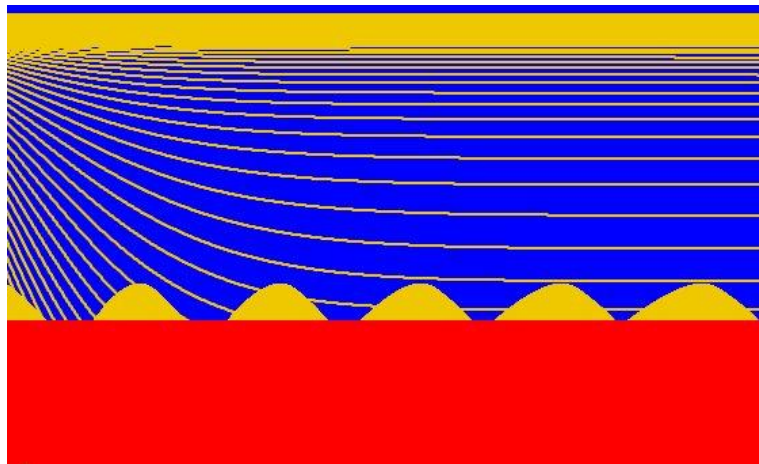




Building and Simulating Models using COPASI

24 November 2016





Licence

This tutorial is © 2016, Nicolas Le Novère, Viji Chelliah, Bhupinder Virk.

This manual is distributed under the creative commons Attribution-Share Alike 4.0 licence. This means that you are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

Under the following conditions:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Please note that:

- For any reuse or distribution, you must make clear to others the licence terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

Full details of this licence can be found at:

<https://creativecommons.org/licenses/by-sa/4.0/legalcode>

Introduction

This tutorial is not meant to be an advanced course. However, it aims at being practical and after completing it you should be able to create your own simple models and run a few simulations and analyses. There are many modelling approaches, and many model analysis procedures. We will only concentrate on one type of model: Chemical kinetics in well mixed solutions. Such models have been around since the mid-XXth and are still the most used type in systems biology.

We will use a mix of models that were actually published as part of research projects, and toy models specially created for the sake of this tutorial. Their size and complexity have been chosen to fit the duration of the tutorial.

Many concepts will be introduced and not developed. For more general background on the modelling of biochemical networks, see:

- Le Novère. Quantitative and logic modelling of gene and molecular networks. *Nature Reviews Genetics* (2015) 16: 146–158

For a complete introduction to systems biology, including modelling, see:

- Edda Klipp *et al.* Systems Biology: A Textbook. *Wiley* (2016)

A good book focussed on modelling, with tutorials, is:

- Fall *et al.* Computational Cell Biology. *Springer* (2002)

Finally, the following book presents different approaches used to model dynamic phenomena in life sciences, expanding further than the previous ones. It comes with plenty of exercises in MatLab and R.

- Ellner and Guckenheimer. Dynamic Models in Biology. *Princeton University Press* (2006)



Modelling software

This tutorial will use the modelling and simulation tool COPASI (standing for “**C**omplex **P**athway **S**imulator”). COPASI is an open-source software widely used in computational systems biology because of its versatility and ease of use. It comes with a command-line version that can be used for instance on clusters, and a user friendly graphical interface - which we will use today. The tool is used for teaching but also for research (the main paper, Hoops et al 2006 has been cited > 1400 times). Plenty of information including manuals, videos and user forums is available at:

<http://www.copasi.org>

Models

Some of the example models used for this tutorial are taken from the BioModels database (<https://www.ebi.ac.uk/biomodels>). BioModels contains thousands of mathematical models covering a large diversity of biological processes. Many of them have been carefully verified and annotated, and are a great starting point for further modelling endeavours.

In particular, we will use:

- BIOMD0000000009 – Huang 1996
Huang CY, Ferrell JE Jr. Ultrasensitivity in the mitogen-activated protein kinase cascade. Proc Natl Acad Sci U S A. 1996 Sep 17;93(19):10078-83.
(<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000009>)
- BIOMD0000000010 - Kholodenko2000
Kholodenko BN. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. Eur J Biochem. 2000 Mar;267(6):1583-8.
(<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000010>)

The Mitogen Activated Protein Kinase cascade is a prototypic example of a behaviour emerging from a system. As we will see, the response of a MAP kinase to an upstream signal is shaped by the whole cascade of successive phosphorylation. The prediction that such a cascade would engender ultrasensitivity is one of the great successes of mathematical modelling in biochemistry, made in 1981 by Albert Goldbeter and Daniel Koshland, a decade before the discovery of MAP kinase cascades. More can be found at:

<https://nlenov.wordpress.com/2013/07/02/modelling-success-stories-3-goldbeter-and-koshland-1981/>

1) MAPK cascade - Ultrasensitivity:

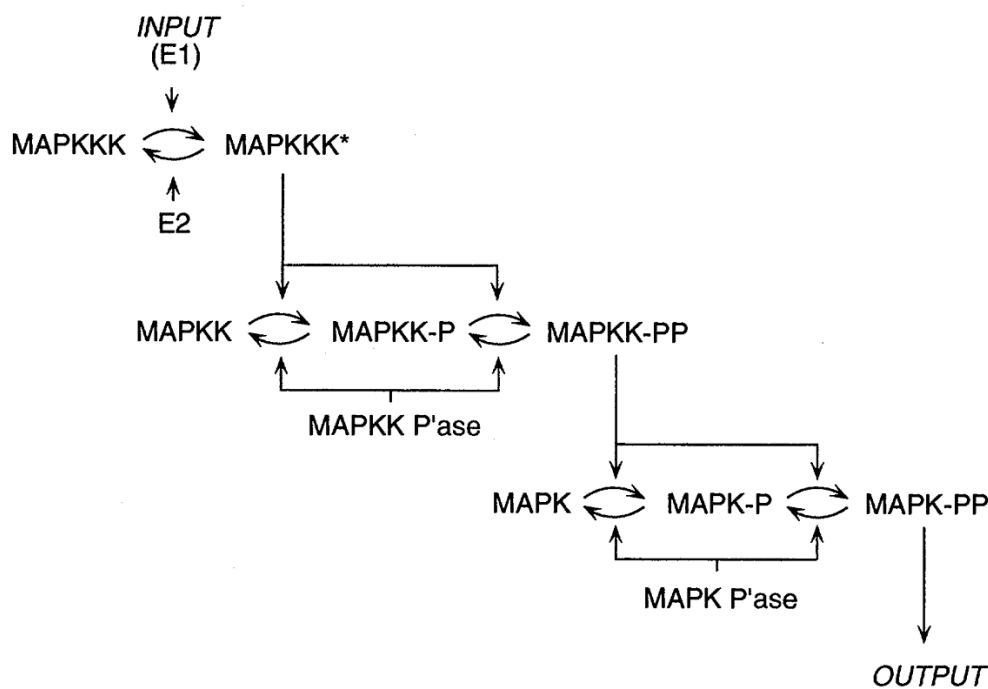
Model:

BIOMD000000009 – Huang 1996

Huang CY, Ferrell JE Jr. Ultrasensitivity in the mitogen-activated protein kinase cascade. Proc Natl Acad Sci U S A. 1996 Sep 17;93(19):10078-83.

<http://www.ebi.ac.uk/biomodels-main/BIOMD000000009>

This was the first detailed model of MAPK signalling showing the effect of cascading phosphorylation steps on MAPK signalling (although as mentioned above, the theory was there already). All intermediate complexes were modelled explicitly, the reactions using Mass Action kinetics (compare this approach with the one used in the next exercise). Parameters and concentrations were estimated from experimental results. The rate equations of the cascade were solved numerically.



Schematic view of MAPK cascade (Huang and Ferrell. 1996)

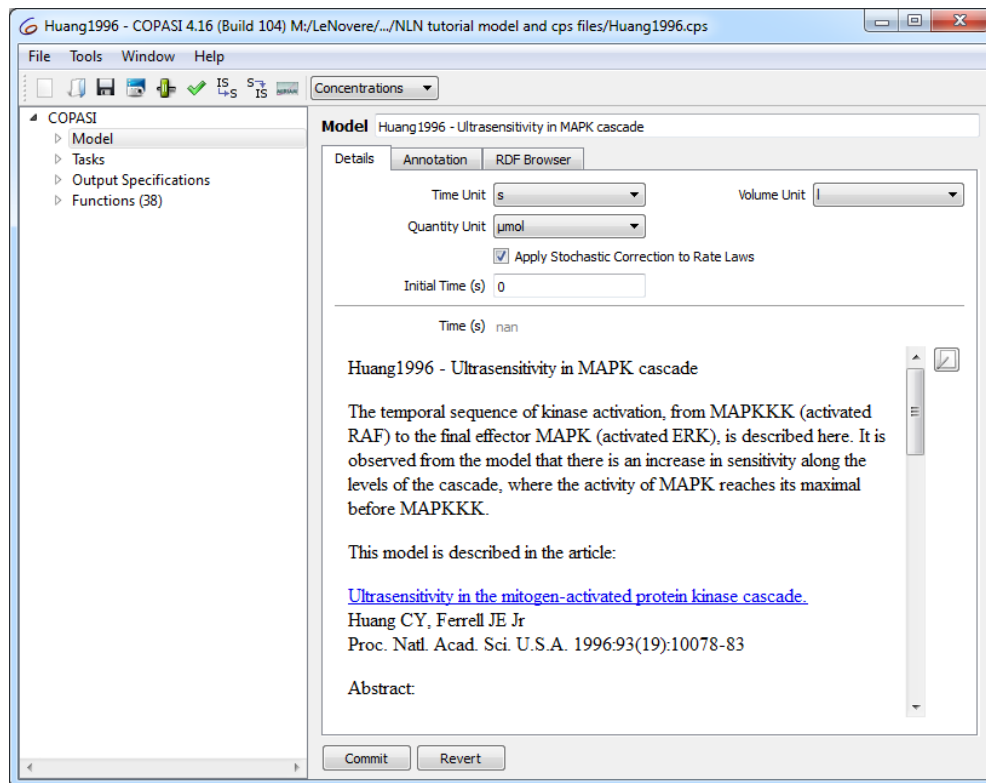
Predicted results were similar to experimental results obtained with *Xenopus laevis* oocytes. In particular, the dose response curves for MAPK, MAPKK, MAPKKK were predicted to show increasing ultrasensitivity (or supra-linearity), with the MAPK curve predicted to be the steepest.

In this exercise, we will import a model already written, using the format SBML (Systems Biology Markup Language, <http://sbml.org>).

1.a Running a time course simulation:

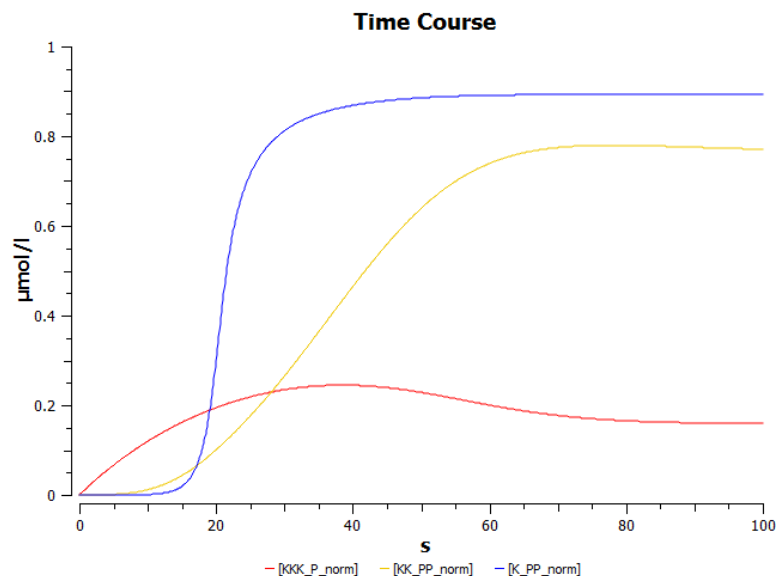
A time course simulation shows the evolution of a system over time. It computes the values of the variables (here the concentrations) after a small amount of time based on their values now. Time courses are particularly important to understand signalling events and their consequences.

- Import **Huang1996.xml** into COPASI.



- Explore the model and try to understand what each reaction does. Use the map on the previous page for context. Look at the “Annotation” tab of “Model”, a selected “Species” and a “Reaction”.
- Prepare a Time Course (Under Tasks) with duration = 100 seconds, Interval Size = 0.1 seconds, Method = Deterministic (LSODA).
- Use the Output Assistant to create a “Concentrations, Volumes and Global Quantity Values” plot, and then run the time course.
- Hide all of the curves except for the normalized active forms of MAPK (“K_PP_norm”), MAPKK (“KK_PP_norm”) and MAPKKK (“KKK_P_norm”).
- Save your model (e.g myHuang1996-TimeCourse.cps)

The plot below shows that MAPKKK (e.g. Raf) is activated first, then MAPKK (e.g. MEK) and finally MAPK (e.g. ERK). However, MAPK reaches maximal activation in 30 seconds, well before MAPKK (40 secs) and MAPKKK (70 secs). Why does MAPKKK shows a maximum at 40 secs, then decreases to reach a lower plateau?

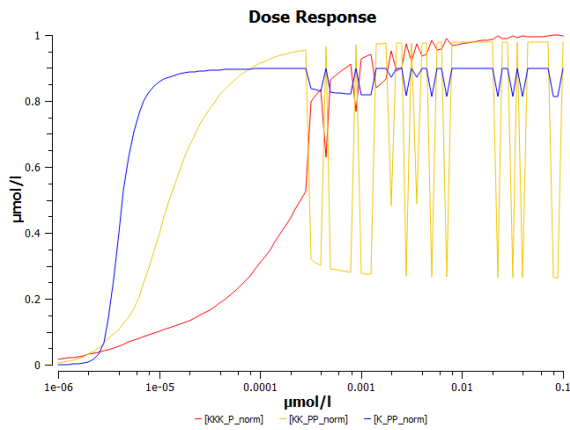


1.b Creating a dose-response curve – sensitivity to signal:

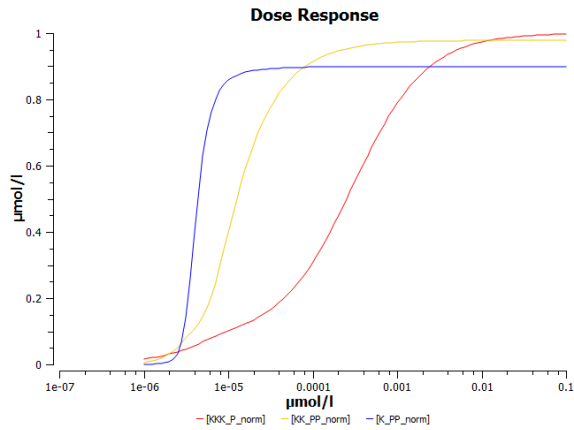
Dose response studies help us understand any change caused by varying a dosage on the dynamics of the system. These are used for instance to determine the efficient, “safe” and “hazardous” dosages of drugs, and also to optimize any perturbation protocol. The numerical procedure is called parameter scan, or parameter sweep. As the name indicates, they can also be used to study the effect of varying any parameter and therefore study the sensitivity of the system towards this parameter (there are more efficient way to do this, which are beyond the scope of this course).

Here, we will study the stimulus/response curves for three components of the MAPK cascade, the input stimulus being the MAPKKK_activator (INPUT(E1) on the map above, for instance Ras). We will compute the steady-state of all concentrations (when their values do not vary anymore) for increasing values of the MAPKKK_activator.

- Select Parameter Scan (Under Tasks), and change the Task type to “Steady State”.
- Set new scan item to Scan and click create.
- Select the parameter to be scanned as Initial Concentration of the [MAPKKK activator](t=0)
- In the scan item you have just created set: Intervals = 100, min = 1e-6, max=0.1, and select logarithmic
- Use the Output Assistant to create a “Scan of Concentrations. Volumes and Global Quantity Values” plot. Use a log scale for the x-axis (MAPKKK_activator). Run the scan to get the dose response curves.
- What is happening? The very weird curves come the fact that several unstable steady states have been plotted in addition to the stable steady states. To view only the steady states in the plot, we need to change the settings for Steady-State (Under Tasks) – Change the Use Newton parameter from 1 to 0 and run the plot again.
- Hide all of the curves except for the normalized active form of MAPK (“K_PP_norm”), MAPKK (“KK_PP_norm”) and MAPKKK (“KKK_P_norm”).
- Save your model (e.g myHuang1996-parameterScan.cps)



“use Newton“=1



“use Newton“=0

This plot teaches us two different, very important, characteristics of the MAP kinase cascade.

1. The half activation of ERK happens at lower RAS concentration ($4.4 \mu\text{M}$) than that of MEK ($13 \mu\text{M}$) or RAF ($260 \mu\text{M}$).
2. The slope of the curve for ERK is steeper than for MEK, itself steeper than for RAF

Such a cascade of covalent modifications provided a new mechanism to generate ultrasensitivity, in addition to the allosteric regulation of multi-site multimeric proteins. In addition, this is a brilliant demonstration of the emergent properties of a biological system, and the absolute need to study the behaviour of the whole system. An *in vitro* reconstitution of RAS and RAF would tell us that RAS “ EC_{50} ” is $260 \mu\text{M}$, and its dynamic range (the range of signal giving 10% to 90% of response) of [$10 \mu\text{M}$, 2.4mM]. While in fact, if we consider that ERK is the final readout, the “ EC_{50} ” is 100 fold lower at $4.4 \mu\text{M}$, and the dynamic range of [$3 \mu\text{M}$, $7.4 \mu\text{M}$]. That makes the MAP kinase a thresholding system, which transforms a continuous quantity into a binary response, while RAF transmits a quantitative measurement below. NB: ERK on its own would not produce a threshold when stimulated by MEK. The behaviour emerges from the stacking of the three protein activations.

2) MAPK cascade - Oscillations:

Model:

BIOMD000000010 – Kholodenko 2000

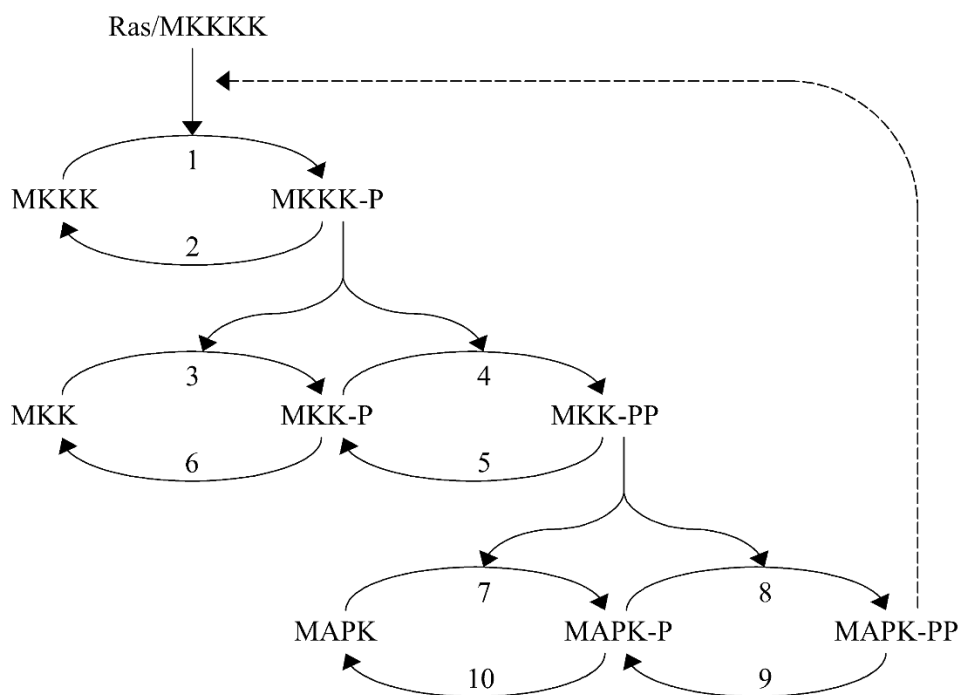
Kholodenko BN. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. Eur J Biochem. 2000 Mar;267(6):1583-8.

(<http://www.ebi.ac.uk/biomodels-main/BIOMD000000010>)

Inhibitory phosphorylation of SOS by p42/p44 MAPK (ERK) provides a mechanism for switching off Ras signalling. This inhibition creates a negative-feedback in the MAPK cascade (below). Indeed, while tyrosine phosphorylated Raf brings ERK activation, ERK mediated inhibition of Raf stimulation by SOS decreases ERK phosphorylation. The paper by Kholodenko attempted to show that a combination of negative feedback with ultrasensitivity can lead to sustained biochemical oscillations.

Here, we will see the effect of negative feedback (MAPK on its activator MAPKKK) on the dynamics of MAPK signalling cascade.

The topology of the reaction network of this model is identical to Huang and Ferrell's model (the model we used in the previous example), apart from the negative feedback. However, for this model, all reactions were modelled as simple Michaelis-Menten kinetics, unlike in Huang and Ferrell model where the reactions are modelled using mass-action kinetics.



Schematic view of MAPK cascade with negative feedback loop. (Kholodenko 2000)

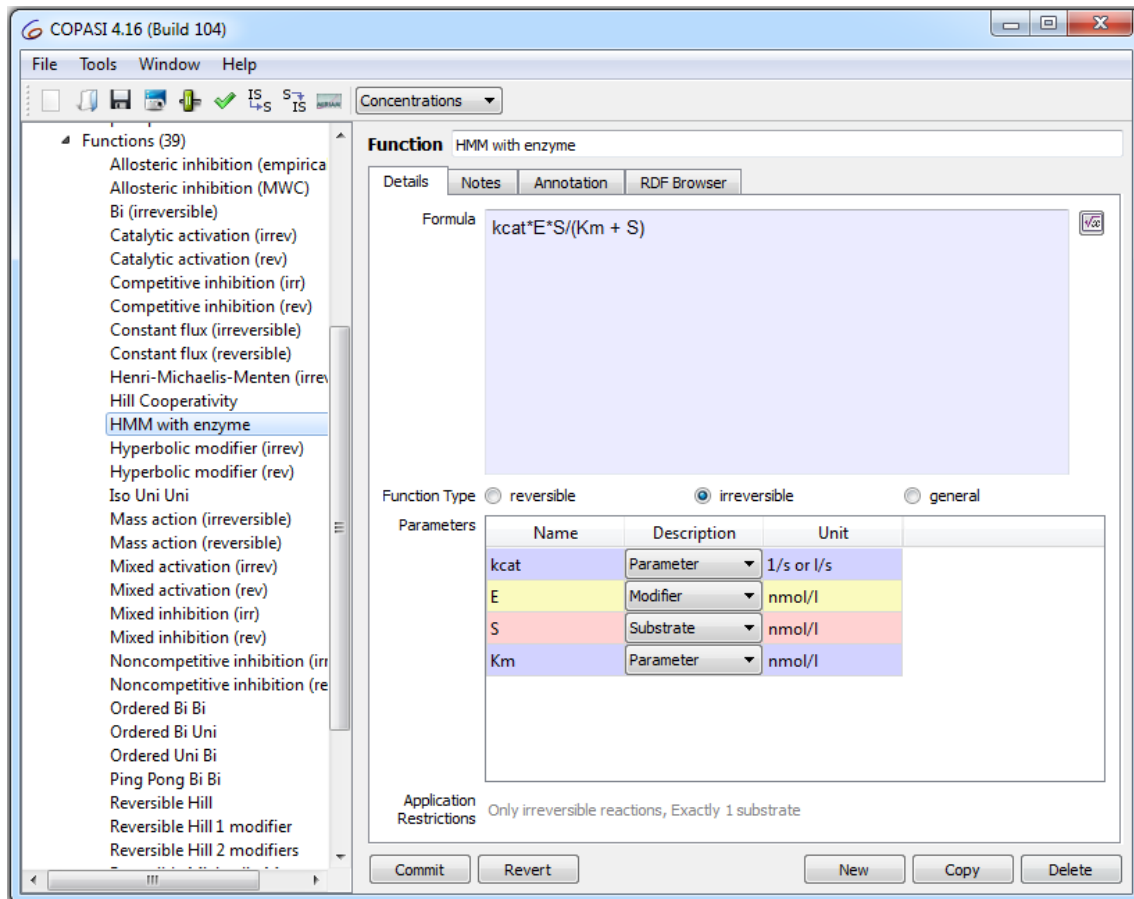
2.a Building the model without the negative feedback:

- Open a new COPASI file, and set the name of the model to “Kholodenko2000”
- Set the units: Time = s, Volume = l, Quantity = nmol
- Create a new compartment (Model>Biochemical>Compartments), name it “cell” and set the initial volume to 1

For the purpose of this tutorial we need the form of the Henri-Michaelis-Menten law that uses k_{cat} and the concentration of enzyme. COPASI only comes with the classic “Michaelis-Menten” law using V_{max} .

- Create (and commit) the following function (Under Functions):

Function : HMM with enzyme
Formula : $k_{cat} \cdot E \cdot S / (K_m + S)$
Function type : irreversible
Parameters : Change the description for S to substrate
Change the description for E to modifier



- Create (and commit) the following parameters as Global Quantities (Model>Biochemical>Global Quantities):

Parameter	Initial Value
V1	2.5
K1	10.0
V2	0.25
K2	8.0
KCAT	0.025
KM	15.0
V5_6	0.75
V9_10	0.5

- Add the following reactions (Model>Biochemical>Reactions):

(HMM = Henry-Michaelis-Menten (irreversible) ; HMMenz = HMM with enzyme)

Name	Equation	Rate Law	Mapping
v1	MAPKKK -> MAPKKK-P	HMM	$V \rightarrow V1; K_m \rightarrow K1$
v2	MAPKKK-P -> MAPKKK	HMM	$V \rightarrow V2; K_m \rightarrow K2$
v3	MAPKK -> MAPKK-P ; MAPKKK-P	HMMenz	$k_{cat} \rightarrow K_{CAT}; K_m \rightarrow K_M; E \rightarrow \text{MAPKKK-P}$
v4	MAPKK-P -> MAPKK-PP ; MAPKKK-P	HMMenz	$k_{cat} \rightarrow K_{CAT}; K_m \rightarrow K_M; E \rightarrow \text{MAPKKK-P}$
v5	MAPKK-PP -> MAPKK-P	HMM	$V \rightarrow V5_6; K_m \rightarrow K_M$
v6	MAPKK-P -> MAPKK	HMM	$V \rightarrow V5_6; K_m \rightarrow K_M$
v7	MAPK -> MAPK-P ; MAPKK-PP	HMMenz	$k_{cat} \rightarrow K_{CAT}; K_m \rightarrow K_M; E \rightarrow \text{MAPKK-PP}$
v8	MAPK-P -> MAPK-PP ; MAPKK-PP	HMMenz	$k_{cat} \rightarrow K_{CAT}; K_m \rightarrow K_M; E \rightarrow \text{MAPKK-PP}$
v9	MAPK-PP -> MAPK-P	HMM	$V \rightarrow V9_10; K_m \rightarrow K_M$
v10	MAPK-P -> MAPK	HMM	$V \rightarrow V9_10; K_m \rightarrow K_M$

COPASI 4.16 (Build 104)

File Tools Window Help

Concentrations

Model

- Biochemical
 - Compartments (1)
 - Species (8)
 - Reactions (10)
 - v1
 - v10
 - v2
 - v3**
 - v4
 - v5
 - v6
 - v7
 - v8
 - v9
 - Global Quantities (8)
 - K1
 - K2
 - KCAT
 - KM
 - V1
 - V2
 - V5_6
 - V9_10
 - Events (0)
 - Parameter Overview
 - Parameter Sets (0)
- Mathematical Diagrams
- Tasks
- Output Specifications

Reaction v3

Details Notes Annotation RDF Browser

Reaction MAPKK -> MAPKK-P; MAPKKK-P

Reversible Multi Compartment

Rate Law HMM with enzyme New Rate Law

Flux (nmol/s) 0 Edit Rate Law

Symbol Definition

Role	Name	Mapping	Value	Unit
Parameter	kcat	KCAT	0.025	1/s
Modifier	E	MAPKKK-P		nmol/l
Substrate	S	MAPKK		nmol/l
Parameter	Km	KM	15	nmol/l

Commit Revert New Copy Delete

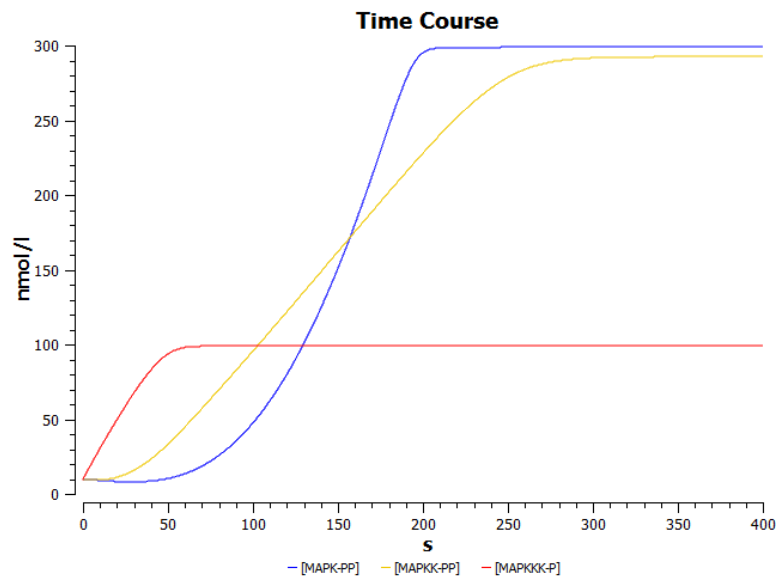
- Set the initial concentrations for the following species:

Species Name	Initial Concentration
MAPKKK	90
MAPKKK-P	10
MAPKK	280
MAPKK-P	10
MAPKK-PP	10
MAPK	280
MAPK-P	10
MAPK-PP	10

- Save your model (e.g myKholodenko.cps)

2.b Running a time course simulation with our model:

- Create a time course (as in Exercise 1), with a Duration = 400 secs and Interval Size = 0.1 secs.
- Create a “Concentrations, Volumes and Global Quantity Values” plot and run the time course.
- Show only the curves for the normalized active form of MAPK (MAPK-PP), MAPKK (MAPKK-PP) and MAPKKK (MAPKKK-P).



Do we retrieve the results obtained in 1.a? What is different in the curves representing RAF (MAPKKK)? What causes the difference?

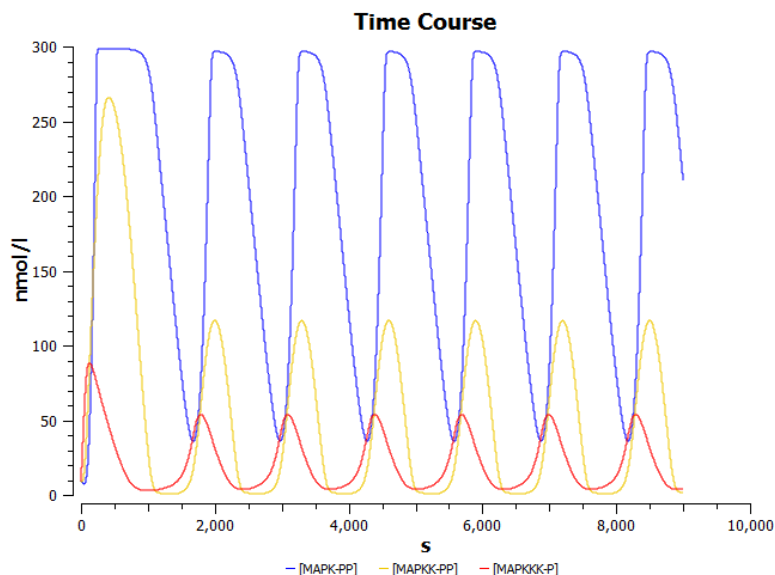
2.c Introduce negative feedback to the model:

- Create a new function:
Function : HMM with inhibition
Formula : $V \cdot S / ((1 + (I/K_i)^n) \cdot (K + S))$
Function type : irreversible
Parameters : Change the description for S to substrate
Change the description for I to modifier
- Change the rate law of MAPKKK activation (reaction v1) to the new function “HMM with inhibition”, and set the parameter values as follows (first, set the “mapping” to “--local--”)

Parameter	Value
Modifier	MAPK-PP
V	2.5
K _i	9
K	10
n	1

- Save your model (e.g myKholodenko_nf.cps)
- Run a time course as above, with a duration of 9000 seconds

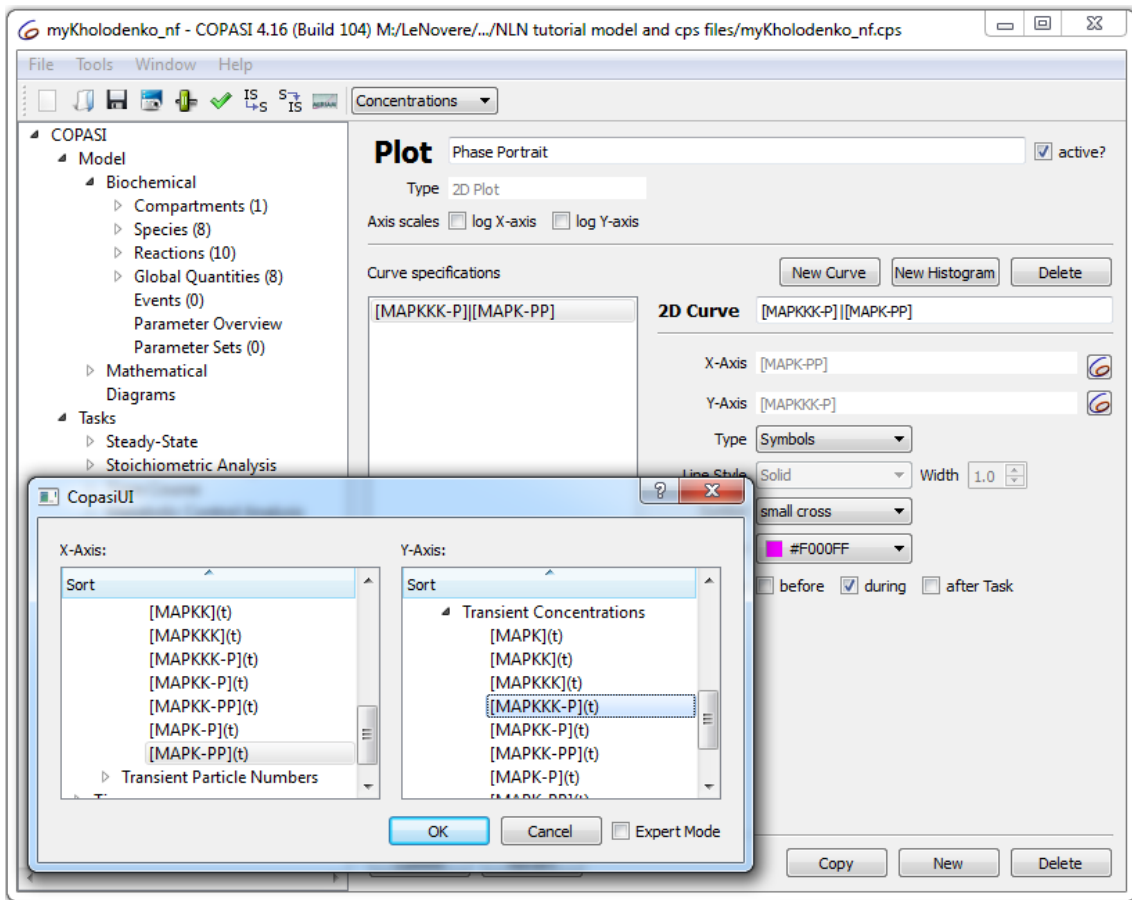
What happens? What causes the dynamical behaviour you observe? Would negative feedback always produce such a result?



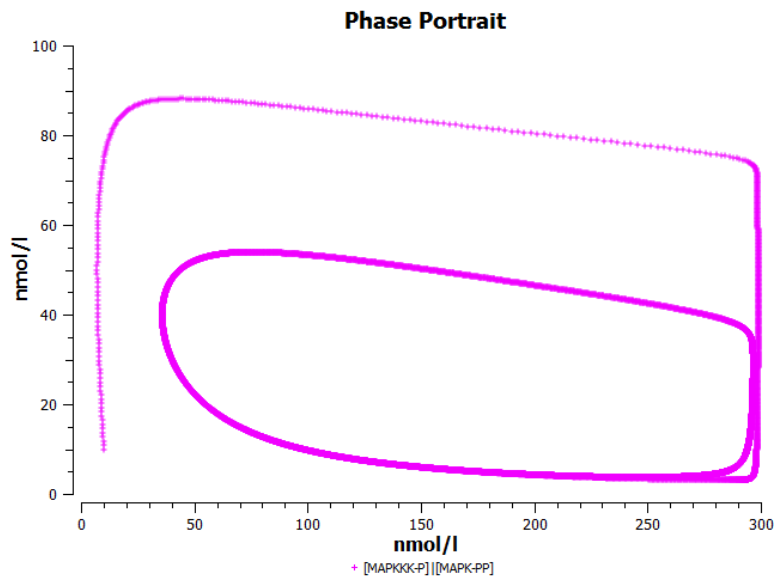
[2.c cont'd - optional/advanced]

Plotting the values of state variables (here concentrations) versus time is not the only way to depict the dynamical behaviour of a system. Other graphical representations can reveal regulatory structures. A plot frequently used to display dynamical behaviours is the phase portrait, where we plot the value of a variable versus the value of another over time.

- Create a plot that reports MAPKKK-P as a function of MAPK-PP. Rerun the simulation.

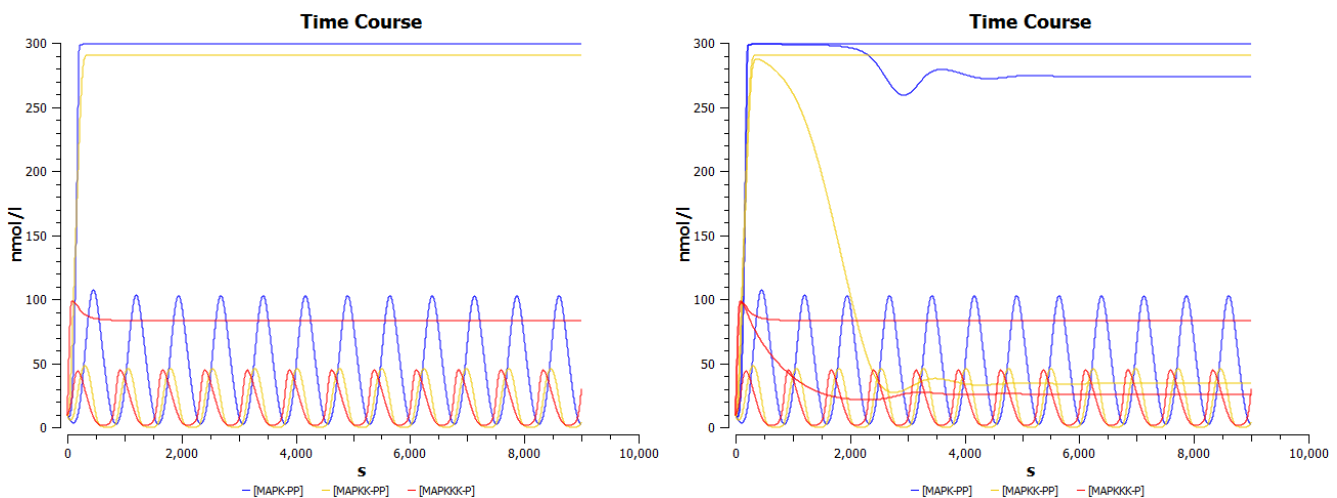


What do you observe? \Can you predict what such a plot would look like for a steady state? A damped oscillation? A bi-stable switch?



2.d Dependence on inhibition:

- Select Parameter Scan, and change the Task type to “Time Course”, then click “Create”.
- Vary K_i (Reactions>Reaction Parameters> v_1 (MAPKKK activation)> K_i) from 1 to 51 in a linear fashion (start with one interval). This causes a decrease of the inhibitory feedback (K_i indicates the concentration of the inhibitor needed for half-maximal inhibition).
- Run the scan and view the plot (you don’t need to create new plots - they were already generated in the previous section).
- Note the changes in the time course, compared to that of the plot obtained in the previous section.
- If we make 2 intervals, then the point where the oscillation begins to damp can be observed. Try a few more intervals.

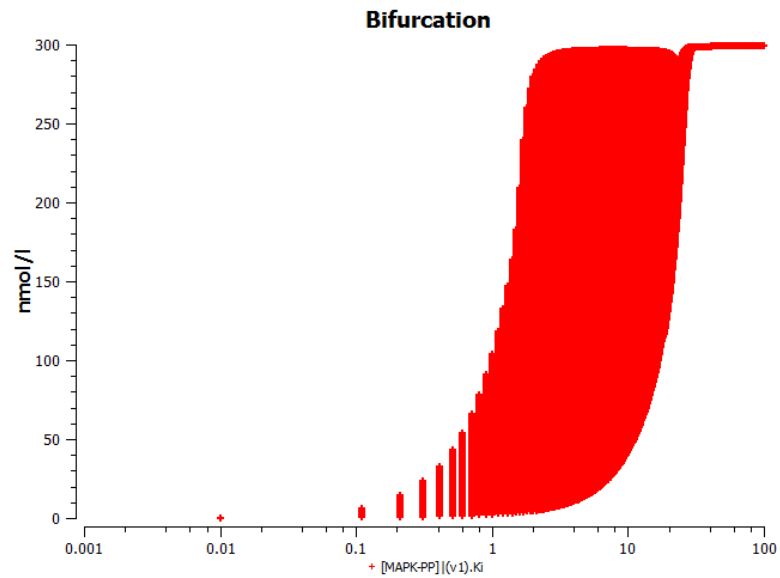


[2.d cont'd -optional/advanced]

The point at which the behaviour switches from reaching steady-state to stable oscillations is called a bifurcation. We can try to pinpoint the respective values of K_i .

- Create a plot “bifurcation”, that will display MAPK-PP as a function of reaction v_1 's parameter K_i . Select a log X-axis, “Symbols” for type and data capture during the tasks.
- Modify the time course to suppress output before 2000 s (I should still stay on a duration of 9000 s. Interval size of 1 are fine).
- Create a time course parameter scan, exploring logarithmically the values of v_1 's parameter K_i from 0.01 to 100 (1000 intervals)
- Run it.

Can you pinpoint the value of K_i for which we start to see big stable oscillations? What happens below? For which value of K_i do the oscillations disappear? What happens above? [look at what happens to the time course. Art and Science ...]



Examples of oscillatory behaviours in MAP kinase cascades have since then be shown experimentally.

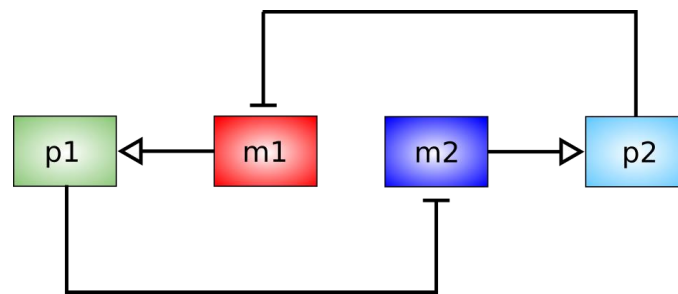
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2602854/>

<https://www.ncbi.nlm.nih.gov/pubmed/18384751>

3) Stochastic simulations

So far we have simulated the models with a “deterministic algorithm”. The numerical solver computed the changes in the concentrations of molecular species based on ordinary differential equations reconstructed from all the reactions involving them. This approach presents two problems. Firstly, each time we simulate the model, we will get the same result. There is no “noise” taken into account. Secondly, those solvers use continuous variables. So in small volumes, we can get half a molecule reacting with an eighth of another one to form a third of a complex. A side effect is that if there is a discrete decision to be made, the solver won’t be able to choose, but will “cut the pear in half”. Therefore, if small amounts are involved, it is often useful to use a “stochastic algorithm”. There are several classes of such algorithms. The most used in systems biology is based on the independent simulation of all reactions, random numbers being used to decide which reaction to fire and when.

To demonstrate the usefulness of such a simulation framework we will use a “toy model”, representing a system able to exist in two different states. We have two genes, whose activity will be modelled using the concentrations of their messenger RNAs and the corresponding proteins. Each protein represses the expression of the other gene.



3.a System of two non-interacting genes:

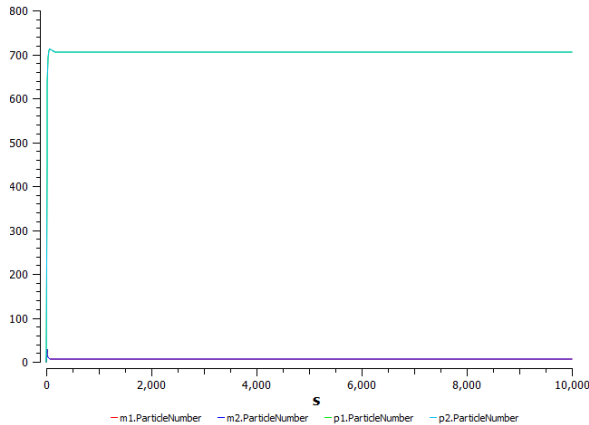
- Open the COPASI file “Switch-noFeedback.cps”
- Go to “Mathematical>>Matrices” and look at the stoichiometry matrix. Does that make sense? Look at “Differential equations”. Is the effect of mRNA clear?
- Run a time course simulation of 1000 s with the LSODA method. As the output select “Particle Numbers, Volumes, and Global Quantity Values”.
- Run a time course simulation of 1000 s with the Gibson-Bruck method
What are the difference between the two results? Are the average number of protein reached the same between both simulations? Hide the proteins. How is the noise on mRNAs compared to proteins? Why?

3.b System of two interacting genes:

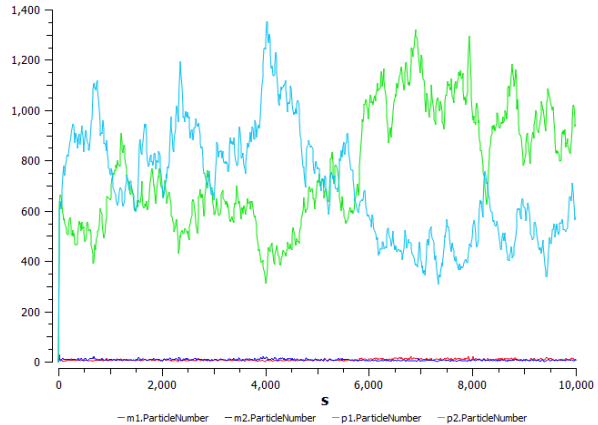
We will now incorporate the reciprocal inhibitions.

- Open the COPASI file “Switch-Feedback.cps”
- Go to “Mathematical>>Matrices”. Did the matrix change? Look at “Differential equations”. Do you understand how p1 inhibits the production of m2 and vice-versa?
- Run a 10000 s simulation with the LSODA algorithm. What happens? Can we see two different states? Change the initial amount of the protein p1 to 100. What happens? What will happen eventually?

- Change p1 back to 0. Run a 10000 s simulation with the Gibson-Bruck algorithm? What happens? Run the simulation several times. Is the result always the same?
- Change p1 to 100. Run the simulation. Again. Again. What are the differences with the previous situation?



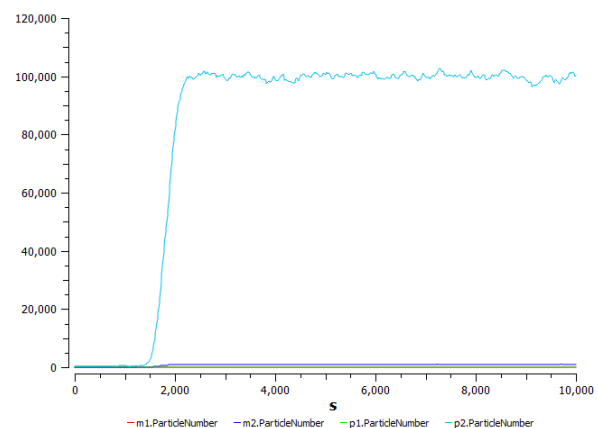
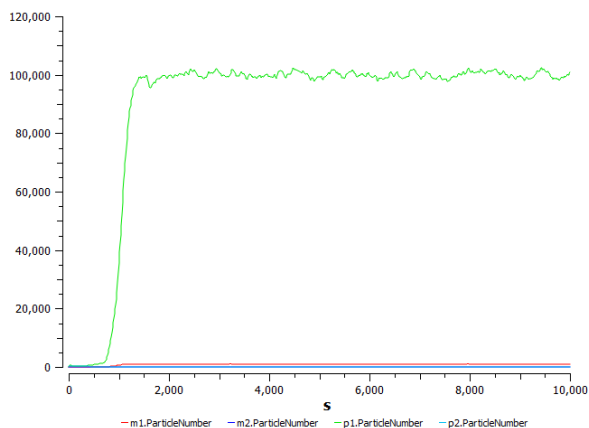
Simulation with LSODA



Simulation with Gibson-Bruck

So, this is a switch, but it is unstable. If we start with a symmetrical situation, we get oscillations, going from p1 high to p2 high. One way of getting a more robust decision system is to incorporate cooperativity in the repression. The most iconic example of such a system is the lysis-lysogeny mechanism of E coli in response to lambda phage infection (https://en.wikipedia.org/wiki/Lambda_phage).

- Change p1 back to 0. Change the value of the global variables n1 and n2 from 1 to 1.5.
- Run the simulation. Again. Again. What happens? Is it always the same protein that wins? Do we have oscillations?



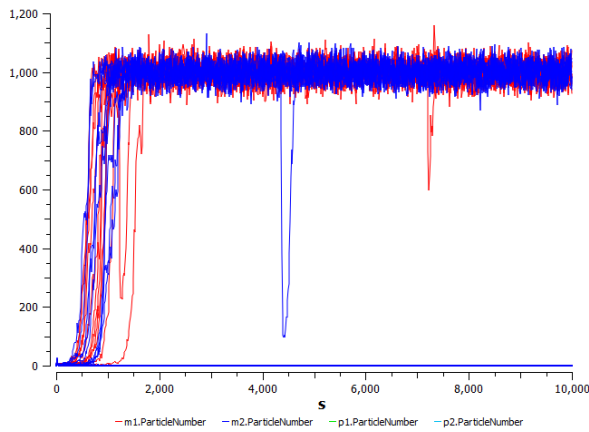
Cooperative inhibition

As we saw in exercise 1.b, we can run many simulations at once with COPASI. Let's do that, but instead of a parameter scan, where we change the value of a parameter, we will just repeat the same simulation.

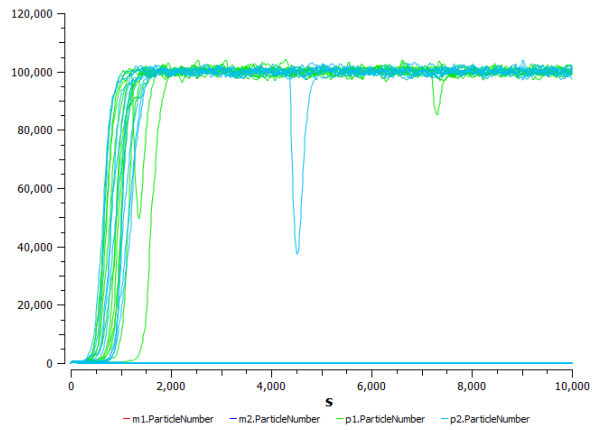


- Go to parameter scan. For “New scan item”, select “repeat”. Set the number of iterations to 20, and the Task to Time Course.
- Launch the scan. Be patient, it will take a while!

How many times does p1 win? To count more easily, select the rising phase with the mouse (this should be 0-2000 seconds) to zoom in.



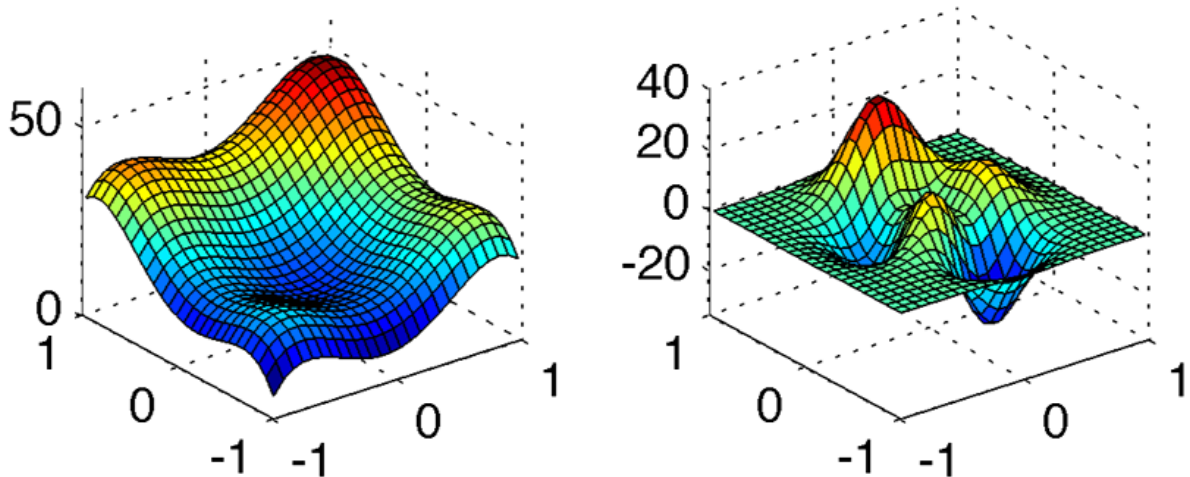
20 simulations – mRNA time courses



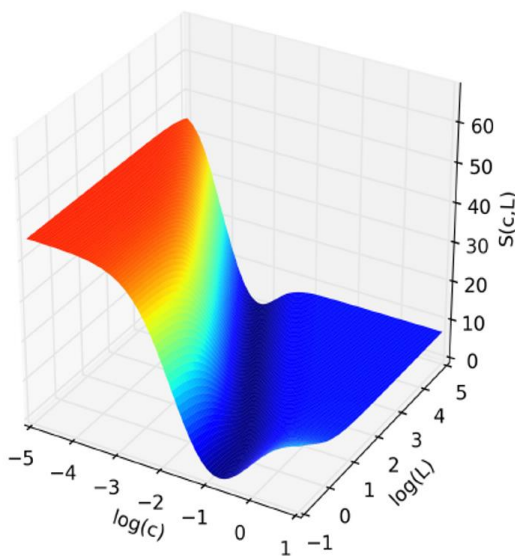
20 simulations – protein time courses

4) Estimating parameters using experimental results

A very important feature of the modelling and simulation procedure in systems biology is the estimation of parameter values based on experimental measurements. Sometimes accurate values can be found in the literature or in databases. However, most often they are not available or need to be adapted to the context. Therefore, we need to estimate the parameters, that is to find the values that provide the best reproduction of observed quantities. (Note that those quantities do not necessarily need to be explicitly in the model, but they need to be related to variables of the model). To do that, we set parameter values, we run simulations and we compute the difference between the results and the observations. Below are two surfaces showing this difference for a 2 parameter system.



On the left, the system presents only one “dip” showing the set of parameters that give the optimal solution (Notice something odd though?) On the right, the systems present two dips, representing a local and an absolute minimum.



Note that sometimes, there are no optimal sets of parameters. The image on the left shows a system with two coupled parameters. If we fix one, then there is an optimal value for the other. However, all the combinations corresponding to the “valley” are equally good. One says that the model is not “identifiable”.

How do we choose the parameters to try? We could systematically sample all parameter values and choose the ones that fit better. However, such a procedure would be prohibitively long if we want sufficient granularity. Moreover, it might be that our sampling misses the actual optimal points. Instead we use algorithms that try to optimise the selection of parameter sets based on the result of the previous ones. There are many such methods, and a good sample is provided by COPASI.

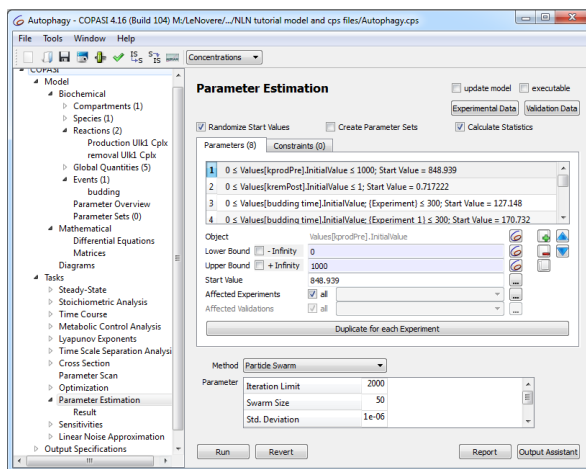
- Open the COPASI file “Autophagy.cps”

Explore the model. This is a simple model representing the accumulation of Ulk1 complex during the formation of an autophagosome and its disappearance after budding of the vesicle. In the initial phase, the Ulk1 complex aggregates with a constant flux (this assumes an unlimited supply of the complex from the cytosol). The removal rate is 0. The budding is represented by a discrete event.

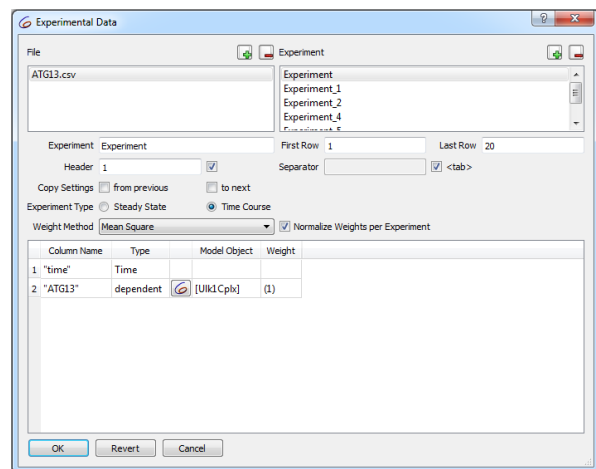
- In “Model>Biochemical”. Click on the “Budding” “Event”.

After the budding, the creation rate is turned to zero, and the removal rate is set up, so that a unimolecular removal takes place, that is a reaction in which velocity is proportional to the concentration of the complex. We have quantitative recordings of live fluorescent imaging of ATG13, a protein part of the Ulk1 complex. We will use these recordings to determine the values for the creation of Ulk1 complex before budding and removal after budding.

- Go to “Tasks>Parameter Estimation”. The list of parameters shows what we will estimate. “kprodPre” will be estimated using all experiments, as will “kremPost”. Lower and upper bounds are set using reasonable assumptions for constant flux and first order rates. On the contrary, the budding time will be estimated independently for each experiment since autophagosomes take a variable time to grow.
- Click on “Experimental Data”. Here you see that we import the data from a file “ATG13.csv”. We use 6 of the 9 experiments contained by the file. For each experiment, we identify the recording columns (here ATG13) and the corresponding model variable ([Ulk1Cplx]). Try to infer what the other options mean.

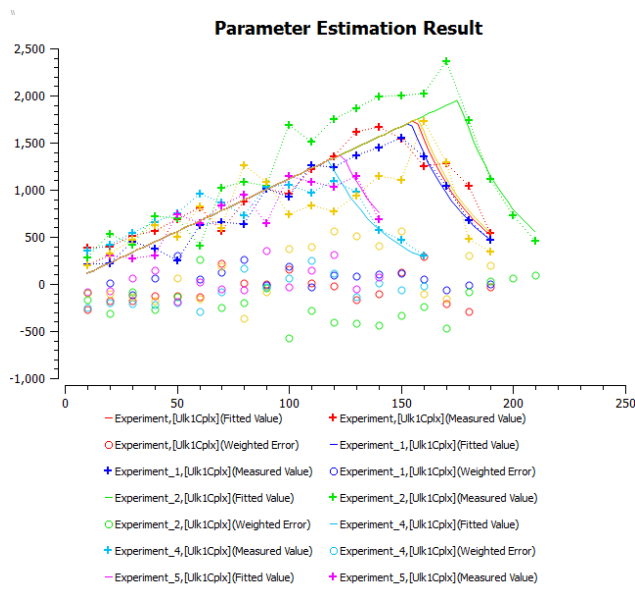
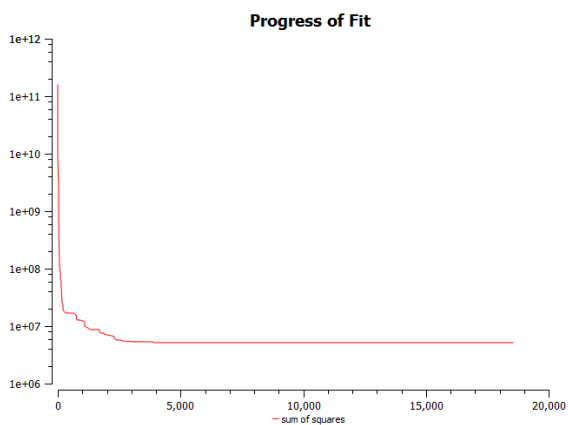


Variables to estimate.



Readout to simulate.

- Close the Experimental Data window.
- We now need to choose the algorithm used to sample the parameters. Try the following: Random search, Steepest Descent, Genetic Algorithm, Particle Swarm. Ignore the messages about the Fisher matrix.
- Observe what happens on the plot “Progress of fit”. The plot “Parameter Estimation Results” shows the experimental measurement (crosses), the error (circle) and the fitted curves (continuous lines). Are all methods equivalent?



The scoring function decreases while running a parameter estimation procedure.

Experimental data, fitting kinetics and errors.

Depending on the structure of the multidimensional objective function, some methods will be better than others. Some are faster, but more sensitive to local minima. Some methods are better at avoiding local minima but are not very precise etc. In addition, a good parameter estimation protocol would run such parameter estimation many times (hundreds or thousands) and study the distribution of the resulting fitting scores in function of the estimated parameter values. All that is beyond the scope of the present course.