

Simulation of Cellular Systems

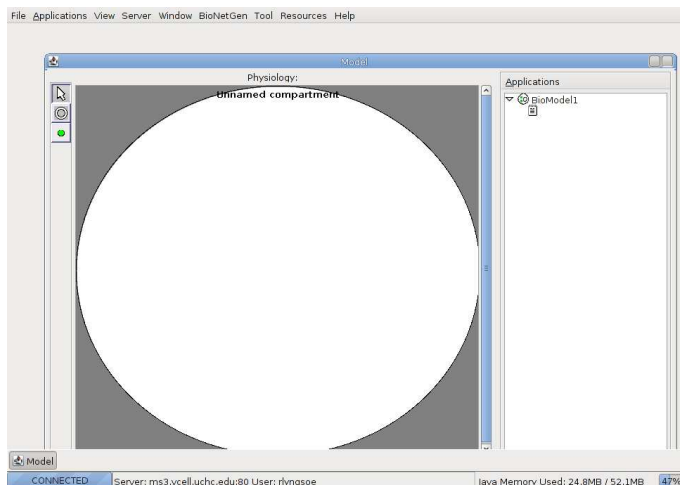
Practical – Topics in Computational Biology

14th of June 2010

When only a few key reaction pathways, small regulatory systems, and simple signalling where under active scrutiny, manual investigation by analytic methods and special purpose software implementing specific models suffices. However, with the increasing interest in and knowledge of systems governing the dynamics of cellular mechanisms it has become increasingly useful to develop general purpose systems for modelling and simulating such systems. There are several such systems, e.g. VCell available from <http://www.nrcam.uchc.edu/> and the E-Cell project available from <http://www.e-cell.org/ecell/> – many more references are available at <http://systems-biology.org/software/>.

In this practical we will make a brief foray into the capabilities of V-Cell (The Virtual Cell) for building mathematical models of regulatory systems in a cell. VCell is a distributed application used over the internet. A disadvantage of this is that you need to have internet connection to use it, but the great advantages are that the installation is relatively simple as V-Cell just needs a Java client to run locally, and that models that you construct are stored centrally and can even be made publicly available to all V-Cell users. It comes with a simple – which can at times also mean slightly cumbersome – graphical interface for designing a cell model, or indeed a model of almost any biological system.

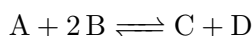
The first thing you need to do is get the system started. Go to <http://www.nrcam.uchc.edu/>, select Run VCell Software, and choose the latest stable version. This should start installation of the local client software, and once this has finished you should have VCell running. The first thing you need to do is register, which should happen instantaneously. After that you should see a window similar to



This is the blank canvas on which you can start building biomodels.

Single Chemical Reaction

Rather than going straight for a model of a complex biological system, it may be a good idea to start out small and learn how to use the various features of the software. So to begin with we will just try to model the simple reaction



assuming simple mass action reaction kinetics. The latter means that the kinetics of the reaction can be described by the equation

$$v = k_+ [A] [B]^2 - k_- [C] [D]$$

where v is the 'speed' of the reaction, e.g. the rate with which the concentration of A is decreasing.

The initial frame presented by VCell essentially allows us to build an abstract model of our system, specifying the features and species of our model. Features correspond to the various structures a physical area is segmented into, e.g. nucleus, mitochondria etc. Species are the objects existing inside a feature, e.g. small molecules and ions, catalysts, etc. As already stated, in the biomodel window these things are only specified in an abstract way, so exact location is unimportant (though nesting of features and species within other features is). One can define a physical geometry and link this to the abstract biomodel, but this will not be part of the set part of this practical. If no physical geometry is specified, VCell will assume that each feature can be treated as a homogeneous solution of the species within it.

In our small example we only have one feature, namely the solution in which the reaction takes place, and four species corresponding to the two reactants and the two products. Once you have used the Species Tool to add these four species to the system, a right click should bring up a menu allowing further manipulation of the feature. Most notably, you will be offered a choice to add

reactions. This will bring up a very similar interface, but with a slight variation of the tools available on the left hand side. Apart from a Reaction Tool, there is also a Flux Tool for connecting adjoining features via a cross-membrane channel. With just one feature this is not relevant for us, so simply use the Reaction Tool to add a reaction. Once this is done you need to connect each species involved to the reaction. This is done using the RX Connection Tool, starting from the species and ending at the reaction. Depending on where you put the end point in the reaction the species will take the part of either a reactant, a product or a catalyst.

Once you have connected the all species, right click the reaction to choose properties. The most important property to update is the reaction rate, that defaults to 0.0. Replace this with the expression above (notice that the species names are suffixed with their location – VCell offers completion possibilities, so this shouldn't be a cause for much concern). Once you have entered the correct rate expression, you will notice that this adds fields for k_+ and k_- to the table of relevant parameters for this reaction. These need to be given some value (though not necessarily fixed, they can depend on time, physical coordinates, etc.) to be able to simulate the system. You can also set them as global parameters, so that if they are also used in other reactions they need not be entered separately for each. In fact, in the biomodel window another right click option are for adding global parameters without first having to use them in a reaction rate expression and for viewing all global parameters and rate expressions.

Once you have defined the reaction and set values for k_+ and k_- , you should now be ready to convert the model to an application. This is done in the Applications menu. Set up a deterministic application. You will need to change initial conditions, as these defaults all concentrations to zero, in which case v will remain zero and nothing will change. Once that is done, you should be ready to set up a simulation of the system.

Fission Yeast Cell Cycle

Unless you are actively involved in a biomodelling project, it can at times be difficult to locate the parameters specifying a dynamic system. One exception is the cell cycle in fission yeast, where [1] contains a full set of equations and parameters for the yeast cell cycle. This paper is available from <http://jigcell.biol.vt.edu/Pubs/Novak01.pdf>. Try to see whether you can build a model for this system in which the cyclic nature can be observed. It may be beneficial to forget about the connectivity shown in Fig. 2 and simply focus on Tab. 1 instead.

One problem with the equations in Tab. 1 is that they do not include cell division. The M entity captures the cell mass, or size, and when mitosis is triggered by the cell cycle, this should be halved. In my experience, it is possible to get the cyclic behaviour up and running for a limited number of cycles, but eventually the system breaks down. One could probably solve the problem by adding some complex technical pseudo-species and reactions to the system that would respond to the appropriate signals and roughly halve M . A simpler way

to get a more stable cyclic behaviour going would be to keep M of manageable magnitude. Try to enclose the system in a feature, and add a channel for exporting excess M out once it grows beyond a certain level. Once you have set up the full model, it is quite easy to explore the importance of various reaction for the stability of a system, as you can untick whether they are enabled or not. Are there any reactions in the yeast cell cycle that can be disabled without the cyclic behaviour disappearing?

Other Models

As mentioned in the introduction, one huge advantage of the distributed nature of VCell is that it is quite easy to share models. You may already have noticed this if attempting to load or save a model or application, as there is a folder with shared models. However, it may be difficult to separate the wheat from the chaff in this collection, so to see some well worked examples of models of dynamic systems it may be a better choice to locate the Biomodels.net in the Resources menu. This will allow you to choose between a handful of curated models to load. Try e.g. to load the circadian clock model. The circadian clock systems are so called as they have a cycle of approximately one day. What would be a natural end point to choose in the simulation to see the cyclic nature? If we forget about nature and try to design things, could you set up a system with two cells synchronising their clocks by passing signals through their membranes?

References

- [1] Béla Novák, Zsuzsa Pataki, Andrea Ciliberto, and John J. Tyson. Mathematical model of the cell division cycle of fission yeast. *Chaos*, 11(1):277–286, 2001.