

CPIB SUMMER SCHOOL 2011: INTRODUCTION TO BIOLOGICAL MODELLING

Analysis of single-variable models / CellDesigner

Markus Owen

1 Getting started in UoN computer labs

Find a free computer. If it needs to be switched on, you should choose "start Windows XP" at the initial screen. Except for those with a University of Nottingham (UoN) account, all course participants will use the same shared account:

Username: "pmasb"

Password: "SystemsBiology"

This account includes 1Gb of "personal" filestore. To find it, go to "My Computer", where the filestore is mounted as drive "Z:". This is independent of which computer you use to login, so you can save your work and return to it later during the course. However, because the account is shared, you should save your work in a separate directory, e.g. using your name. Alternatively, if you have access to a memory stick or other USB storage device, you could use that to save your work, and to take away all your computer work from the course after it has finished.

You should work through the construction of the model in Section 3 before attempting the exercises in Section 4. Files for this practical can also be found at:

<http://www.maths.nottingham.ac.uk/mathsforslife/cpib2011/>

If you download any files, be careful to save them either to your named directory on the Z-drive, or to your own storage device. **Please ensure that you save all your files from this course in your own separate directory, so that participants on the course do not accidentally overwrite each others work.**

2 Launching CellDesigner

CellDesigner (<http://celldesigner.org/>, <http://www.systems-biology.org/>) is a free diagram editor for drawing gene-regulatory and biochemical networks. Networks are stored using the Systems Biology Markup Language (SBML), a standard for representing models of biochemical and gene-regulatory networks. Networks are able to link with simulation and other analysis packages through the Systems Biology Workbench (SBW).

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

3 Single positive feedback loop

In this section you will construct a simple model for a single gene feedback loop, as was discussed in lecture 1.2. You will implement a model which corresponds to the ODE:

$$\frac{dx}{dt} = P(x) - \delta x, \quad \text{with} \quad P(x) = \frac{Ax}{h+x}. \quad (1)$$

The structure of the network is shown in Figure 1.

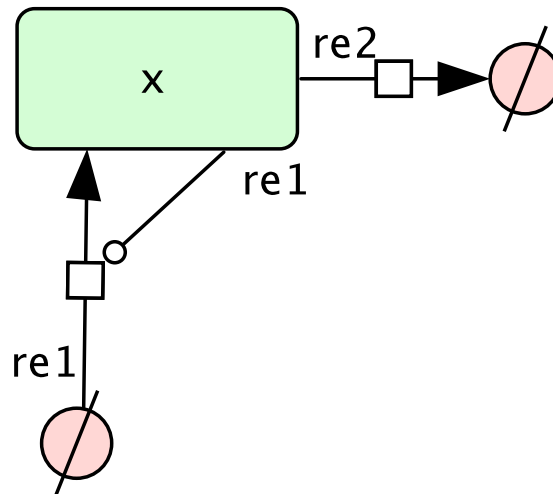


Figure 1: Schematic of the positive feedback loop in the model given by equation (1). x activates its own production (reaction “re1”) and degrades (reaction “re2”).

3.1 Creating the model in CellDesigner

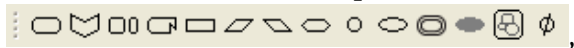
Follow the instructions below to create the model in CellDesigner. If you have difficulty, you can download `PositiveFeedback.xml` from the course web-page.

1. Start a new model.

Click ‘File’, ‘New’ and ‘OK’. Then ‘File’, ‘Save’ and save it (either to your named folder on the Z-drive or to your own storage device) as e.g. `PositiveFeedback`. Make sure you save the file periodically during the steps below (using “File->Save”, or “Ctrl-S”).

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 (you can always move objects later). A dialog should pop up, asking for a name for the species (e.g. ‘source’).

3. Create the ‘protein’ x .

Click protein (the first entry on the Species toolbar), click where you want it on the diagram, and then name the protein x .

4. Create transitions from the source to x , representing production of x .

Click ‘state transition’, the first arrow on the Reaction toolbar:



then click on the canvas at the starting point, ‘source’, and target, x . You can name these reactions by clicking the ‘Reactions’ tab and the double-clicking the ‘name’ column to enter text (e.g. ‘ x -production’).


5. Make these production terms activated (or catalysed).

Click ‘Catalysis’, first icon on the Modifier section of the Reaction toolbar:



then click the x icon on the canvas, followed by the x production reaction (to implement x activating production of itself).

6. Now implement degradation of x.

Click 'Degradation' in the toolbar () , then click the x icon on the canvas.

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 rate. Each of the reactions needs *rate laws* specifying.

7. Right click on reaction 1 (this should be production of x, activated by x), click 'Edit KineticLaw'. This opens a window showing the reaction of interest, where you must specify the reaction rate and relevant parameters.

In the 'math' textbox, set the Kinetic Law: $A*s_2/(h+s_2)$. Note that s_2 here refers to species 2, which is x. When entering expression for rate laws, you can click on the relevant species to get its id.

Click 'update', 'close', and 'close' in the reaction window.

The parameters also need adding. Click the 'Parameters' tab. Click New button. In the Parameter dialog, specify id, name, value and units (use "substance") for the new parameter. Do this for A and h (use $A=1$, $h=1$).

For degradation, simply right click on reaction 2 and click 'Edit KineticLaw'. In the 'math' textbox, set the Kinetic Law: $\delta*s_2$. Again, s_2 refers to species 2, which is x. Add the parameter, with $\delta=0.5$.

Save the file, 'File->Save', or Ctrl-S.

3.2 Simulations using the "ControlPanel"

Now let's try a simulation. First we set the initial value of x. Click the "Species" tab, then double click the "initialQuantity" entry for x, and set it to 0.5.

Click "Simulation", then "ControlPanel" in the main menu bar. Change "End Time" to 40, then click "Execute". The simulation will run and you should get a plot of the result with three curves. If you click on a species in the panel to the right of the plot, the relevant curve flashes.

We need to make a couple of changes now. First, we don't really need to see the source and degraded species; we need to set the "boundaryCondition" for these 'empty' states to "true" (this means that the species are not determined by a chemical reaction). Do this by double clicking the "b.c." entries in the "Species" tab, or right clicking on the species and selecting "Edit Species", and then set "boundaryCondition" to "true". Save the file, "File->Save", or "Ctrl-S".

Now click 'Execute' again. You should now see just one curve, with x starting at 0.5 and rising to reach 1.0 (see Figure 2).

From Lecture 1.2, we expect that different initial conditions will give the same outcomes (because there is only ever one stable steady state). We can check this by changing the initial value of x. In the ControlPanel "Species" tab, click the x "InitialValue", and change it to 1.5. Then click "Execute" — this time you should see x go down to approach one from above.

You can even see the result for a range of initial values for x: in the ControlPanel is a "Parameter Scan" tab. Click the "Scan parameter" checkbox, make sure "Initial value" is selected, change the "Name" to x, change "To" to 2, then click "Execute" and click the "Results" tab. Click on "Overlay Setting" and select x only, then "OK", then "Show Graph". You should see a set of curves, one for each initial value of x. To see which curve is which, look at the color scheme and associated initial x values at the right of the window. Solutions from low initial values approach the steady state at 1.0 from below, and high initial values approach from above (see Figure 3).

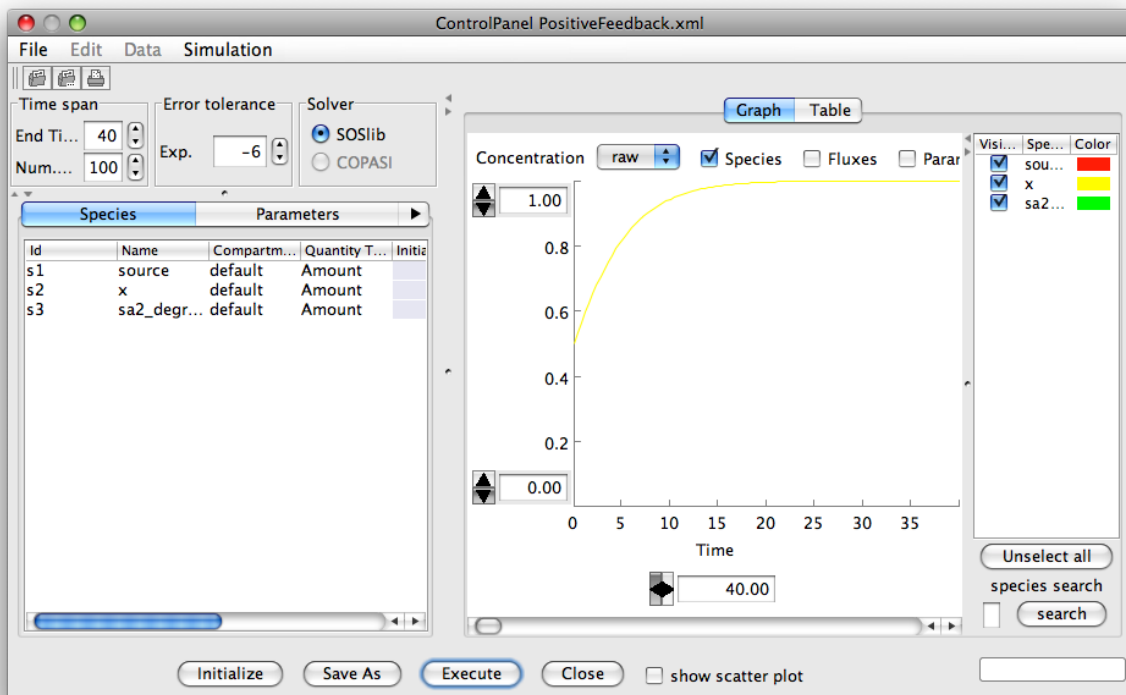


Figure 2: Simulation, using the CellDesigner “ControlPanel”, of the positive feedback model given by equation (1), with $x(t = 0) = 0.5$. x goes to the stable steady state at $x = 1$.

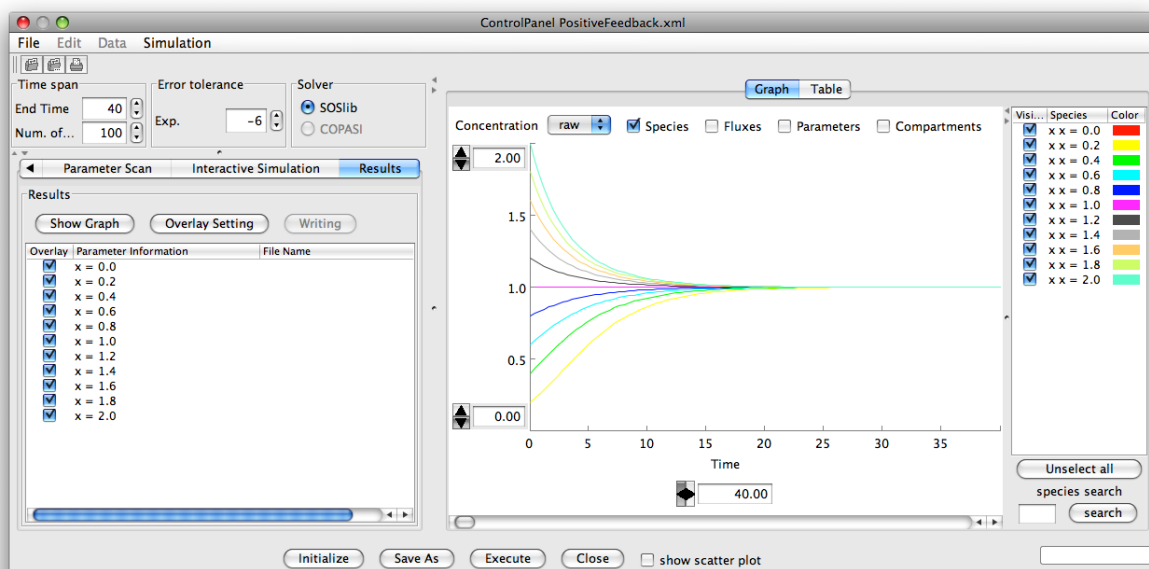


Figure 3: Simulation of the positive feedback model given by equation (1), using the “Parameter Scan” tab to vary the initial condition $x(t = 0)$ from 0 to 2. x goes to the steady at 1.0 from below or above, depending on the initial value.

Alternatively, we could consider what happens as a parameter varies. First go to the “Species” tab and set the x “InitialValue” to 0.3. Now go to the “Parameter Scan” tab, select “Parameter value”, change the “Name” to “delta”, change “From” to 0.5, “To” to 1.5 and “Interval” to 0.1, then click “Execute” and click the “Results” tab. Click on “Show Graph”. You should see a set of curves, one for each value of “delta” (see Figure 4). To see which curve is which, look at the colors and associated parameter values at the right of the window.

Low values of “delta” give a higher steady state x-value; as “delta” increases, the steady state decreases. However, for $\delta > 1$, x goes to zero (see Figure 4) — this is an example of a bifurcation, as illustrated in slide 23 of lecture 1.2. If you are not convinced that x goes to zero when $\delta > 1$, you can increase the “End Time” and decrease the range on the vertical axis (so you can focus on small values of x).

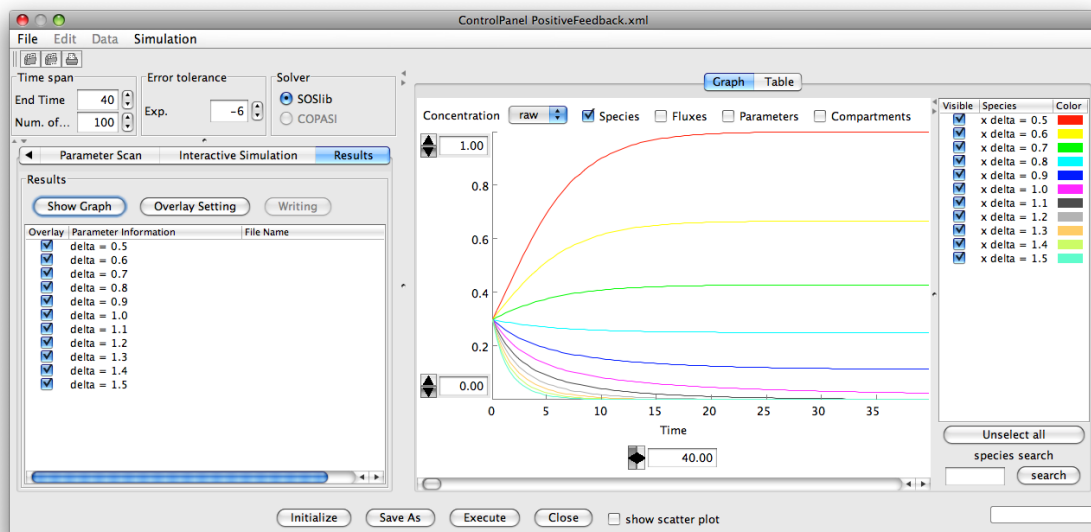


Figure 4: Simulation, in the CellDesigner “ControlPanel”, of the positive feedback model given by equation (1), using the “Parameter Scan” tab to vary the parameter delta from 0.5 to 1.5 (with $x(0) = 0.3$). The steady state decreases with delta until a change in behaviour, so that, for $\delta > 1$, x goes to zero.

You can save plots using “File” > “Save Image / Print” on the “ControlPanel” window.

4 Exercises

The above should be enough to get you started with building models and simulating them in CellDesigner. The following exercises will build on this so you can use CellDesigner and graphical techniques to analyse models of the kind discussed in the lectures.

We will study the case of

$$\frac{dx}{dt} = P(x) - \delta x$$

with various forms for $P(x)$. This simple model represents transcriptional autoregulation, where $P(x)$ characterises the effect of an mRNA on its own rate of transcription, and we assume a constant rate of mRNA degradation.

4.1 Positive feedback: $P(x) = \frac{Ax}{h+x}$

1. Sketch $P(x)$ and δx on the same axes.

Recall that steady states will be where these two cross (where $\frac{dx}{dt} = 0$).

How many ways can $P(x)$ and δx cross? How does this depend on the parameter values?

Do your sketches agree with the graphical analysis in the lectures? Identify any bifurcations.

2. Use your answer to sketch the possible forms for $\frac{dx}{dt} = P(x) - \delta x$.

What is special about the points where the curve crosses the horizontal axis?

On your sketches, mark any steady states and add arrows indicating where x is increasing or decreasing. Convince yourself that increasing δ should eliminate the non-zero steady state (and make $x = 0$ stable).

3. **Challenge:** based on your phase-line diagrams, sketch representative solutions (i.e. sketch x as a function of t for different initial conditions $x(0)$).
4. You have already simulated this system in CellDesigner for a range of initial conditions, for $A = 1$, $h = 1$ and $\delta = 0.5$. You should have seen a stable steady state at $x = 1$. You then simulated for a range of δ values. Based on your sketches, explain what happens as δ increases.

4.2 Positive feedback: $P(x) = \frac{Ax^n}{h^n + x^n}$

When $n = 1$ the model is the same as in section 4.1. However, with $n > 1$ this Hill function form has a different shape which raises additional possibilities including **bistability**. Here, and in the following, it is best to save your modified models under different names. Start by closing the ControlPanel, and saving "PositiveFeedback.xml" as a new file "hill.xml".

Then modify the model to allow a Hill function with arbitrary n . This should simply require modifying the kinetic law for reaction 1 ("x-production") to $A*s^{2n}/(h^n+s^{2n})$, and adding the new parameter n . Initially set $n = 1$ and run a couple of simulations from the previous exercise to check that it gives the same results when $n = 1$.

1. Sketch $P(x)$ for $n = 2$. Then sketch δx on the same axes.

How many ways can these two cross? How does this depend on the parameter values?

Do your sketches agree with the graphical analysis in the lectures? Identify any bifurcations.

2. Use your answer to sketch the possible forms for $P(x) - \delta x$.

Remember that this is the graph used for the phase-line diagram.

Mark steady states and add arrows indicating where x is increasing or decreasing.

3. In CellDesigner, set $n = 2$, $A = 1$, $h = 0.5$ and $\delta = 0.9$. In the "ControlPanel", run a parameter scan for initial values of x from zero to 1.5 (similar to that in Figure 3).
Do all initial conditions go to the same steady state (as they did in the previous example)?
Do the simulated solutions agree with your qualitative analysis?
4. What happens as you increase δ ? Identify any bifurcations.
If you increase δ just beyond the bifurcation, you should find that solutions starting from large x take a long time to reach the remaining steady state (you may have to increase the "End Time" - e.g. with $\delta = 1.001$ try "End Time=200").
5. What happens as you decrease δ ?
6. What happens to the shape of the function $P(x)$ if you increase n ?
Does this change the qualitative nature of the system?

4.3 Positive feedback with constitutive transcription: $P(x) = \frac{Ax^n}{h^n + x^n} + B$

When $B = 0$ this is of course the same as the previous example 4.2. We know that with δ sufficiently large we can lose the upper steady states. Here we see that increasing B can regain those states, but also as we increase B further we lose the *lower* two steady states.

Start by saving “hill.xml” as a new file “hillbasal.xml”. Then modify the model to add the basal transcription term to $P(x)$. This should simply require modifying the Kinetic Law for reaction 1 (“x-production”) to $A*s^{2^{\wedge}n}/(h^{\wedge}n+s^{2^{\wedge}n}) + B$, and adding the new parameter B .

1. You have already sketched $P(x)$ when $B = 0$. What happens to $P(x)$ as B increases? How does this change the ways that $P(x)$ can cross δx ?
2. Use your answer to sketch the possible phase-line diagrams for $\frac{dx}{dt} = P(x) - \delta x$. You should find that there are three qualitatively different pictures. Mark steady states and add arrows indicating where x is increasing or decreasing.
3. Now set $n = 4$, $A = 1$, $h = 0.5$ and $\delta = 1.2$. Run a “Parameter scan” for a range of initial x values, and repeat for $B = 0, 0.1, 0.2, 0.3$. How does the dynamics change as B varies? HINT: you should find *bifurcations* at $B \approx 0.04$ and $B \approx 0.245$.
4. As you increase B just beyond the second bifurcation at $B \approx 0.245$, you should find that all solutions converge to a steady state with a high TF level, but that solutions starting from small x take a long time to reach that remaining steady state (you may have to increase the maximum time in $tspan$).
5. What are the biological implications of sigmoidal feedback plus basal transcription?

This completes the exercises.

5 Other models

There are many other models available in SBML format that can be loaded into CellDesigner (and other packages). A selection are available from the course web-page:

<http://www.maths.nottingham.ac.uk/mathsforslife/cpib2011/>

The BioModels Database includes over 600 models available for download:

<http://www.ebi.ac.uk/biomodels-main/>

5.1 Stem cell, trophoctoderm and endoderm lineage determination

BIOMD0000000209_Chickarmane.xml contains a model for stem cell regulation. Try simulating the model, changing the value of A in the Species / Initial Quantity from 1, to 10 and 25. Each value should give a different end result, corresponding to Trophoctoderm (High CDX2), Stem Cell (Sox2, Oct2 high) and Endoderm (High GATA6).

5.2 Modelling the cell division cycle

The proteins cdc2 and cyclin form a heterodimer (maturation promoting factor) that controls the major events of the cell cycle. A mathematical model for the interactions of cdc2 and cyclin shows

that the control system can operate in three modes: as a steady state with high maturation promoting factor activity (as in metaphase arrest in unfertilized eggs), as a spontaneous oscillator (as for rapid division cycles in early embryos), or as a switch (as in growth-controlled division cycles).

The biomodel is `BIOMD0000000005.xml`. Load it, open the Simulation ControlPanel, and click Execute. You should see oscillations in the amounts of each component.

For further information, see

<http://www.ebi.ac.uk/biomodels-main/static-pages.do?page=ModelMonth%2F2006-10>

To see the transition into and out of spontaneous oscillations, do a Parameter scan: set the Parameter value `Reaction1:k6`, from 0.1 to 2 with interval 0.1. For the Overlay setting, choose `cdc2k` only. Up to `Reaction1:k6=0.3`, `cdc2k` is at high steady state levels, then it oscillates, until about `Reaction1:k6=1.5` oscillations are lost again.

5.3 Human weight cycling

A simple model for the dynamics of human weight cycling (A Goldbeter: A model for the dynamics of human weight cycling. *J Biosci* 2006 Mar;31(1):129-36):

<http://www.ebi.ac.uk/biomodels-main/static-pages.do?page=ModelMonth%2F2007-08>

The biomodel is `BIOMD0000000079.xml`. The model variables are: weight, P; dietary input, Q; and cognitive restraint, R. The idea is that excess weight increases cognitive restraint (i.e. you choose to eat less), limiting dietary input to allow weight control. `reaction_5:V` measures the rate at which cognitive restraint decays (i.e. how fast you forget that you should eat less).

For low `reaction_5:V`, a steady weight is reached. For intermediate `reaction_5:V` (less than about 9.4), weight oscillates, for `reaction_5:V` greater than 9.4, a steady state is again reached.

To see these features together, you could try a parameter scan. Click the 'Parameter Scan' tab, click 'Scan Parameter', select the 'Parameter value' radio button, and for Name choose `reaction_5:V`. Set From=0.1, To=15, execute and plot (use Overlay Setting in the Results panel to just show P, the weight).

5.4 Oscillations in the mitogen-activated protein kinase cascades

`MAPK.xml`: open the file, and the simulation Control Panel, change the end time to 10000 and the Num. of Points to 1000, and click 'Simulate'. You should see oscillations.

`MAPK41.xml` contains a heavily annotated version of the same model, using MIRIAM (the Minimal Information Requested In the Annotation of Models, <http://www.ebi.ac.uk/miriam/>), a standard to annotate and curate computational models in biology.

6 Conclusions

This session should have given you an idea as to the sorts of things that can be done with software such as CellDesigner, and given you practice in using sketches and phase-line diagrams to analyse simple models. There are numerous other tools that build upon SMBL, there are alternative model markup languages (particularly CellML), and increasingly tools for organising models, experimental data and relevant literature in a linked manner.