transcription rate  $k_0$ . Try adding this to your model in CellDesigner. You should modify the Kinetic Law for production of  $m$  to something like “ $k_1 * p_0^n / (p_0^n + s_5^n) + k_0$ ”, where “ $s_5$ ” should correspond to the species  $p_2$ , and add the new parameter  $k_0$ . With  $k_D = 0.001$  and  $n = 10$  as before, find out what happens to the oscillations as you increase  $k_0$  from zero.

## 4 Repressilator

**Challenge:** Implement a repressilator-like model, by using the toggle-switch example to build a network of three repressors  $u$ ,  $v$  and  $w$  (see Figure 7). Use the same forms of repression as before, with maximum production rates of 8 (for  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$ ), and powers of 4 (for  $\beta$ ,  $\gamma$ , and a new power,  $\phi$ ).

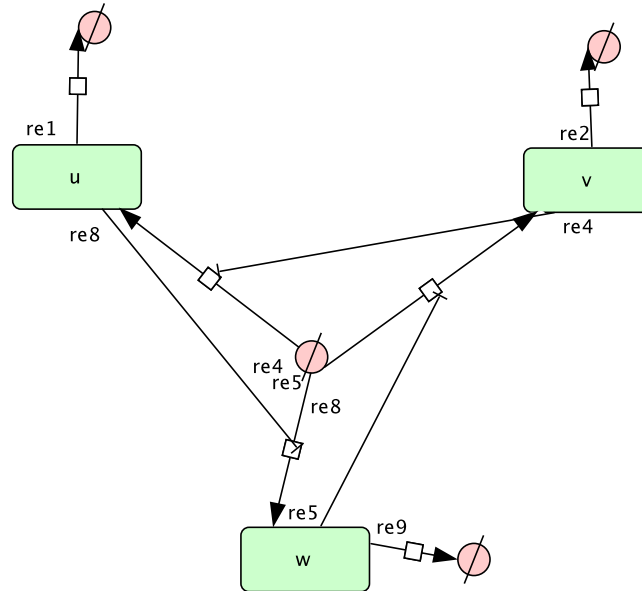


Figure 7: Repressilator network.  $u$  represses  $w$ -production,  $v$  represses  $u$ -production, and  $w$  represses  $v$ -production, and all three species degrade.

If you don't want to create this yourself, the model is available as `repressilator_mro.xml` on the course web-site. Load the model, click "Simulation" -> "ControlPanel", change the end time to 30 and Num Pts to 300, then click "Execute". You should see the three components oscillate.

### 4.1 Full repressilator model

`rsif-2009-0035-File007.xml` specifies the complete repressilator model, including genes, mRNAs and Proteins, and transcription and translation steps. Try loading this and simulating, with an end time of 500—you should see oscillatory solutions for the default parameters.