Kinetic Modeling With Copasi

Fachkurs Theoretische Biophysik, WS 2016/17

Copasi[1] is a software application for simulation and analysis of biochemical networks. It is available for MS Windows, Linux and Mac OS X[2] and can import and export models in the standardized SBML format. This introductory course will explain how to implement simple biochemical models as systems of ordinary differential equations in the graphical user interface (GUI) of Copasi. The major capabilities of Copasi for the analysis of the implemented model will be presented. The introduction is followed by some hands on exercises with different models and functionalities.

This accompanying document is a Copasi reference rather than a manual. A very detailed and easy to understand Copasi manual can be acessed on http://www.copasi.org. If you encounter any problems with Copasi that cannot be answered in this course, you can also check the User Support Forum there. Online tutorials are available at:
http://copasi.org/Support/Video_Tutorials/
http://vimeo.com/user3657452/videos/sort:date

[1] Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu I, Mendes P, Kummer U. *COPASI: a COmplex PAthway SImulator.* **Bioinformatics** 2006; 83:3067–3074

[2] available at http://www.copasi.org

Contents

1 Copasi

A model consists of species and interactions between species. Copasi allows to compile these elements in a structured way. Different aspects of the model as well as Copasi functions are browsed via a menu on the left side of the main window and edited inside the main window (see Figure 1). Plots for different tasks are issued in additional windows. We will walk through some of the submenus:

Model - *describe model, set units, see* **2.1**

- Biochemical - *edit model, see* **2.2**

  Compartments

  Species

  Reactions - *define reactions, see* **2.3**

  Global Quantities - *parameters and assignment variables, see* **2.4**

  Events - *discrete events, see* **2.5**

  Parameter Overview - *view parameters*

- Mathematical - *view equations and matrices*

  Differential Equations - *view model equations, see* **2.7**

  Matrices - *view stoichiometry*

  Diagrams - *create simple network diagrams of your model*

Tasks - *model simulation and functional analyses, see* **3.1**

- Steady State - *find and analyze the steady state of the model, see* **4.1**
- Time Course - *simulate the model, see* **4.1**
- Parameter Scan - *simulate the model varying parameters, see* **4.3**

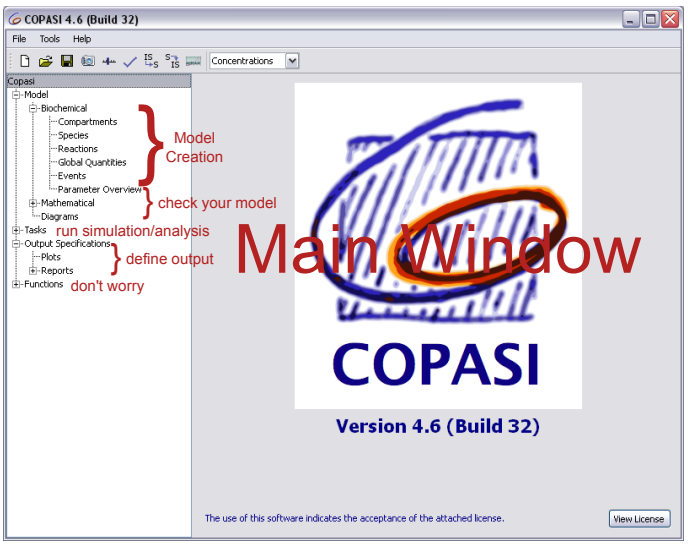

Figure 1: Copasi upon start-up. The function of different submenus is indicated. The main window will display different contents depending on the active submenu.

- other advanced tasks: Parameter Estimation, Optimzation, Linear noise approximation...

Output Specifications

- Plots - *add, change or remove plots* **3.2**
- Reports - *add, change or remove reports of tasks to be saved externally* **3.3**

Functions - *add, change or remove functions for reaction rate laws* **2.6**

Units - *Predefined units that can be used in the model to ensure consistency*

An important and ubiquitous **button** is the one with the **Copasi icon** on it ("curly button"). It is used **to select** previously defined **model entities** in different contexts.
Another important concept in fields that require free input: **Blue** background means (formally) correct. **Red** background indicates formal mistakes.

2 Creating a Model

2.1 Model

Define model **units** and **describe** the model (e.g. leave notes to remember what you want to do).

2.2 Biochemical - compartments and species

Define model variables and the compartments that contain them.
Fixed species have a fixed particle number. Compartment volume changes will change their concentration!

2.3 Biochemical - reactions

Define reactions (Figure 2). Chemical equations use **'+' '->' '=' ';' '*'** where **each operator is separated by blanks**. Any name in the chemical equation that is neither an operator nor a species will **create a new species**. Depending on the stoichiometry of the chemical equation, a kinetic rate law can be chosen from the drop-down list of possible kinetic laws. It is also possible to create a specific new kinetic law.
The bottom part gives an overview of the assignments of numbers, species and parameters to the kinetic law.

2.4 Biochemical - global quantities

Define global parameters as **fixed** values, **assignment** variables or **ODE**. Only if the moedl parameters are defined here, they will be accessible to the tasks manipulating parameters such as Parameter Scan or Parameter Estimation.
In the case of **assignment** variables, the empty field is used to enter the **right hand side** of the equation determining the assignment variable (see Figure 3). Use the **Copasi-button** to select model variables to be used in the equation.

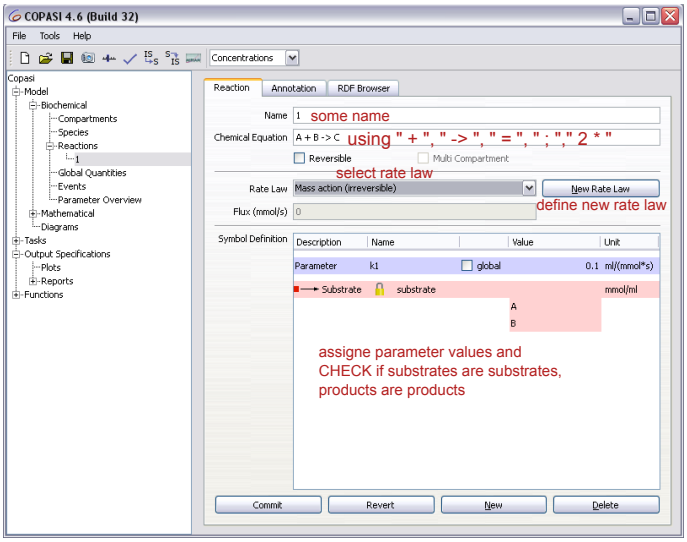

Figure 2: Creating reactions in Copasi.

2.5 Biochemical - events

Events are used to define assignments that happen **once** a certain condition is fulfilled. This could for example be the increase of a concentration across a certain threshold or a specific time. Such trigger expressions can be defined using model variables (via the curly button) and relation operators '**lt**' (less than), '**le**' (less equal), '**gt**' (greater than), '**ge**' (greater equal), '**eq**' (equal).
The curly button is also used to define event targets, *e.g.* what happens when the event is triggered. Be careful to not leave any empty assignments, when changing event targets, as this causes errors.

2.6 Functions

Functions are used to define rate laws for the model reactions. There is a number of predefined rate laws that can't be changed. If you define additional rate laws, they are stored here. If a function is

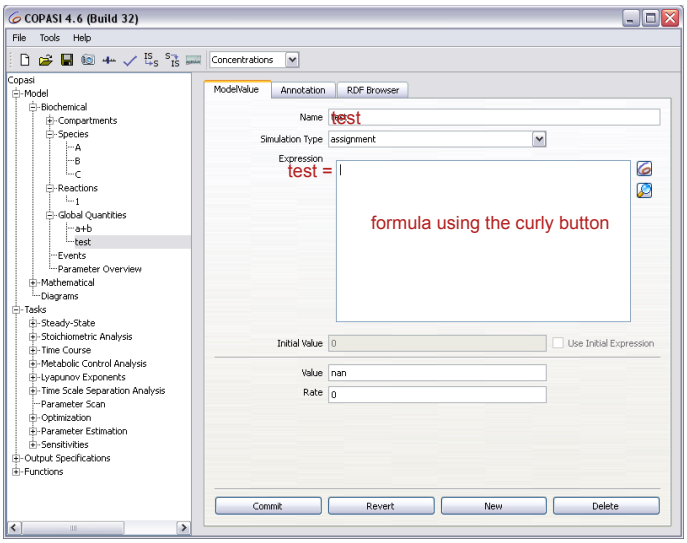

Figure 3: Creating assignment quantities in Copasi.

in use in a reaction it can only be modified to some extent!

Formula: The formula can be entered freely using a wide range of operators. Just make sure you are mathematically consistent, *e.g* never divide by zero. The names of the species and parameters in the function do not need to match the names of the reacting species. They are mapped to the reacting species for each reaction in the Biochemical - reactions section.

Function type: Checking **General** puts you on the safe side.

Parameters: Here, you define which name in the formula corresponds to which type of model entity (substrate, product, modifier, parameter,...). Make sure that this classification complies with the chemical equation of the reaction you want to use this formula for (a reaction without products can not be assigned a formula that uses products).

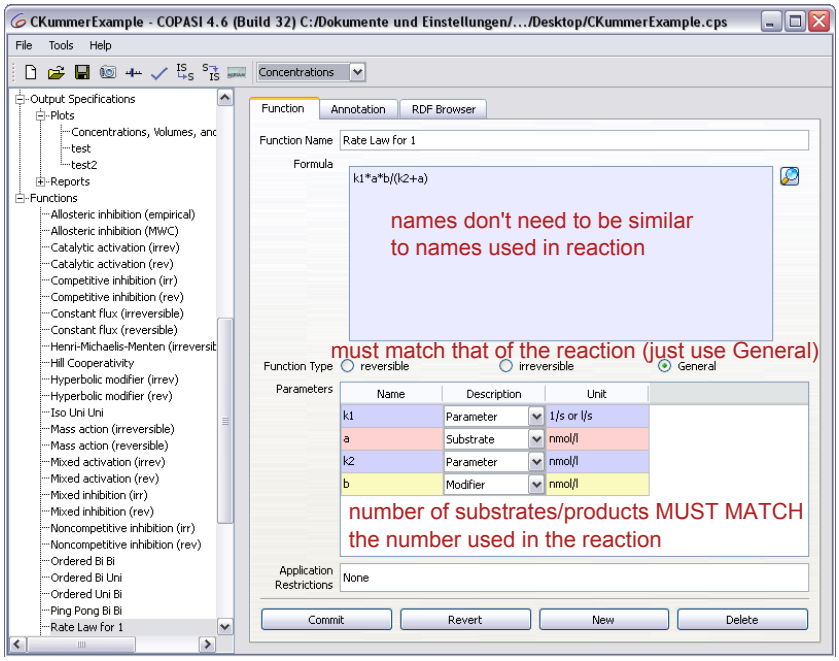

Figure 4: Adding a new function to Copasi.

2.7 Mathematical - Differential Equations

View the equations generated from the reactions and global quantities. Be sure to **check** if everything is correct. Also, notice that most reaction rates take the **volume of the containing compartment** into account. **Transport reactions** between compartments don't.

3 Model Analysis Using Copasi

3.1 Tasks

Copasi offers a number of analyses that can help you to characterize and understand the function of your model and the underlying biological process. We will go through some of the basic tasks that cover the following topics:

Time Course - *simulate your model's evolution over time, see* **4.1**

Steady-State - *find and analyze steady state, see* **4.2**

Parameter Scan - *Systematically change parameters of the model and run analyses on the different parameter sets, see* **4.3**

Sensitivities - *analyze how model parameters affect model variables, see* **4.4**

Stoichiometric Analysis - *analyze the model structure, see* **4.5**

- Elementary Modes - *find pathways that can generate a steady state*
- Mass Conservation - *check if input equals output*

Metabolic Control Analysis - *compute elasticities and control coefficients, see* **4.6**

Further advanced tasks - *Lyapunov exponents, Time Scale Separation Analysis, Cross Section, Optimization, Parameter Estimation, Linear Noise Approximation – not covered here but also cool, see the Copasi homepage for details.*

Most of these tasks do no generate graphical output, but the results are displayed in the respective 'Result' submenu. For some, the numbers can be visualized using bar graphs there. **Numerical results** are usually accessible via the respective **'Result' submenu**. **Plots** in extra windows **need to be created** before execution of the task. For more detail, see **3.2**. For more detailed numerical output, Reports in form of text files can be generated, see **3.3**.

3.2 Plots

The results of each task can be plotted as x-y-graphs within Copasi. The plots appear as external plots in extra windows and are somewhat **independent** of the main program. Plots are not generated automatically, nor are they destroyed automatically when switching tasks. A plot receives signals from Copasi and updates accordingly, even if these signals do not make sense. Since Copasi is in active development, new tasks (and also subtasks) are frequently added, so not all tasks might be covered in this tutorial. You can also **create a new plot** with the help of the Output Assistant for each task or individually in the Plot submenu here. To do so, you can enter a new name into the last field in the list of plots. This plot can then be filled with curves, histograms, banded graphs or contour plots of model variables, which you can select using again the curly button. You have the choice to plot variables as functions of time or as functions of other variables. In case you created too many plots using the Output Assistant, this submenu is the place to **remove plots**. In a plot window, you can save the image and the raw data.

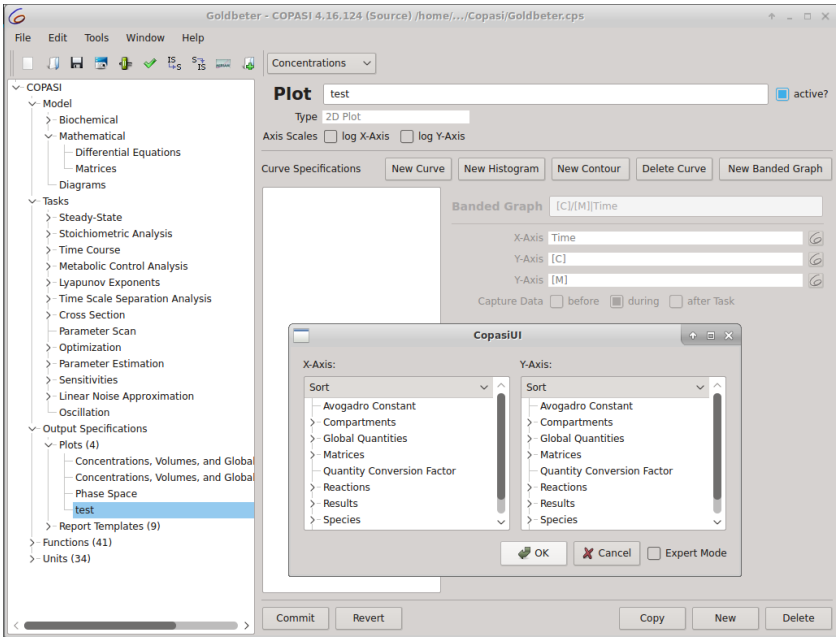

Figure 5: Creating a new plot.

You can change a **plot's status from active to inactive**. Inactive plots are not updated. You can inactivate a plot, change the model, create a new plot and compare the results.

3.3 Reports

For each task, a written report can be generated that summarizes all results and is saved as a text file outside of Copasi. There is a predefined report available for each task which you can activate using the Report button of each task. You can also define custom reports in the respective submenu.

4 Analysis and Simulation Tasks

4.1 Simulating the Model

This task allows you to simulate the trajectories of your model species over time.
In the Time Series The inputs on the top determine duration of simulation and step size. Too many **intervals increase computation** time, **too few lead to erroneous trajectories**.
Copasi can simulate deterministically and stochastically (required for systems with very low numbers of molecules), you can select the respective integration algorithm in the 'Method' tab. Each method requires parameters that control numerical precision. **Before** running **stochastic** simulations, check that the **particle numbers are low**, otherwise an error will occur.
To **plot** the simulation results, use the **Output assistant button** at the bottom to create a plot, as introduced before. The plots occur in an extra window and will be updated every time a new task is executed. Therefore, you don't need to add plotting windows for each simulation you run.
Be careful with the **'update model'** check-box: If marked, the initial concentrations will be overwritten with the concentrations at the end of the simulation!
Suggested Output Assistant Plots: the first 7.

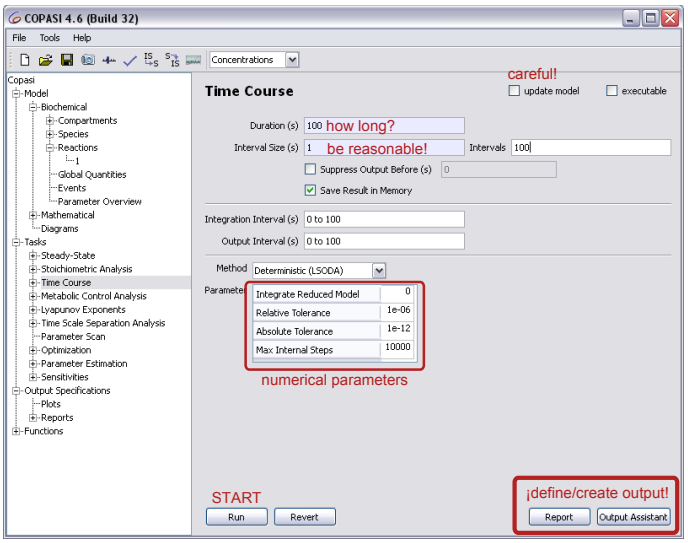

Figure 6: Model simulation with Copasi.

4.2 Steady-State

This task numerically searches for a steady state. The stability of the steady state can be assessed using the Jacobian (the steady state is stable if all eigenvalues are negative), which is summarized in the Stability analysis.

4.3 Parameter Scan

If you now want to explore how single parameters of your model influence its time evolution you can conduct a Parameter Scan. It simulates the model after varying one or more parameters. Which parameter to scan can be selected with the curly button. You can choose between scanning a certain

range for a parameter value systematically or a random distribution of parameters (with a Repeat for more than one set of random parameters). For each interval (Scan) or iteration (Repeat), Copasi will run the specified task (usually time-course, but any other task can be chosen as well). Increasing the number of scan parameters increases computation time. Also, make sure you have appropriate plots or reports to collect the outputs of the scanned simulations for each task.

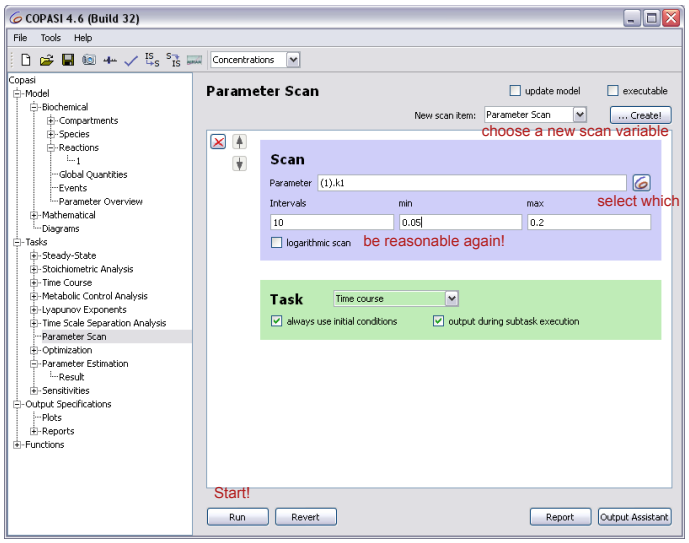

Figure 7: Parameter scan dialog

Scan: Runs simulations varying the specified parameter using the intervals defined.
Random Distribution: Samples specified parameter from a random distribution **once**. To run multiple times, needs a Repeat-item **before**.

4.4 Sensitivities

Sensitivities, also called response coefficients, express the direct dependence of steady state variables on parameters. The sensitivity of a flux J_j to a parameter p_m is given by

$$R_m^j = \frac{p_m}{J_j}\frac{\partial J_j}{\partial p_m}$$

and the sensitivity of a steady state concentration S_i to a parameter p_m by

$$R_m^i = \frac{p_m}{S_i}\frac{\partial S_i}{\partial p_m}$$

In Copasi, sensitivities are calculated as the derivatives of model variables with respect to a defined list of model entities, e.g. parameter values. Sensitivities are also calculated as global quantities of the system, i.e. indirect effects are accounted for.
One should consider sensitivities **scaled by the concentrations for comparison**.
Although Copasi implies to compute **sensitivities for a time course**, it will only compute the sensitivities at the **last point** of the time course.

4.5 Stoichiometric Analysis

The model structure of metabolic networks contains important information about the system.
Elementary flux modes show, which parts of the system can independently function in steady state. To use Elementary flux modes, the **stoichiometric coefficients of the reactions have to be integer numbers**.
Mass Conservation is used to check whether any the net output of the system equals the net influx. For glycolytic models, the standard analysis has to be used with care because not every molecule described is necessarily of the same mass. For example, Fructose-1,6-diphosphate is cleaved into glyceraldehyde phosphate and dihydroxyacetone phosphate, producing 2 molecules from one without violating the factual mass balance.

4.6 Metabolic Control Analysis

Metabolic control analysis is a powerful theoretical framework to analyze the dependencies and regulation in biochemical models in steady state. Copasi can be used to compute three different measures of control:

ε-elasticities: given all other parameters remain constant, which effect will the change of a metabolite S_i have on the rate v_k of a specific reaction?

$$\varepsilon_i^k = \frac{S_i}{v_k}\frac{\partial v_k}{\partial S_i}$$

flux control coefficients: given all other parameters remain constant, how will a small change in reaction rate v_k affect the flux J_j through reaction j?

$$C_k^j = \frac{v_k}{J_j}\frac{\partial J_j}{\partial v_k}$$

concentration control coefficients: given that all other parameters remain constant, how will a small change in reaction rate v_k influence the steady state concentration S_i?

$$C_k^i = \frac{v_k}{S_i}\frac{\partial S_i}{\partial v_k}$$

While elasticities are local properties, control coefficients are global quantities of the system: Even if a particular metabolite does not directly influence a certain reaction rate, its indirect effects are taken into account.

5 Exercises 1: A Simplistic Model for Cell Cycle Oscillations

5.1 Biology of the Cell Cycle

A series of well timed events is needed to advance cells through their life cycle and allow them to grow and divide. In *Saccharomyces cerevisiae* (aka yeast), a model organism for the eukaryotes, the cell cycle can be divided into distinct phases (see Figure 8, left) that are tightly controlled by the expression of specific cell cycle genes, the cyclins and cyclin-dependent kinases. Specific events take place in each phase and checkpoint mechanisms assure that everything is in best order before the cell commits irreversibly to the next cell cycle phase.

If the concentrations of the cyclins and kinases are plotted over time, an oscillatory behavior can be observed, which repeats the same pattern with a fixed frequency. We will see that even for the most simple system, dynamic dependencies are rather non-intuitive and, therefore, mathematical models are a powerful tool for dissecting the detailed functionality and timing of cell cycle regulation.

One of the most simple control systems for the cell cycle can be found in amphibian embryonic cells, where the accumulation of one cyclin compound is sufficient to trigger cell cycle progression. Goldbeter et al.[3] presented a first mathematical model with self-sustained periodic dynamics based on this system. It contains merely 3 agents:

> The cyclin **C** itself, which accumulates steadily during the growth of the cell and at the same time triggers the activating dephosphorylation of
>
> the cdc2 kinase, also known as M-phase promoting factor **M**. This protein triggers entry into cell division as well as the re-setting of the cell cycle by degradation of the cyclin. The degradation is facilitated by
>
> the protease **X** that needs the phosphorylation by cdc2 to become active.

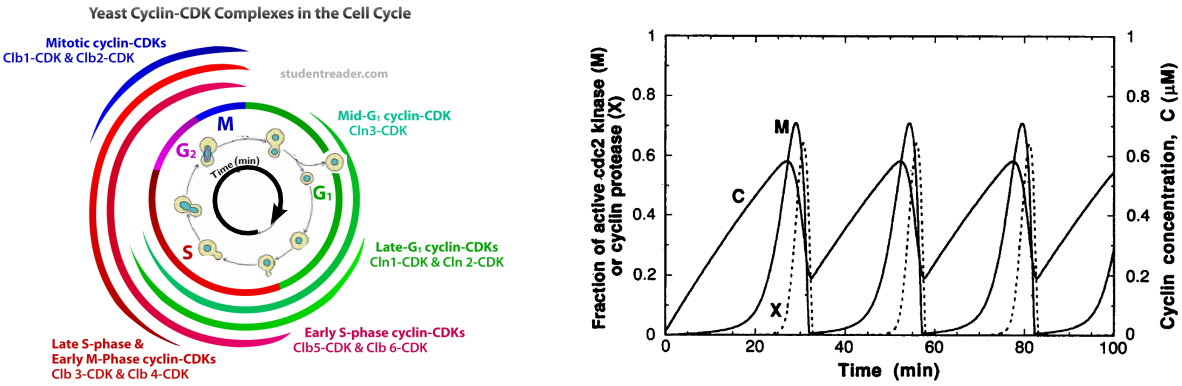

Figure 8: The cell cycle. *left:* schematic representation of the cell cycle phases in yeast along with the expressed cyclins at each stage; *right:* simulated cyclin dynamics of the Goldbeter model.

[3] A. Goldbeter. A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. Proceedings of the National Academy of Sciences of the United States of America, 88(20):9107–11, Oct. 1991

The sustained oscillations in the model originate from the **negative feedback loop** structure of the system along with activation thresholds for M and X. The thresholds are reached by a process called **zero-order ultrasensitivity**, e.g. the level of activated M depends in an almost switch-like manner on the concentration of the cyclin. Hence, in the sensitive region of the cyclin concentration, small changes in C will result in large changes in M, whereas outside of this region changing C concentrations will have close to no effect.
These two conditions are sufficient to achieve cyclic behavior of the cell cycle components, as can be seen in Figure 8 on the right side.

5.2 Goldbeter's cell cycle oscillator

Here, we want to apply our newly learned skills in Copasi to simulate a model of the cell cycle by Goldbeter *et al.*. The model consists of a cyclin (C), a kinase (M) and a protease (X). The kinase and the protease have two states – an active (M, X resp.) and an inactive one (M^+, X^+ resp.). The reaction scheme is given in Figure 9.

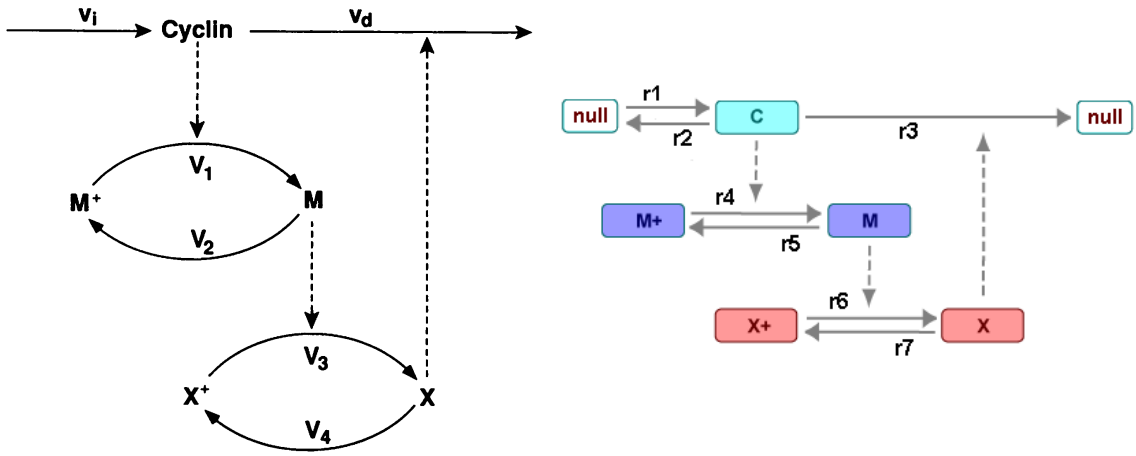

Figure 9: Reaction schemes of the Goldbeter model. *left:* from the original publication *right:* scheme for our implementation.

1. **Model implementation**

   Implement the reaction scheme according to Figure 9 in Copasi.
   Hint: Use mass action kinetics (Constant flux if not otherwise possible) to express the reactions *r1* and *r2*, and Michaelis-Menten type equations for the remaining reactions. You might have to define new functions.

   Parametrize the model arbitrarily and check if it can be simulated.

   As in the original publication, we can assume that $M+M^*=1$ and $X+X^*=1$. Modify the reaction rates accordingly to obtain the following set of differential equations:

   (a) $\frac{\mathrm{d}C}{\mathrm{d}t} = v_i - v_d X \frac{C}{K_d+C} - k_d C$
   (b) $\frac{\mathrm{d}M}{\mathrm{d}t} = V_1 \frac{1-M}{K_1+(1-M)} - V_2 \frac{M}{K_2+M}$

(c) $\frac{\mathrm{d}X}{\mathrm{d}t} = V_3 \frac{1-X}{L_3+(1-X)} - V_4 \frac{X}{K_4+X}$

with $V_1 = \frac{C}{K_c+C} V_{M1}$ and $V_3 = M V_{M3}$.

Extract the parameter values given in the publication to reproduce the original trajectories (Figure 8, right side).

2. **Model visualization** After you have obtained a sensible parameterization that leads to oscillations, visualize the oscillations:

   Generate a phase plane plot of C vs M.

   Plot the phase plane for different initial conditions.

   Conduct a parameter scan (task: Time course) for one of the model parameters. Compare your results to the results of a sensitivity analysis for that parameter.

   Optional: Use events to extract amplitude and phase of the oscillation of each species.

6 Exercises 2: Functional Analyses

6.1 Metabolic Control Analysis

1. Download the model BIOMD0000000253 from www.biomodels.org and open the associated article. Import the model into Copasi.

2. Run the *Metabolic Control Analysis* task and observe the matrix. What do positive and negative values mean in these three cases? Which flux and concentration control coefficients are strong? Can you relate this to the topographie of your model?

3. Modify the model to describe unguarded glycolysis, rerun MCA and compare your results (you can also start a 2nd Copasi instance to better compare).

4. Run the *Sensitivities* task (time course subtask). To which parameters are the steady state concentrations (or a certain steady state concentration) in particular sensitive? Chose one of them and observe how the system reacts when changing this parameter (*parameter scan*).

6.2 Bistability:

Find and visualize bistability in a simple MAPK model.

1. Download and import BIOMD0000000027.xml from www.biomodels.org and read the model.

2. Define an initial expression for the initial concentration for Mpp of the form $Mpp(0) = 500 - M(0) - Mp(0)$.

3. Set up a plot with $MAPKK(t)$ on the x-axis and $Mpp(t)$ on the y-axis using symbols instead of lines for plotting.

4. Run a *Parameter Scan* over the initial concentration of $MAPKK$ using steady state as subtask and run 100 intervals from 0 to 100. Add another scan to the task, namely over the initial concentration of M, using 20 intervals from 100 to 450. What does the plot tell you?

5. Add a third scan for the global quantity Km1, chose an interval from 50 to 500.

6. The range of the bistable region in fact depends on the ratio between Km1 and Km2. Modify the model to enable a scan over this ratio. You can visualize the result in a contour plot with $MAPKK$ on the x-axis, Km2 (with fixed Km1) on the y-axis and Mpp on the z-axis (color).

6.3 Populations and evolution

1. Implement a model in Copasi that considers the reproduction and degradation of some population pop. Reproduction requires consumption of $food$, which is able to reproduce itself.

2. Find a parameterization that gives classic predator-prey dynamics over a given time.

3. Introduce a second population pop_2 that either growths faster or dies slower. Find the parameter values for which pop_2 out-competes pop within a certain time range:

Create a global variable that is the difference between pop and pop_2

Use the *Parameter Scan* and an appropriate plot output to define the critical value where the behavior switches for each parameter.