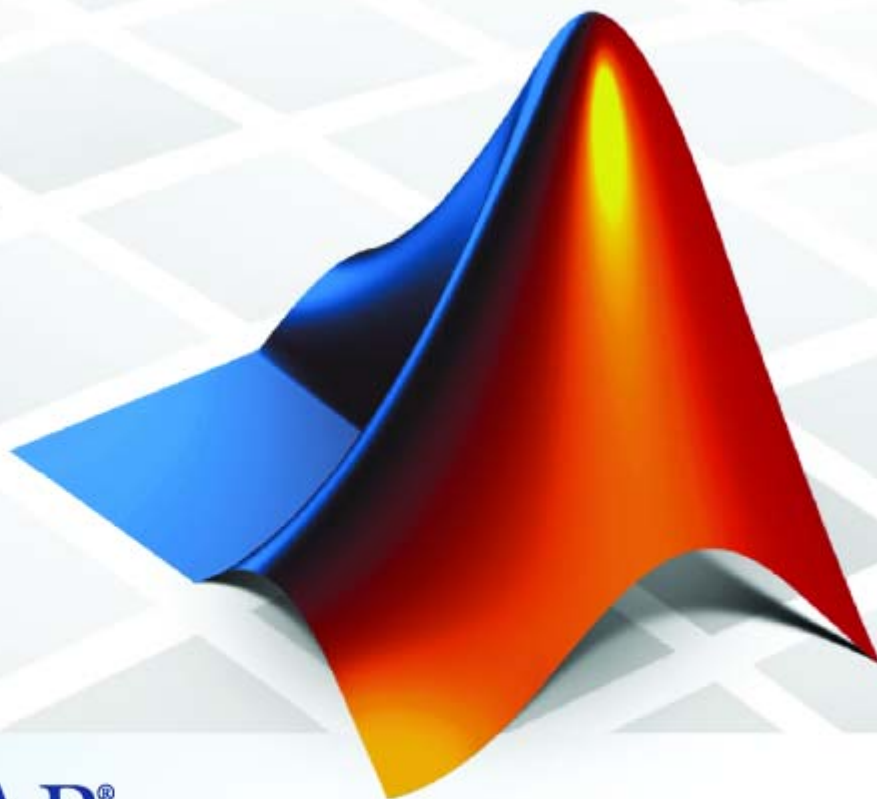


# SimBiology 2

## Model Reference



MATLAB®

## How to Contact The MathWorks



www.mathworks.com  
comp.soft-sys.matlab  
www.mathworks.com/contact\_TS.html

Web  
Newsgroup  
Technical Support



suggest@mathworks.com  
bugs@mathworks.com  
doc@mathworks.com  
service@mathworks.com  
info@mathworks.com

Product enhancement suggestions  
Bug reports  
Documentation error reports  
Order status, license renewals, passcodes  
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

*SimBiology Model Reference*

© COPYRIGHT 2005–2007 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

### Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, SimBiology, SimHydraulics, SimEvents, and xPC TargetBox are registered trademarks and The MathWorks, the L-shaped membrane logo, Embedded MATLAB, and PolySpace are trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

### Patents

The MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

**Revision History**

September 2005 Online only  
March 2006 Online only  
May 2006 Online only  
September 2006 Online only  
March 2007 Online only  
September 2007 Online only  
October 2007 Online only

New for Version 1.0 (Release 14SP3+)  
Updated for Version 1.0.1 (Release 2006a)  
Updated for Version 2.0 (Release 2006a+)  
Updated for Version 2.0.1 (Release 2006b)  
Rereleased for Version 2.1.1 (Release 2007a)  
Rereleased for Version 2.1.2 (Release 2007b)  
Updated for Version 2.2 (Release 2007b+)



## Minimal Cascade Model for a Mitotic Oscillator

# 1

<b>Goldbeter Model</b> .....	<b>1-2</b>
About the Goldbeter Model .....	<b>1-3</b>
Reaction Descriptions and Model Assumptions .....	<b>1-4</b>
Mathematical Model .....	<b>1-4</b>
<b>SimBiology Model with Rate Rules</b> .....	<b>1-6</b>
SimBiology Model with Rules .....	<b>1-6</b>
SimBiology Simulation with Rules .....	<b>1-9</b>
<b>SimBiology Model with Reactions</b> .....	<b>1-10</b>
Converting Differential Rate Equations to Reactions .....	<b>1-10</b>
Calculating Initial Values for Reactions .....	<b>1-12</b>
SimBiology Simulation with Reactions .....	<b>1-20</b>
<b>References</b> .....	<b>1-21</b>

## Model of the Yeast Heterotrimeric G Protein Cycle

# 2

<b>Objectives of This Section</b> .....	<b>2-3</b>
<b>Background on G Protein Cycles</b> .....	<b>2-4</b>
G Proteins .....	<b>2-4</b>
G Proteins and Pheromone Response .....	<b>2-5</b>
<b>Modeling a G Protein Cycle</b> .....	<b>2-6</b>
Reactions Overview .....	<b>2-6</b>
Assumptions, Experimental Data, and Units in the G Protein Model .....	<b>2-8</b>

<b>Building the G Protein Cycle Model in SimBiology</b> . . .	<b>2-11</b>
Goals of the Tutorial . . . . .	<b>2-11</b>
Opening the SimBiology Desktop . . . . .	<b>2-12</b>
Saving Your Work as a SimBiology Project File . . . . .	<b>2-13</b>
Adding a Reaction to the SimBiology Model . . . . .	<b>2-13</b>
Determining the Reaction Rate Equation . . . . .	<b>2-15</b>
Setting Compartment Name . . . . .	<b>2-18</b>
Setting Initial Amounts of Species . . . . .	<b>2-19</b>
<b>Completing the SimBiology Model</b> . . . . .	<b>2-21</b>
Reactions . . . . .	<b>2-21</b>
Alternative Ways to Build Reactions in the Desktop . . . . .	<b>2-22</b>
Parameters . . . . .	<b>2-24</b>
Species . . . . .	<b>2-25</b>
Creating a Rule for the G Protein Model . . . . .	<b>2-28</b>
Verifying the Model . . . . .	<b>2-29</b>
<b>Simulating the G Protein Cycle Model in SimBiology</b> . .	<b>2-30</b>
Setting Conditions Before Simulating the G Protein Model . . . . .	<b>2-30</b>
Simulation Results for the Model of the Wild-Type Strain . . . . .	<b>2-34</b>
<b>Building a Model for the Mutant Strain</b> . . . . .	<b>2-37</b>
Modeling the Mutant Strain . . . . .	<b>2-37</b>
Copying a Model . . . . .	<b>2-37</b>
Changing Model Settings . . . . .	<b>2-39</b>
<b>Simulating the G Protein Cycle Model of the Mutant Strain</b> . . . . .	<b>2-41</b>
Creating a Custom Plot for Simulation Results . . . . .	<b>2-41</b>
Plotting Using a Custom Plot . . . . .	<b>2-42</b>
Simulation Results for the Model of the Mutant Strain . . .	<b>2-43</b>
<b>Plotting Species from Two Different Data Sets</b> . . . . .	<b>2-45</b>
Procedures Described in this Section . . . . .	<b>2-45</b>
Plotting the Active G Protein Fraction from the Model of the Wild-Type Strain . . . . .	<b>2-45</b>
Plotting the Active G Protein Fraction from the Model of the Mutant Strain . . . . .	<b>2-46</b>

<b>Plotting Experimental Data with Simulation Data</b> . . . .	<b>2-48</b>
About the Experimental Data . . . . .	<b>2-48</b>
Creating a Custom Plot for Experimental Data . . . . .	<b>2-48</b>
Plotting the Data . . . . .	<b>2-50</b>
 <b>References</b> . . . . .	 <b>2-52</b>

## M-Phase Control in *Xenopus* Oocyte Extracts

# 3

<b>M-Phase Control Model</b> . . . . .	<b>3-2</b>
Synthesis Reactions . . . . .	<b>3-2</b>
Regulation Reactions with Active MPF . . . . .	<b>3-2</b>
 <b>M-Phase Control Equations</b> . . . . .	 <b>3-4</b>
About the Rate Equations Described in this Example . . . .	<b>3-4</b>
Converting Differential Equations to Reactions . . . . .	<b>3-5</b>
Equation 1 Cyclin B . . . . .	<b>3-6</b>
Equation 2 M-Phase Promoting Factor (MPF) . . . . .	<b>3-6</b>
Equation 3 Inhibited M-Phase Promoting Factor (pMPF) . . . . .	<b>3-7</b>
Equation 4 Inhibited and Activated M-Phase Promoting Factor (pMPFp) . . . . .	<b>3-8</b>
Equation 5 Activated M-Phase Promoting Factor (MPFp) . . . . .	<b>3-8</b>
Equation 11 Cell Division Control 25 (Cdc25) . . . . .	<b>3-9</b>
Equation 12 Wee1 Activation/Deactivation . . . . .	<b>3-10</b>
Equation 13 Intermediate Enzyme (IE) Activation/Deactivation . . . . .	<b>3-10</b>
Equation 14 APC Activation/Deactivation . . . . .	<b>3-11</b>
Equation 17 Rate Parameter K2 . . . . .	<b>3-11</b>
Equation 18 Rate Parameter Kcdc25 . . . . .	<b>3-12</b>
Equation 19 Rate Parameter Kwee1 . . . . .	<b>3-12</b>
 <b>SimBiology Model with Rate and Algebraic Rules</b> . . . .	 <b>3-14</b>
Overview . . . . .	<b>3-14</b>
Writing Differential Rate Equations as Rate Rules . . . . .	<b>3-15</b>
Species . . . . .	<b>3-15</b>
Parameters . . . . .	<b>3-16</b>

Rate Rule 1 Cyclin B (CycB) .....	3-17
Rate Rule 2 M-Phase Promoting Factor (MPF) .....	3-18
Rate Rule 3 Inhibited M-Phase Promoting Factor (pMPF) .....	3-18
Rate Rule 4 Activated but Inhibited M-Phase Promoting Factor (pMPFp) .....	3-19
Rate Rule 5 Activated M-Phase Promoting Factor (MPFp) .....	3-19
Rate Rule 11 Activated Cdc25 (Cdc25p) .....	3-19
Rate Rule 12 Inhibited Wee1 (Wee1p) .....	3-19
Rate Rule 13 Activated Intermediate Enzyme (IEp) .....	3-19
Rate Rule 14 Activated APC (APCa) .....	3-19
Algebraic Rule 17 Rate Parameter K2 .....	3-20
Algebraic Rule 18 Rate Parameter Kcdc25 .....	3-20
Algebraic Rule 19 Rate Parameter Kwee1 .....	3-20
Simulation of the M-Phase Control Model with Rules .....	3-20

### **SimBiology Model with Reactions and Algebraic**

<b>Rules</b> .....	3-22
Overview .....	3-23
Reaction 1 Synthesis of Cyclin B .....	3-23
Reaction 2 Degradation of Cyclin B .....	3-24
Reaction 3 Dimerization of Cyclin B with Cdc2 Kinase ...	3-25
Reaction 4 Degradation of Cyclin B on MPF .....	3-26
Reaction 5 Deactivation of Active MPF .....	3-27
Reaction 6 Activation of MPF .....	3-28
Reaction 7 Remove Inhibiting Phosphate from Inhibited MPF .....	3-29
Reaction 8 Inhibition of MPF by Phosphorylation .....	3-30
Reaction 11 Degradation of Cyclin B on Inhibited MPF ...	3-32
Reaction 12 Deactivation of MPF to Inhibited MPF .....	3-32
Reaction 13 Activation of Inhibited MPF .....	3-32
Reaction 15 Degradation of Cyclin B on Active but Inhibited MPF .....	3-33
Reaction 16 Inhibit MPF by Phosphorylation .....	3-33
Reaction 17 Remove Inhibiting Phosphate from Activated MPF .....	3-34
Reaction 19 Degradation of Cyclin B on Activated MPF ..	3-34
Reaction 36 Activation of Cdc25 by Activated MPF .....	3-34
Reaction 37 Deactivation of Cdc25 .....	3-35
Reaction 38 Deactivation of Wee1 by Active MPF .....	3-35
Reaction 39 Activation of Wee1 .....	3-35
Reaction 40 Activation of Intermediate Enzyme by Active MPF .....	3-36



Reaction 41 Deactivation of IE .....	<b>3-36</b>
Reaction 42 APC Activation by IE <sub>p</sub> .....	<b>3-36</b>
Reaction 43 APC Deactivation .....	<b>3-36</b>
Block Diagram of the M-Phase Control Model with Reactions .....	<b>3-37</b>
Simulation of the M-Phase Control Model with Reactions .....	<b>3-39</b>
<b>References</b> .....	<b>3-40</b>

## Index





# Minimal Cascade Model for a Mitotic Oscillator

---

Albert Goldbeter modified a model with enzyme cascades (Goldbeter and Koshland 1981) to fit cell cycle data from studies with embryonic cells (Goldbeter 1991). He used this model to demonstrate thresholds with enzyme cascades and periodic behavior caused by negative feedback.

There are two SimBiology model variations using Goldbeter's model. The first model uses the differential rate equations directly from Goldbeter's paper. The second model is built with reactions using Henri-Michaelis-Menten kinetics.

Goldbeter Model (p. 1-2)

Description and graphical representation of a simple enzyme cascade with a feedback loop

SimBiology Model with Rate Rules (p. 1-6)

Enter differential rate equations directly from the literature

SimBiology Model with Reactions (p. 1-10)

Convert differential rate equations to reactions and reaction rates

References (p. 1-21)

Literature references and Web links for additional information

## Goldbeter Model

In this section...
“About the Goldbeter Model” on page 1-3
“Reaction Descriptions and Model Assumptions” on page 1-4
“Mathematical Model” on page 1-4

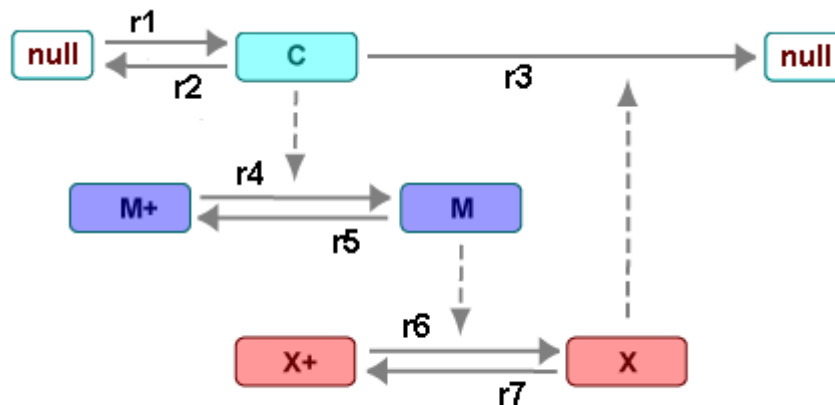
## About the Goldbeter Model

Albert Goldbeter created a simple cell division model from studies with embryonic cells (Goldbeter 1991). This model demonstrates thresholds with enzyme cascades and periodic behavior caused by negative feedback.

There are six species in Goldbeter's minimal mitotic oscillator model (Goldbeter 1991).

- C — Cyclin. The periodic behavior of cyclin activates and deactivates an enzyme cascade.
- M+, M — Inactive (phosphorylated) and active forms of cdc2 kinase. Kinases catalyze the addition of phosphate groups onto amino acid residues.
- X+, X — Inactive and active (phosphorylated) forms of a cyclin protease. Proteases degrade proteins by breaking peptide bonds.

The reactions are labeled r1 to r7 on the diagram



This model shows

- How thresholds with cdc2 kinase activation ( $M^+ \rightarrow M$ ) and protease activation ( $X^+ \rightarrow X$ ) can occur as the result of covalent modification (for example, phosphorylation or dephosphorylation), but without the need for positive feedback.

- How periodic behavior with cdc2 kinase activation can occur with negative feedback and the time delay associated with activation/deactivation enzyme cascades.

## Reaction Descriptions and Model Assumptions

The following list describes each of the reactions in Goldbeter's minimal mitotic oscillator with some of the simplifying assumptions. For a more detailed explanation of the model, see Goldbeter 1991.

- Cyclin (C) is synthesized at a constant rate ( $r_1$ ) and degraded at a constant rate ( $r_2$ ).
- Cyclin (C) does not complex with cdc2 kinase (M).
- Cyclin (C) activates cdc2 kinase ( $M^+ \rightarrow M$ ) by increasing the velocity of the phosphatase that activates the kinase. Inactive cdc2 kinase ( $M^+$ ) is activated by removing inhibiting phosphate groups ( $r_4$ ).
- The amount of deactivating kinase (not modeled) for the cdc2 kinase (M) is constant. Active cdc2 kinase (M) is deactivated by adding inhibiting phosphate group ( $r_5$ ).
- The activation of cyclin protease ( $X^+ \rightarrow X$ ) by the active Cdc2 kinase (M) is direct without other intervening cascades. Cyclin protease (X) is activated by adding phosphate groups ( $r_6$ ).
- The amount of deactivating phosphatase (not modeled) for the cyclin protease (X) is constant. Active cyclin protease (X) is deactivated by removing the activating phosphate groups ( $r_7$ ).
- The three species of interest are cyclin (C), active dephosphorylated cdc2 kinase (M), and active phosphorylated protease (X). The total amounts of (M +  $M^+$ ) and (X +  $X^+$ ) are constant.

## Mathematical Model

Goldbeter's minimal mitotic oscillator model is defined with three differential rate equations and two algebraic equations that define changing parameters in the rate equations

### Differential Rate Equation 1, Cyclin (C)

The following differential rate equation is from Goldbeter 1991 for cyclin (C).

$$\frac{dC}{dt} = v_i - v_d X \frac{C}{K_d + C} - k_d C$$

### Differential Rate Equation 2, Kinase (M)

The following differential rate equation is for cdc2 kinase (M). Notice that (1 - M) is the amount of inactive (phosphorylated) cdc2 kinase (M+).

$$\frac{dM}{dt} = V_1 \frac{(1 - M)}{K_1 + (1 - M)} - V_2 \frac{M}{K_2 + M}$$

$$V_1 = \frac{VM_1[C]}{K_c + [C]}$$

### Differential Rate Equation 3, Protease (X)

Differential rate equations for cyclin protease (X). Notice that (1 - X) is the amount of inactive (unphosphorylated) cyclin protease (X+).

$$\frac{dX}{dt} = V_3 \frac{(1 - X)}{K_3 + (1 - X)} - V_4 \frac{X}{K_4 + X}$$

$$V_3 = VM_3[M]$$

## SimBiology Model with Rate Rules

### In this section...

“SimBiology Model with Rules” on page 1-6

“SimBiology Simulation with Rules” on page 1-9

### SimBiology Model with Rules

In the literature, many biological models are defined using differential rate and algebraic equations. With SimBiology, you can enter the equations directly as SBML rules. The example in this section uses Goldbeter’s mitotic oscillator to illustrate this point.

Writing differential rate equations in an unambiguous format that a software program can understand is a fairly simple process.

- Use an asterisk to indicate multiplication. For example,  $k[a]$  is written  $k*a$ .
- Remove square brackets that indicate concentration from around species. The units associated with the species will indicate concentration (moles/liter) or amount (moles, molecules).

SimBiology uses square brackets around species and parameter name to allow names that are not valid MATLAB® variable names. For example, you could have a species named `glucose-6-phosphate dehydrogenase` but you need to add brackets around the name in reaction rate and rule equations.

- Use parentheses to clarify the order of evaluation for mathematical operations. For example, do not write a Henri-Michaelis-Menten rate as  $V_m*C/K_d + C$ , because  $V_m*C$  is divided by  $K_d$  before adding  $C$ , and then  $C$  is added to the result.

The following equation is the SimBiology rate rule for “Differential Rate Equation 1, Cyclin (C)” on page 1-4:

$$dC/dt = v_i - (v_d*X*C)/(K_d + C) - k_d*C$$

The following equations are the SimBiology rate and algebraic rules for “Differential Rate Equation 2, Kinase (M)” on page 1-5:



$$dM/dt = (V1*Mplus)/(K1 + Mplus) - (V2*M)/(K2 + M)$$

$$V1 = (VM1*C)/(Kc + C)$$

$$Mplus = Mt - M$$

The following equations are the rate and algebraic rules for “Differential Rate Equation 3, Protease (X)” on page 1-5:

$$dX/dt = (V3*Xplus)/(K3 + Xplus) - (V4*X)/(K4 + X)$$

$$V3 = VM3*M$$

$$Xplus = Xt - X$$

## Species

The following table is a list of species in the model with their initial amounts.

The two parameters V1 and V3 are in the species list. You could enter the parameters in the parameter table with the ConstantAmount check boxes cleared. Here, the parameters are modeled as species but without reactions.

Name	InitialAmount	ConstantAmount	BoundaryCondition
C	0.01	<input type="checkbox"/>	<input type="checkbox"/>
M	0.01	<input type="checkbox"/>	<input type="checkbox"/>
Mplus	0.99	<input type="checkbox"/>	<input type="checkbox"/>
Mt	1.0	<input type="checkbox"/>	<input type="checkbox"/>
X	0.01	<input type="checkbox"/>	<input type="checkbox"/>
Xplus	0.99	<input type="checkbox"/>	<input type="checkbox"/>
Xt	1.0	<input type="checkbox"/>	<input type="checkbox"/>
V1	0.0	<input type="checkbox"/>	<input type="checkbox"/>
V3	0.0	<input type="checkbox"/>	<input type="checkbox"/>

## Parameters

The following table is a list of parameters in the model with their initial values. The **ConstantValue** property is selected for all the parameters.

Name	Value	ConstantValue
vi	0.025	<input checked="" type="checkbox"/>
kd	0.01	<input checked="" type="checkbox"/>
vd	0.25	<input checked="" type="checkbox"/>
Kd	0.02	<input checked="" type="checkbox"/>
VM1	3.0	<input checked="" type="checkbox"/>
K1	0.0050	<input checked="" type="checkbox"/>
Kc	0.5	<input checked="" type="checkbox"/>
V2	1.5	<input checked="" type="checkbox"/>
K2	0.0050	<input checked="" type="checkbox"/>
VM3	1.0	<input checked="" type="checkbox"/>
K3	0.0050	<input checked="" type="checkbox"/>
V4	0.5	<input checked="" type="checkbox"/>
K4	0.0050	<input checked="" type="checkbox"/>

### Rules

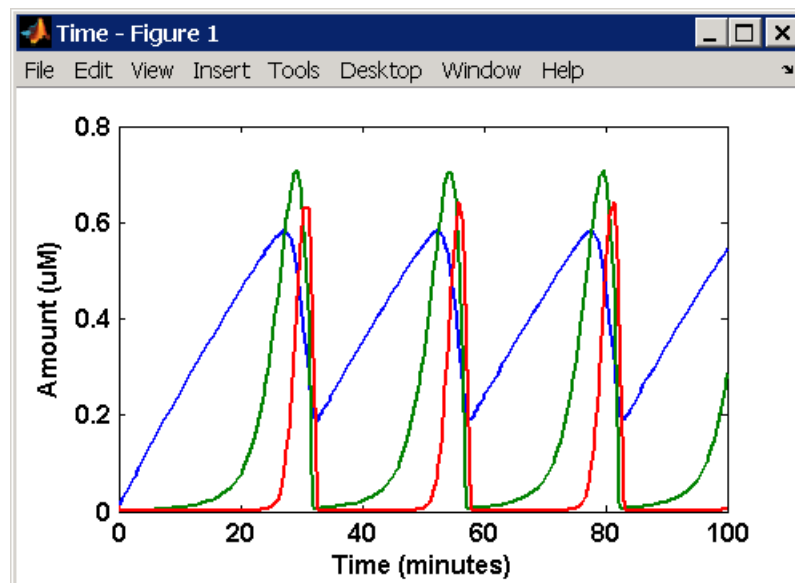
The active (M) and inactive (Mplus) forms of the kinase are assumed to be part of a conserved cycle with the total concentration (Mt) remaining constant during the simulation. You need only one differential rate equation with a mass balance algebraic equation to define the amounts of both species. Similarly, the active (X) and inactive (Xplus) forms of the protease are part of a second conserved cycle.

In the SimBiology desktop, you enter rate rules of the form  $\frac{dX}{dt} = \text{Expression}$  as  $X = \text{Expression}$ , and algebraic rules of the form " $X = \text{Expression}$ ", where X is the independent variable, as  $\text{Expression} - X$ .

Name ▾	Rule	RuleType
Cyclin	$C = v_i - (v_d * X * C) / (K_d + C) - k_d * C$	rate ▾
Kinase	$M = (V_1 * M_{plus}) / (K_1 + M_{plus}) - (V_2 * M) / (K_2 + M)$	rate ▾
	$M_t - M_{plus} - M$	algebraic ▾
	$(V_{M1} * C) / (K_c + C) - V_1$	algebraic ▾
Protease	$X = (V_3 * X_{plus}) / (K_3 + X_{plus}) - (V_4 * X) / (K_4 + X)$	rate ▾
	$X_t - X_{plus} - X$	algebraic ▾
	$V_{M3} * M - V_3$	algebraic ▾

## SimBiology Simulation with Rules

This is a simulation of Goldbeter's minimal mitotic oscillator using differential rate and algebraic equations. Simulate with the ode15s solver and plot species C, M, and X. For a description of the model, see "SimBiology Model with Rules" on page 1-6.



## SimBiology Model with Reactions

### In this section...

“Converting Differential Rate Equations to Reactions” on page 1-10

“Calculating Initial Values for Reactions” on page 1-12

“SimBiology Simulation with Reactions ” on page 1-20

### Converting Differential Rate Equations to Reactions

In the literature, many models are defined with differential rate equations. With SimBiology, creating the differential equations from reactions is unnecessary, you can enter the reactions and let the software calculate the equations.

Some models are defined with differential rate equations, and you might need the reactions to be compatible with your model. Two rules you can use to convert differential rate equations to reactions are

- **For a positive term** — The species described by the equation is placed on the right as a product, and the species in the term are placed on the left as reactants.
- **For a negative term** — The species described by the equation is placed on the left as a product, and the species in the term are also placed on the left as reactants.

You need to determine the products using additional information, for example, a reaction diagram, a description of the model, or an understanding of a reaction. If a reaction is catalyzed by a kinase, then you can conclude that the product has one or more additional phosphate groups.

A simple first order reaction has differential rate equation  $dR/dt = +kr[P] - kf[R]$ . The negative term implies that the reaction is  $R \rightarrow ?$  with an unknown product. The positive term identifies the product and completes the reaction,  $R \leftrightarrow P$ .

### Reactions R1 to R3 from Equation E1

The differential rate equation 1 is repeated here for comparison with the reactions. See “Differential Rate Equation 1, Cyclin (C)” on page 1-4.

$$\frac{dC}{dt} = v_i - v_d X \frac{C}{K_d + C} - k_d C$$

The reaction and reaction rate equations for SimBiology from the differential rate equation E1 are given below.

```
r1      reaction: null -> C
        reaction rate: v_i

r2      reaction: C -> null
        reaction rate: k_d*C

r3      reaction: C -> null
        reaction rate: (v_d*X*C)/(K_d + C)
```

### Reactions R4 and R5 from Equation E2

The differential rate equation 2 and algebraic equation 2 are repeated here for comparison with the reactions. See “Differential Rate Equation 2, Kinase (M)” on page 1-5.

$$\frac{dM}{dt} = V_1 \frac{(1-M)}{K_1+(1-M)} - V_2 \frac{M}{K_2+M}$$

$$V_1 = \frac{VM_1[C]}{K_c + [C]}$$

The reaction and reaction rate equations for SimBiology from the differential rate equation E2 are given below.

```
r4      reaction: Mplus -> M
        reaction rate: V1*Mplus/(K1 + Mplus)
        algebraic rule: V1 = VM1*C/(Kc + C)

r5      reaction: M -> Mplus
```

reaction rate:  $V2 \cdot M / (K2 + M)$

### Reactions R6 to R7 from Equation E3

The differential rate equation for equation 3 and algebraic equation 3 is repeated here for comparison with the reactions.

$$\frac{dX}{dt} = V3 \frac{(1-X)}{K3+(1-X)} - V4 \frac{X}{K4+X}$$

$$V3 = VM3 \cdot [M]$$

The reaction and reaction rate equations for SimBiology from the differential rate equation E3 are given below:

r6            reaction: Xplus -> X  
               reaction rate:  $V3 \cdot Xplus / (K3 + Xplus)$   
               algebraic rule:  $V3 = VM3 \cdot M$

r7            reaction: X -> Xplus  
               reaction rate:  $V4 \cdot X / (K4 + X)$

### Calculating Initial Values for Reactions

After you converted the differential rate equations to the reactions and reaction rate equations, you can start to fill in initial values for the species (reactants and products) and parameters.

The initial values for parameters and amounts for species are listed with four different units in the same dimension:

- A — Original units in the Goldbeter 1991 paper.
- B — Units of concentration with time converted to second. When converting a to b, use 1 minute = 60 second for parameters.

$$\frac{X \text{ uM}}{\text{minute}} \times \frac{1e-6 \text{ mole/liter}}{1 \text{ uM}} \times \frac{1 \text{ minute}}{60 \text{ second}} = \frac{Y \text{ mole}}{\text{liter} \cdot \text{second}}$$

- C — Units of amount as moles. When converting concentration to moles, use a cell volume of 1e-12 liter and assume that volume does not change.

$$\frac{Y \text{ mole}}{\text{liter} \cdot \text{second}} \times \frac{1 \text{e-12 liter}}{1} = \frac{Z \text{ mole}}{\text{second}}$$

- D — Units of amount as molecules. When converting amount as moles to molecules, use  $6.022 \times 10^{23} \text{ molecules} = 1 \text{ mole}$ .

$$\frac{Z \text{ mole}}{\text{second}} \times \frac{6.022 \times 10^{23} \text{ molecule}}{1 \text{ mole}} = \frac{N \text{ molecules}}{\text{second}}$$

With dimensional analysis on and unit conversion off, select all of the units for one letter. For example, select all of the As. If dimensional analysis and unit conversion are on, you can mix and match letters and get the same answer.

### Reaction 1 Cyclin Synthesis

R1		Value	Units
reaction	null -> C	----	----
reaction rate	vi	----	A. uM/minute
		----	B. mole/(liter*second)
		----	C. mole/second
		----	D. molecule/second
parameters	vi	0.025	A. uM/minute
		4.167e-10	B. mole/(liter*second)
		4.167e-22	C. mole/second
		205	D. molecule/second
species	C	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule

### Reaction 2 Cyclin Undifferentiated Degradation

R2		Value	Units
reaction	C -> null	----	----
reaction rate	kd*C	----	A. uM/minute
		----	B. mole/(liter*second)
		----	C. mole/second
		----	D. molecule/second
parameters	kd	0.010	A. 1/minute
		1.6667e-4	B, C, D. 1/second
species	C	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule

### Reaction 3 Cyclin Protease Degradation

R3		Value	Units
reaction	C -> null	----	----
reaction rate	$(vd * X * C) / (Kd + C)$	----	A. uM/minute
		----	B. mole/(liter*second)
		----	C. mole/second
		----	D. molecule/second
parameter	vd	0.25	A. 1/minute
		0.0042	B, C, D. 1/second
parameter	Kd	0.02	A. uM
		2.0e-8	B. mole/liter
		2.0e-020	C. mole
		12044	D. molecule



<b>R3</b>		<b>Value</b>	<b>Units</b>
species	C (substrate)	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule
species	X (enzyme)	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule

#### Reaction 4 Cdc2 Kinase Activation

<b>R4</b>		<b>Value</b>	<b>Units</b>
reaction	Mplus -> M	----	----
reaction rate	$(V1 * Mplus) / (K1 + Mplus)$	----	A. uM/minute
		----	B. mole/(liter*second)
		----	C. mole/second
		----	D. molecule/second
algebraic rule	$V1 = (VM1 * C) / (Kc + C)$	----	
parameter	V1 (variable by rule)	0.00	A. uM/minute
			B. mole/(liter*second)
			C. mole/second
			D. molecule/second
parameter	VM1	3.0	A. uM/minute
		5.0e-8	B. mole/(liter*second)
		5.0000e-020	C. mole/second

<b>R4</b>		<b>Value</b>	<b>Units</b>
		30110	D. molecule/second
parameter	Kc	0.5	A. uM
		5.0000e-7	B. mole/liter
		5.0e-19	C. mole
		3.011e+5	D. molecule
parameter	K1	0.005	A. uM
		5e-9	B. mole/liter
		5e-21	C. mole
		3.011e+3	D. molecule
species	Mplus (inactive substrate)	0.99	A. uM
		9.9e-7	B. mole/liter
		9.9e-19	C. mole
		5.962e+5	D. molecule
species	M (active product)	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule
species	C	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule

**Reaction 5 Cdc2 Kinase Deactivation**

<b>R5</b>		<b>Value</b>	<b>Units</b>
reaction	M -> M_plus	----	----
reaction rate	$(V2 * M) / (K2 + M)$	----	A. uM/minute B. (mole/liter-second) C. mole/second D. molecule/second
parameter	V2	1.5 2.5000e-008 2.5000e-020 15055	A. uM/minute B. mole/liter-second C. mole/second D. molecule/second
parameter	K2	0.005 5.0000e-009 5.0000e-021 3011 1.0e-20	A. uM B. mole/liter C. mole D. molecule
species	Mplus (inactive)	0.99 9.9e-7 9.9e-19 5.962e+5	A. uM B. mole/liter C. mole D. molecule
species	M (active)	0.01 1e-8 1.0e-20 6.022e+3	A. uM B. mole/liter C. mole D. molecule

### R6 Protease Activation

R6		Value	Units
reaction	Xplus -> X	----	----
reaction rate	$(V3 * Xplus) / (K3 + Xplus)$	----	A. uM/minute
		----	B. mole/(liter*second)
		----	C. mole/second
		----	D. molecule/second
algebraic rule	$V3 = VM3 * M$	----	
parameter	V3 (variable by rule)		A. uM/minute
			B. mole/liter-second
			C. mole/second
			D. molecule/second
parameter	VM3	1.0	A, 1/minute
		0.0167	B, C, D. 1/second
parameter	K3	0.005	A. uM
		5e-9	B. mole/liter
		5e-21	C. mole
		3.011e+3	D. molecule
species	Xplus (inactive substrate)	0.99	A. uM
		9.9e-7	B. mole/liter
		9.9e-19	C. mole
		5.962e+5	D. molecule
species	X (active product)	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole

<b>R6</b>		<b>Value</b>	<b>Units</b>
		6.022e+3	D. molecule
species	M (enzyme)	0.01	A. uM
		1e-8	B. mole/liter
		1.0e-20	C. mole
		6.022e+3	D. molecule

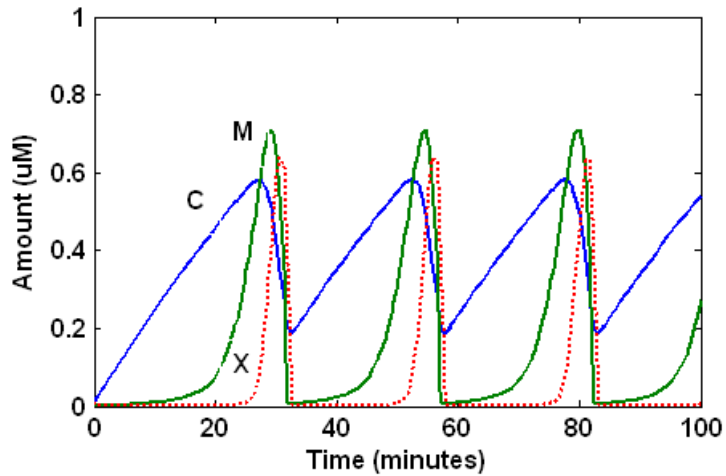
### Reaction 7 Protease Deactivation

<b>R7</b>		<b>Value</b>	<b>Units</b>
reaction	$X \rightarrow X_{plus}$	----	----
reaction rate	$(V4 * X) / (K4 + X)$	----	A. uM/minute
		----	B. mole/(liter*second)
		----	C. mole/second
		----	D. molecule/second
parameter	V4	0.5	A. uM/minute
		8.3333e-009	B. mole/(liter*second)
		8.3333e-021	C. mole/second
		5.0183e+003	D. molecule/second
parameter	K4	0.005	A. uM
		5e-9	B. mole/liter
		5e-21	C. mole
		3011	D. molecule
species	Xplus (inactive)	0.99	A. uM
		9.9e-7	B. mole/liter
		9.9e-19	C. mole
		5.962e+5	D. molecule
species	X (active)	0.01	A. uM

R7	Value	Units
	1e-8	B. mole/liter
	1.0e-20	C. mole
	6.022e+3	D. molecule

### SimBiology Simulation with Reactions

This is a simulation of Goldbeter's minimal mitotic oscillator with rate and algebraic equations. Simulate with the ode15s solver and plot species C, M, and X. For a description of the model, see "SimBiology Model with Reactions" on page 1-10.



## References

- [1] Goldbeter A. (1991), "A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase," *Proceedings of the National Academy of Sciences USA*, 88:9107-9111.
- [2] Goldbeter A., Koshland D. (1981), "An amplified sensitivity arising from covalent modification in biological systems," *Proceedings of the National Academy of Sciences USA*, 78:6840-6844.
- [3] Goldbeter A., Koshland D. (1984), "Ultrasensitivity in biochemical systems controlled by covalent modification," *The Journal of Biological Chemistry*, 259:14441-14447.
- [4] Goldbeter A., home page on the Web,  
<http://www.ulb.ac.be/sciences/utc/GOLDBETER/agoldbet.html>
- [5] Murray A.W., Kirschner M.W. (1989), "Cyclin synthesis drives the early embryonic cell cycle," *Nature*, 339:275-280.





# Model of the Yeast Heterotrimeric G Protein Cycle

---

Objectives of This Section (p. 2-3)	What you will learn in this section
Background on G Protein Cycles (p. 2-4)	Brief overview of G proteins and the G protein cycle in the yeast pheromone response pathway
Modeling a G Protein Cycle (p. 2-6)	Overview of reactions and assumptions
Building the G Protein Cycle Model in SimBiology (p. 2-11)	Describes the goals of this tutorial and explains how to build the model in SimBiology
Completing the SimBiology Model (p. 2-21)	Write the rest of the reactions, determine the reaction rate equations, assign the species initial amounts, and create a rule
Simulating the G Protein Cycle Model in SimBiology (p. 2-30)	Determine the settings for simulation and visualize the simulation results
Building a Model for the Mutant Strain (p. 2-37)	Create a model for the mutant strain ( <i>sst2Δ</i> ) by copying and changing the existing model
Simulating the G Protein Cycle Model of the Mutant Strain (p. 2-41)	Plot the simulation results for the mutant strain model using a custom plot

Plotting Species from Two Different Data Sets (p. 2-45)

Plot species data from two different simulation runs

Plotting Experimental Data with Simulation Data (p. 2-48)

Store experimental data and use custom plotting to plot experimental and simulation data together

References (p. 2-52)

Reference material used in this tutorial

## Objectives of This Section

SimBiology lets you build a model using a conceptual framework of biochemical reactions that describe a biological process. You can plot experimental data on top of your model's simulation results to investigate the validity of your model, make predictions based on the model, and test your hypotheses.

Using concepts and data from the published work of Yi and colleagues [Yi et al. 2003], this tutorial shows you how to

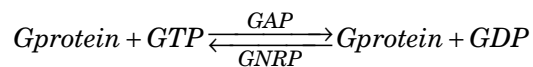
- 1** Build a model using the SimBiology graphical user interface (GUI).
- 2** Copy the model and change settings to create a different model.
- 3** Simulate and save the data from the two models.
- 4** Compare the two simulations.
- 5** Compare the simulation results with the experimental data.

## Background on G Protein Cycles

In this section...
“G Proteins” on page 2-4
“G Proteins and Pheromone Response” on page 2-5

### G Proteins

Cells rely on signal transduction systems to communicate with each other and to regulate cellular processes. G proteins are GTP-binding proteins that are involved in the regulation of many cellular processes. There are two known classes of G proteins: the monomeric G proteins (one GTPase), and the heterotrimeric G proteins (three different monomers). The G proteins usually facilitate a step requiring energy. This energy is supplied by the hydrolysis of GTP by a GTPase activating protein (GAP). The exchange of GDP for GTP is catalyzed by a guanine nucleotide releasing protein (GNRP) [Alberts et al. (1994)].



G protein coupled receptors (GPCRs) are the targets of many pharmaceutical agents. Some estimates suggest that 40 to 50% of currently marketed drugs target GPCRs and that 40% of current drug discovery focus is on GPCR targets. Some examples include those for reducing stomach acid (ranitidine which targets histamine H2 receptor), migraine (sumatriptan, which targets a serotonin receptor subtype), schizophrenia (olanzapine, which targets serotonin and dopamine receptors), allergies (desloratadine, which targets histamine receptors). One approach in pharmaceutical research is to model signaling pathways to analyze and predict both downstream effects and effects in related pathways. This tutorial examines model building and analysis of the G protein cycle in the yeast pheromone response pathway using SimBiology.

## **G Proteins and Pheromone Response**

In the yeast *Saccharomyces cerevisiae*, G protein signaling in pheromone response is a well characterized signal transduction pathway. The pheromone secreted by *alpha* cells activates the G protein coupled  $\alpha$ -factor receptor (Ste2p) in *a* cells which results in a variety of cell responses including cell-cycle arrest and synthesis of new proteins. The authors of the study performed a quantitative analysis of this cycle, compared the regulation of G protein activation in wild-type yeast haploid *a* cells with cells containing mutations that confer supersensitivity to  $\alpha$ -factor. They analyzed the data in the context of cell-cycle arrest and pheromone-induced transcriptional activation and developed a mathematical model of the G protein cycle that they used to estimate rates of activation and deactivation of active G protein in the cell.

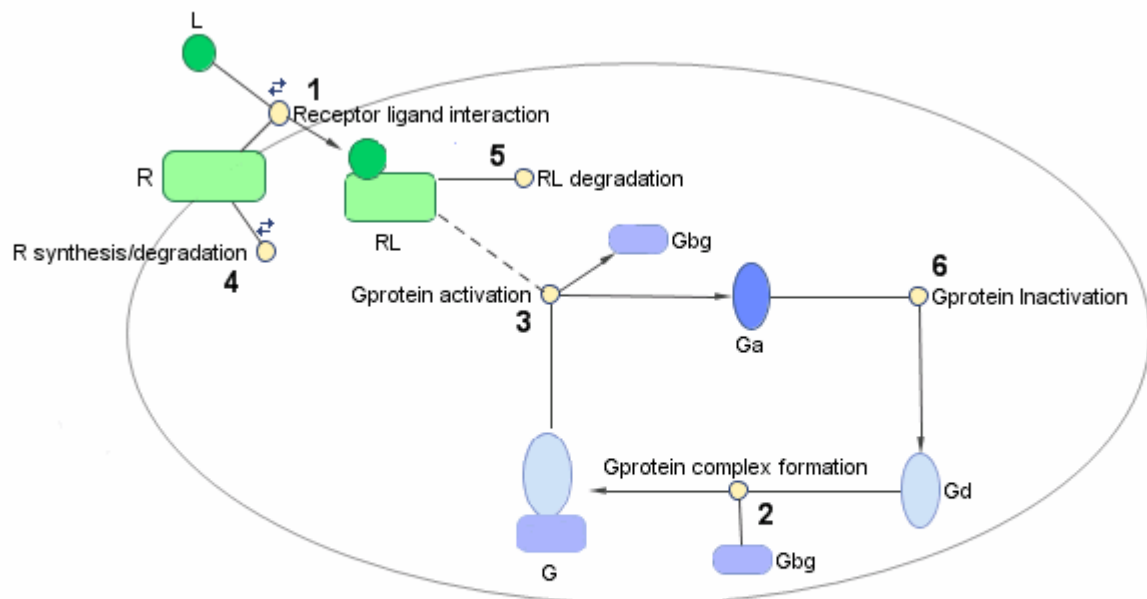
## Modeling a G Protein Cycle

In this section...
“Reactions Overview” on page 2-6
“Assumptions, Experimental Data, and Units in the G Protein Model” on page 2-8

### Reactions Overview

Systems biologists represent biological pathways and processes as reactions with reaction rates and treat the components of these pathways as individual species.

The G protein cycle in the yeast pheromone-response pathway can be condensed into a set of biochemical reactions. These reactions are complex formation, transformation, or disassociation reactions that Yi and colleagues [Yi et al.2003] use to simplify and describe the system. In this example,  $\alpha$ -factor,  $\alpha$ -factor receptor, and the G protein subunits are all treated as species participating in reactions. The system can be graphically represented as follows:



Graphical Representation of the G protein cycle in yeast pheromone response. The numbers represent reaction numbers referenced in the text. L = Ligand (alpha factor), R = alpha-factor receptor, Gbg = free levels of G-beta:G-gamma complex, Ga = active G-alpha-GTP, Gd = inactive G-alpha-GDP, G = inactive Gbg:Gd complex.

The following table shows you the reactions used to model the G protein cycle and the corresponding rate constants (rate parameters) for each reaction. For reversible reactions, the forward rate parameter is listed first.

No.	Name	Reaction	Rate Parameters
1	Receptor-Ligand Interaction	$L + R \leftrightarrow RL$	$k_{RLm}, k_{RL}$
2	Heterotrimeric G protein formation	$Gd + Gbg \rightarrow G$	$k_{G1}$
3	G protein activation	$RL + G \rightarrow Ga + Gbg + RL$	$k_{Ga}$

No.	Name	Reaction	Rate Parameters
4	Receptor synthesis and degradation	$R \leftrightarrow \text{null}$	kRdo, kRs
5	Receptor-Ligand degradation	$RL \rightarrow \text{null}$	kRD1
6	G protein inactivation	$Ga \rightarrow Gd$	kGd

Note that in reaction 3 (G protein activation), RL appears on both sides of the reaction. This is because RL is treated as a modifier or catalyst, and the model assumes that there is no synthesis or consumption of RL in this reaction.

The authors use a set of ordinary differential equations (ODEs) to describe the system. In SimBiology, you can represent the biological pathway as a system of biochemical reactions and the software creates the ODEs for you. Alternatively, if you have a set of ODEs that describe your system you can enter these as rate rules in SimBiology. For an example of modeling using rate rules, see “SimBiology Model with Rate Rules” on page 1-6.

## Assumptions, Experimental Data, and Units in the G Protein Model

The authors have obtained experimental data either through their own measurements or through published literature. As with any other model, the G protein cycle model simplifies the biological process while also trying to reconcile the experimental data. There are the following points to consider:

- Reaction 2 - Binding and formation of the heterotrimeric G protein complex is treated as a single step reaction.
- Reaction 3 - Activation of G protein is modeled as a single step. Guanine nucleotide exchange factors (GEFs) are not modeled.
- Reactions 3 and 6 - The parameters for the rate of G protein activation and deactivation (kGa and kGd) have been estimated based on the dose response curves in the reference paper. The SimBiology model being built in this tutorial directly uses those values.



- Reactions 4 and 5 - Receptor synthesis and degradation are handled purely as two simple reaction steps.
- Reaction 6 - Deactivation of G protein by the regulator of G protein signaling (RGS) protein Sst2p is modeled as a single step. Sst2p is not modeled.

The reaction is modeled with an estimated reaction rate of  $0.11 \text{ s}^{-1}$  in the Sst2p containing wild-type strain. The uncatalyzed reaction rate is estimated to be  $0.004 \text{ s}^{-1}$  in a strain with a deletion of SST2 (*sst2Δ*, mutant strain).

- Free GDP, GTP, and Pi are not included in the model.

This tutorial shows you how to plot the experimental data over the simulation plot of the active G protein fraction. You can estimate the values of the experimental data of interest for this example from the coordinates of the plots found in Figure 5 of the reference paper [Yi et al. 2003]. The following values were obtained by comparing the coordinates of the standards with those of the unknowns in the figure:

Time	Fraction of Active $G\alpha$ (experimental)
0	0.00
10	0.35
30	0.40
60	0.36
110	0.39
210	0.33
300	0.24
450	0.17
600	0.20

---

**Note** The SimBiology **Dimensional Analysis** feature is not used in this tutorial. For this tutorial, the values of all species are converted to have the unit molecule, and all rate parameters are converted to have either the unit 1/second or the units 1/(molecule\*second), depending on whether the reaction is first or second order. You should leave the **InitialAmountUnits** box for species, and **ValueUnits** box for rate parameters empty for the models in this tutorial.

---

# Building the G Protein Cycle Model in SimBiology

## In this section...

“Goals of the Tutorial” on page 2-11

“Opening the SimBiology Desktop” on page 2-12

“Saving Your Work as a SimBiology Project File” on page 2-13

“Adding a Reaction to the SimBiology Model” on page 2-13

“Determining the Reaction Rate Equation” on page 2-15

“Setting Compartment Name” on page 2-18

“Setting Initial Amounts of Species” on page 2-19

## Goals of the Tutorial

This section shows you how to build the example yeast heterotrimeric G protein models using SimBiology’s graphical user interface (GUI) (SimBiology desktop). For an overview of the SimBiology desktop, click [here](#). The SimBiology desktop is also referred to as “the desktop” in the SimBiology documentation.

This section assumes that you are starting with an untitled Project and a default Untitled Model Session in the SimBiology desktop.

If you are running the `gprotein.sbproj` file that contains the models in this tutorial, you can refer to “Simulating the G Protein Cycle Model in SimBiology” on page 2-30 for the sections on the simulation and analysis of these models.

This example uses the yeast G protein cycle (Yi et al. 2003) to illustrate model building and analysis in SimBiology. The goals of this tutorial are the following:

- 1 Build a model of the wild-type strain (TMY101) that has the SST2 gene. This strain shows a catalyzed rate of deactivation of  $G\alpha$ .  $G\alpha$  is represented as Ga in the model.

- 2 Copy the model and change settings to create a different model for the mutant strain (*sst2* $\Delta$ , TMY111) that shows an uncatalyzed rate of G-Protein inactivation.
- 3 Simulate and save the data from the two models.
- 4 Compare the active G protein fractions in the two simulations.
- 5 Compare the simulation results for active G protein fractions with experimental data.

For additional help in each procedure, select **Help > SimBiology Desktop Help** for context-sensitive help.

### Opening the SimBiology Desktop

All of the procedures in this tutorial are performed in the SimBiology desktop. The desktop provides access to command-line functionality through a graphical user interface. You can open the desktop from the MATLAB Command Window.

- 1 To open the desktop, at the MATLAB command line, type

```
sbiodesktop
```

The **SimBiology Desktop** opens. Use the **Project Explorer** in the left pane to navigate.

- 2 Select **File > New Project**.
- 3 In the **New Project** dialog box, from the **Project Type** list select **Model** and click **OK**.

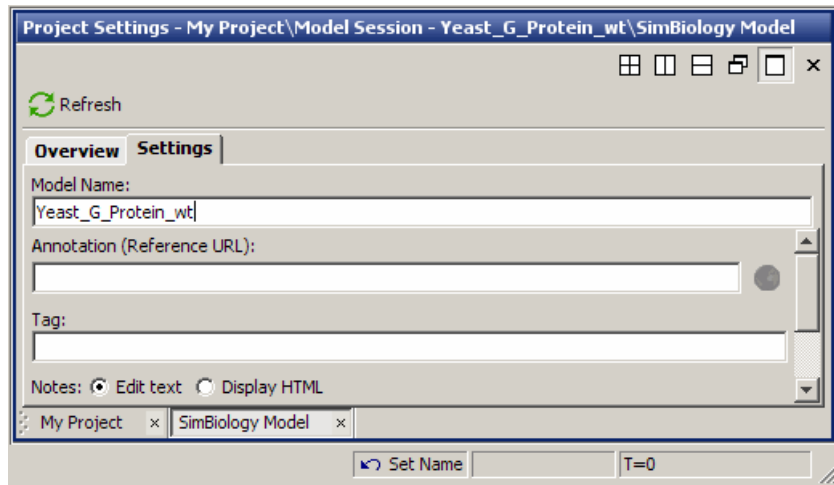
When you select an option, the **Project Description** pane shows you the items included in the project.

A new project, with an untitled model session, opens in the desktop.

- 4 In the **Project Explorer** double-click **SimBiology Model**, to open the **SimBiology Model** pane.
- 5 Click the **Settings** tab.

- 6** In the **Model Name** box, type the name for your model and press **Enter**.

Yeast\_G\_Protein\_wt



## Saving Your Work as a SimBiology Project File

Project (.sbproj) is the file format that SimBiology uses to save one or more model sessions. Projects lets you save custom settings, notes, and data associated with your models.

Save your work as a project now so that you can access this file later.

- 1** From the **File** menu, select **Save Project As**.

The Save SimBiology Project dialog box opens.

- 2** Browse to the folder in which you want to save your projects, enter a name for the project file, and then click **Save**.

yeast\_g\_protein\_cycle.sbproj

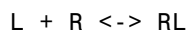
## Adding a Reaction to the SimBiology Model

The next steps show you how to add a reaction and determine the reaction rate equation for your model.

This example shows you the first reaction.

Name	Reaction	Rate Parameters
Receptor-Ligand Interaction	$L + R \leftrightarrow RL$	kRLm, kRL

- 1 In the **Project Explorer**, expand **Table View** and double-click **Reactions**.
- 2 In the **Reactions** pane, enter the reaction in the **Enter Reaction** box, and click **Add**.



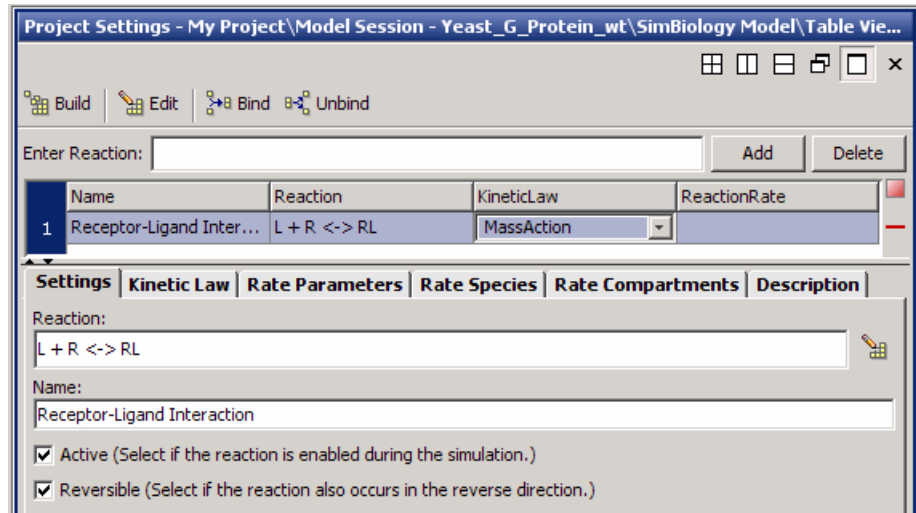
Notice that a colored indicator (red) appears to the right of the table. Move the pointer over the indicator for the reaction, to get more information about the reason for the indicator. In this case, the indicator shows that the reaction rate is as yet undefined. The next section in this tutorial shows how to define the reaction rate.

- 3 Double-click the **Name** box, and type the name for your reaction.

Receptor-Ligand Interaction

- 4 In the reaction table, from the **KineticLaw** list, select MassAction.

Your screen should now look similar to the figure below:



Notice the following in the **Settings** tab:

- Because this reaction is reversible, the **Reversible** check box is selected by default when you enter the reaction.
- All the reactions in this example model are included in the simulation and therefore have the **Active** check box selected. By default, the **Active** check box is selected.

---

**Note** Use spaces between the species and the characters in the reaction. If you have a reaction with different stoichiometry, for example  $2 A + B \leftrightarrow 3 AB$ , you must have a space between the stoichiometric coefficient and the species name for the reaction rate to be accurately determined. Otherwise, the coefficients are considered as part of the species name.

---

## Determining the Reaction Rate Equation

The desktop populates the reaction rate column after you specify the kinetic law and the rate parameters of the reaction.

To assign and configure the kinetic law and the rate parameters,

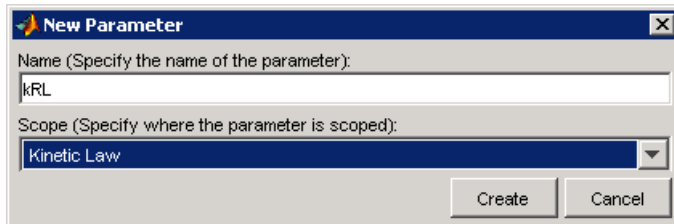
- 1 In the **Reactions** pane, click the **Kinetic Law** tab.
- 2 In the **Specify Corresponding Parameter Names** box, assign names to the parameters by clicking **New**. The New Parameter dialog box opens.
- 3 Specify the name of the parameter in the **Name** box. For **Forward Rate Parameter** enter kRL.
- 4 From the **Scope** list, select **Kinetic Law**. This applies the parameter locally, which means that this parameter is only available for use by this reaction.

If you set the scope of a parameter to Model, you can use this parameter in rules, other reactions, and events. All the parameters for this model are scoped to their respective kinetic laws.

### More About Scoping

*Scoping* lets you overload parameters; you can have parameters with the same name at kinetic law and at model level. Because the desktop searches up from kinetic law to model level, when a parameter is overloaded, a reaction uses the parameter and parameter value in its kinetic law, however rules and events use the parameter value at the model level.

A reaction uses a parameter at the model level, only if a corresponding parameter is not defined at the kinetic law level for the reaction.



- 5 Click **Create**.
- 6 Repeat steps 2 through 5 to create the **Reverse Rate Parameter**. Call the parameter kRLm.



Notice that the desktop updates the **ReactionRate** column and that the indicator is now green showing that this reaction is now completely defined.

The screenshot shows the SimBiology desktop interface for configuring a reaction. At the top, there is an "Enter Reaction:" field and "Add" and "Delete" buttons. Below this is a table with the following data:

Name	Reaction	KineticLaw	ReactionRate
1 Receptor-Ligand Intera...	L + R <-> RL	MassAction	kRL*L*R - kRLm*RL

Below the table, the "Kinetic Law" tab is selected. The "KineticLaw" dropdown is set to "MassAction". The "KineticLaw Expression:" field contains "MassAction". The "ReactionRate:" checkbox is checked and green, and the "ReactionRate:" field contains "kRL\*L\*R - kRLm\*RL". The "Specify Corresponding Parameter Names:" section shows "Forward Rate Parameter:" set to "kRL" and "Reverse Rate Parameter:" set to "kRLm". The "KineticLaw Species (Species defined automatically for MassAction KineticLaw):" section lists "L", "R", and "RL".

At the bottom of the desktop, the status bar shows "Set ParameterVariableNames" and "T=0".

The desktop automatically selects the species when you select MassAction kinetic law.

For other kinetic laws, you should select the species to be included in the rate equation from the **Kinetic Law Species** list.

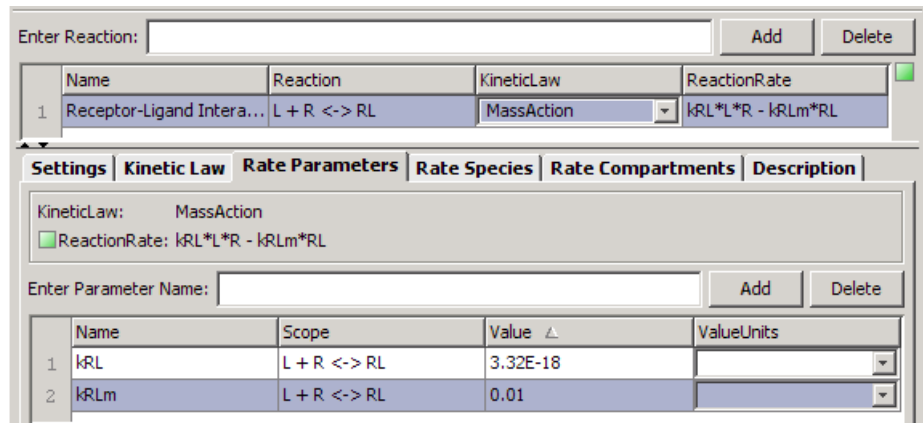
### Using Undo and Redo

Notice **Set ParameterVariableNames** listed in the status bar at the bottom of the pane. This listing shows that the most recent action on the desktop was to set **ParameterVariableNames**.

To **Undo** or **Redo** an action or a sequence of actions, right-click the listing (for example, **Set ParameterVariableNames**) and select the action to perform. There are some actions that the desktop cannot undo (for example adding or deleting blocks in the **Diagram**), these actions clear the sequence stack and you can no longer access the previous sequence of undo and redo actions.

**7** Enter the parameter values in the **Rate Parameters** tab:

$k_{RL} = 3.32E-18$  and  $k_{RLm} = 0.01$ .



## Setting Compartment Name

All models created in the desktop contain a compartment by default. The process of adding a reaction automatically adds the reaction species to a compartment in the model. If there are multiple compartments in the model, you must specify the reactants and products using qualified names (*compartmentName.speciesName*). For example nucleus.DNA denotes the species DNA, in the compartment nucleus.

This example contains only one compartment. Give the compartment a different name by doing the following:

- 1** In the **Project Explorer** double-click **Compartments**.
- 2** In the **Name** cell, double-click and type a name for the compartment and press **Enter**.

```
yeast_cell
```

The compartment table updates with the new name.

Use the default values for the **Owner**, **Capacity**, and **CapacityUnits** cells.

Below is a brief explanation of why you can leave these properties empty.

- **Owner** lets you define whether the compartment is within another compartment. This model has only one compartment within which you define all the species. No **Owner** means that the compartment is a top-level compartment.
- **Capacity** lets you enter the compartment size, such as volume, length, or area. In this model, species are defined in amounts, the reaction rate dimensions are amount/time, and the model assumes that compartment volume is constant. Thus, all the values and the units are internally consistent with each other and the compartment volume can use the default value of 1.0.
- **CapacityUnits** lets you specify the units for **Capacity**, which are not needed for this example.

## Setting Initial Amounts of Species

You can set the initial amounts of all the model species in the **Species** pane.

Name	InitialAmount
L	6.022E17
R	10000.0
RL	0.0

- 1 In the **Project Explorer**, expand **Compartments** and double-click **yeast\_cell Species**.
- 2 In the **Species** pane, double-click in each **InitialAmount** cell and enter the preceding values.

Enter Name: <input type="text"/>				Add	Delete
	Name	Scope	InitialAmount	InitialAmountUnits	
1	L	yeast_cell	6.022E17	<input type="text"/>	▼
2	R	yeast_cell	10000.0	<input type="text"/>	▼
3	RL	yeast_cell	0.0	<input type="text"/>	▼

Notice that the **Scope** column lists the name of the compartment containing the species.

You now have a complete reaction with all components added and defined in SimBiology.

## Completing the SimBiology Model

### In this section...

“Reactions” on page 2-21

“Alternative Ways to Build Reactions in the Desktop” on page 2-22

“Parameters” on page 2-24

“Species” on page 2-25

“Creating a Rule for the G Protein Model” on page 2-28

“Verifying the Model” on page 2-29

### Reactions

The previous sections of this tutorial show you how to enter the first reaction for the yeast G protein cycle model and configure the reaction rate equation in SimBiology. Repeat the procedures to add the rest of the reactions, parameters, and species values as described in the previous sections, and create a rule to specify the ratio of active G protein that corresponds to the ratio determined experimentally in the referenced study [Yi et al. 2003].

Add reactions 2 through 6 listed in the following table, set the kinetic law for each reaction to `MassAction`, create parameters, and configure the reaction rate using the procedure for “Determining the Reaction Rate Equation” on page 2-15.

No.	Name	Reaction	Rate Parameters
1	Receptor-Ligand Interaction	$L + R \leftrightarrow RL$	$k_{RL}, k_{RLm}$
2	Heterotrimeric G protein formation	$G_d + G_{bg} \rightarrow G$	$k_{G1}$
3	G protein activation	$RL + G \rightarrow Ga + G_{bg} + RL$	$k_{Ga}$
4	Receptor synthesis and degradation	$R \leftrightarrow null$	$k_{Rdo}, k_{RS}$

No.	Name	Reaction	Rate Parameters
5	Receptor-Ligand degradation	RL -> null	kRD1
6	G protein inactivation	Ga -> Gd	kGd

Your reaction table should look similar to the following figure:

Enter Reaction: <input type="text"/>				Add
	Name	Reaction	KineticLaw	ReactionRate
1	Receptor-ligand	L + R <-> RL	MassAction	kRL*L*R - kRLm*RL
2	Heterotrimeric G protein formation	Gd + Gbg -> G	MassAction	kG1*Gd*Gbg
3	G protein_activation	RL + G -> Ga + Gbg + RL	MassAction	kGa*RL*G
4	Receptor synthesis and degradation	R <-> null	MassAction	kRdo*R - kRs
5	Receptor-Ligand degradation	RL -> null	MassAction	kRD1*RL
6	G protein inactivation	Ga -> Gd	MassAction	kGd*Ga

The next section describes other ways you can build reactions in the desktop. If you want to continue building the model, skip the next section and go to “Parameters” on page 2-24.

### Alternative Ways to Build Reactions in the Desktop

This optional section shows you some alternative methods to build a reaction in the **Reactions** pane. Use the examples above to try out these methods. If you have finished entering the reactions for the model proceed to “Parameters” on page 2-24

The **Reactions** pane has several dialog boxes that are convenient for building reactions.

#### Building a Reaction

You can graphically build a reaction using the Reaction Builder.


- 1** If you are not already in the **Reactions** pane, in the **Project Explorer** double-click **Reactions**.
- 2** Click **Build** to open the Reaction Builder dialog box.
- 3** Select a species from the **Available Species** list and click the **Reactant** or **Product** button.
- 4** To edit stoichiometric relationships, click in the **Stoich** column and type.
- 5** Select the **Reversible** check box if the reaction is reversible.
- 6** Click **Add** to continue editing within the Reaction Builder. Click **OK** to finish and return to the **Reactions** pane.

### Creating a Binding Reaction

Use the **Bind** button to create a bound product from two reactant species.

- 1** If you are not already in the **Reactions** pane, in the **Project Explorer** double-click **Reactions**.
- 2** In the **Enter Reaction** box, enter the reactant species.

lactose + lactase

- 3** Click the  **Bind** button. SimBiology writes the product as a compound name, with a colon between reactant names to indicate binding.


lactose:lactase

### Creating an Unbinding Reaction

Use the **Unbind** button to create two product species from a bound reactant.

- 1** If you are not already in the **Reactions** pane, in the **Project Explorer** double-click **Reactions**.
- 2** In the **Enter Reaction** box, enter the reactant species.

lactose:lactase

- 3 Click  Unbind. SimBiology writes the product names with a + between them to indicate unbinding.

lactose + lactase

### Parameters

In the **Reactions** pane, after you create parameters in the **Kinetic Law** tab, you can set the value of the parameter in the **Rate Parameters** tab.

Alternatively, you can set all the parameters in the **Parameters** pane. In the **Project Explorer** double-click **Parameters** to open the **Parameters** pane.

Use the following values for rate parameters:

#### Parameter Value Table

Name	Value
kRLm	0.01
kRL	3.32E-18
kRdo	4.0E-4
kRs	4.0
kRD1	0.0040
kG1	1.0
kGa	1.0E-5
kGd	0.11

In order to be consistent with units for kRL, RL and L, the value for kRL is converted from the published value,  $2.0E6M^{-1}s^{-1}$ , to  $3.32E-18$  with units  $1/(molecule*second)$  (assuming a volume of unity).



Enter Name:		Add	Delete
Name	Scope $\Delta$	Value	ValueUnits
1	kRLm	L + R $\leftrightarrow$ RL	0.01
2	kRL	L + R $\leftrightarrow$ RL	3.32E-18
3	kRdo	R $\leftrightarrow$ null	4.0E-4
4	kRs	R $\leftrightarrow$ null	4.0
5	kRD1	RL $\rightarrow$ null	0.0040
6	kG1	Gd + Gbg $\rightarrow$ G	1.0
7	kGa	RL + G $\rightarrow$ Ga + Gbg + RL	1.0E-5
8	kGd	Ga $\rightarrow$ Gd	0.11

## Species

Set the species amounts in the **Species** pane. In the **Project Explorer**, double-click **yeast\_cell Species** to access this pane.

- Note that the process of adding reactions resulted in species automatically being added to the species list, with default amounts set to 0.0. Double-click each **InitialAmount** cell to change the values to those given in the table.
- The amount of L ( $\alpha$ -factor) used in the experiments is 1 M. This value when converted to molecule (assuming a volume of unity) is 6.022E17. This is now internally consistent with the units for the species, RL, and the parameter, kRL.

### Species Initial Amounts

Name	InitialAmount (Molecule)
L	6.022E17
R	10000.0
G	7000.0
Gd	3000.0
Gbg	3000.0

**Species Initial Amounts (Continued)**

Name	InitialAmount (Molecule)
Ga	0.0
RL	0.0

In order to replicate the published results, the model needs the definition of the ratio of active G protein; call this ratio `Ga_frac`. `Ga_frac` is  $Ga/Gt$ , where `Ga` is active G protein ( $G\alpha$ -GTP) and `Gt` is the total amount of G protein in a cell. This relationship is defined using a rule, and the procedure to create this rule is described in “Creating a Rule for the G Protein Model” on page 2-28.

Define two additional species, called `Ga_frac` and `Gt` in SimBiology.

**Additional Species**

Name	InitialAmount (molecule)
<code>Ga_frac</code>	0.0
<code>Gt</code>	10000.0

To add a new species,

- 1** In the **Project Explorer**, double-click **Species** to open the **Species** pane.
- 2** In the **Enter name** box, enter the name of a new species, and then click **Add** or press **Enter**.

`Ga_frac`

The species table updates with the new entry and its row selected. Note that the species is now available in the **Settings** tab.

- 3** In the **Initial Amount** cell, enter a value for the amount of the species.

0.0

- 4** Repeat steps 1 through 3 for `Gt`. **InitialAmount** = 10000.0

- 5 In the **Settings** tab, select the **ConstantAmount** check box only for Gt, because the amount of Gt does not vary during the simulation.

Enter Name:  Add Delete

	Name	Scope	InitialAmount	InitialAmountUnits
1	Gt	yeast_cell	10000.0	

**Settings** | **Description**

Name:  
Gt

Scope:  
yeast\_cell

InitialAmount:  
10000.0

InitialAmountUnits:  
▼

ConstantAmount (Select if the species quantity cannot change during the simulation.)  
 BoundaryCondition (Select if the species quantity is not determined by the set of reactions.)

Your species table should now look similar to this:

Enter Name:  Add Delete

	Name	Scope	InitialAmount	InitialAmountUnits
1	L	yeast_cell	6.022E17	▼
2	R	yeast_cell	10000.0	▼
3	RL	yeast_cell	0.0	▼
4	Gd	yeast_cell	3000.0	▼
5	Gbg	yeast_cell	3000.0	▼
6	G	yeast_cell	7000.0	▼
7	Ga	yeast_cell	0.0	▼
8	Ga_frac	yeast_cell	0.0	▼
9	Gt	yeast_cell	10000.0	▼

Notice the yellow indicator; move your pointer over the indicator to see the warning message. The message shows that the species is never used, because `Ga_frac` and `Gt` are not yet defined in a rule. After you define the rule as described in the next section, the indicator shows one green square.

For additional help in each procedure, select **Help > SimBiology Desktop Help** for context-sensitive help.

### Creating a Rule for the G Protein Model

A SimBiology rule is a mathematical expression that modifies a species amount or a parameter value. Use an algebraic rule to define the value of the species `Ga_frac` in the model.

- 1 In the **Project Explorer**, double-click the **Rules**, to open the **Rules** pane.
- 2 In the **Enter Rule** box, type the expression, and then click **Add** or press **Enter**.

```
Ga_frac-Ga/Gt
```

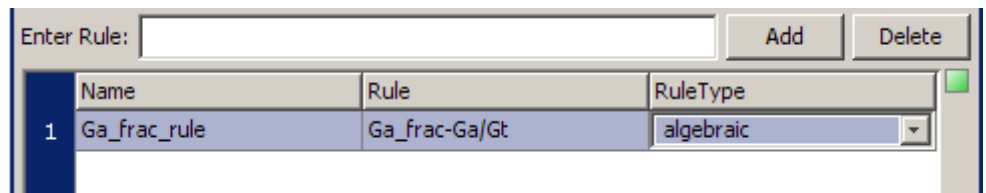
The rule is entered as an expression.

- 3 From the **RuleType** list, select

```
algebraic
```


- 4 Give your rule a name (optional). Double-click and type the following in the **Name** box.

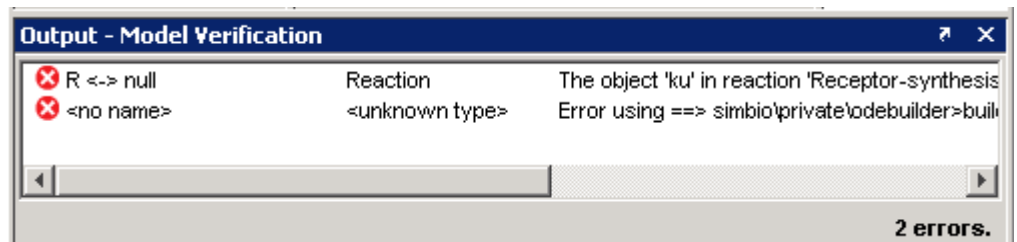
```
Ga_frac_rule
```



This completes the section on building the G protein cycle model for the wild-type strain.

## Verifying the Model

While you are building your model in the SimBiology desktop, you can click  at any time to generate a list of any errors and warnings in the model. The errors and warnings appear in the **Output** pane. Following is an example of the error generated when the reaction rate of a reaction is set to a parameter that you have not defined in SimBiology.



Use **Verify** to see if the example model shows any error or warnings. Double-click the error row to move to the location of the error.

You can interrupt and stop the verification process by clicking **Stop** at any time during verification.

## Simulating the G Protein Cycle Model in SimBiology

In this section...
“Setting Conditions Before Simulating the G Protein Model” on page 2-30
“Simulation Results for the Model of the Wild-Type Strain” on page 2-34

### Setting Conditions Before Simulating the G Protein Model

In the previous sections, this tutorial described building a model in SimBiology for a G protein cycle. This model uses the G protein cycle in the yeast pheromone response pathway. This section describes conditions for simulation and the simulation results for this model.

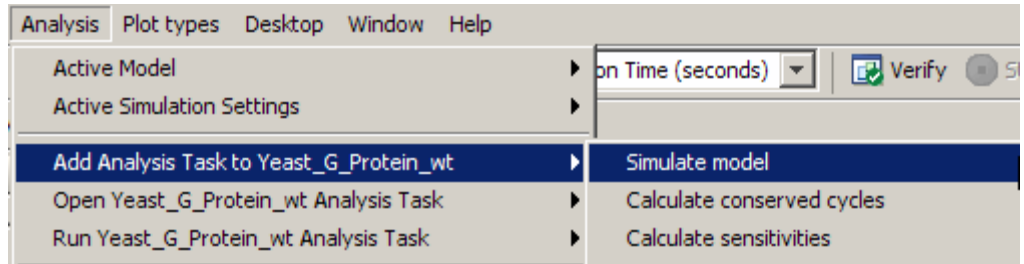
Consider the following points about simulating this model:

- Yi., et al. show data up to 600s for the active G protein time course. To replicate these results, change the simulation settings from the default 10 second to 600 second. This change remains active for this model unless you change it back to the default.
- The ligand species 'L' has values that are magnitudes higher than those of many of the other species. Don't log data for 'L' so that while plotting you can enable instant visualization of the other species through proper scaling of plots. To do this, define **Data Logging** to stop logging data for 'L'.

The first procedure (below) describes how to change the simulation stop time. The second procedure is about recording a subset of data (“Specifying Which Data Is Recorded by SimBiology” on page 2-32).

### Adding a Simulation Task

- From the **Analysis** menu, select **Add Analysis Task to Yeast\_G\_Protein\_wt > Simulate model**



The SimBiology desktop adds the task to the **Project Explorer**, opens the task pane, and highlights the new task.

## Changing the Simulation Stop Time

Change the stop time to replicate the simulation used in the reference paper [Yi et al. 2003], and to facilitate comparison with the experimental results presented in the study.

You will modify the default settings for this example. If you do not want to modify the default in your models, you can add another configuration set that you can then modify.

- 1 In the **Simulation** task pane, click the **Simulation Settings** tab.
- 2 In **Configuration Settings**, click [View](#). The **Configuration Settings** pane opens.
- 3 In the **Settings** tab, enter the stop time in the **Stop** box and press **Enter**.

600.0

- 4 From the **DefaultSpeciesDimension** list, select substance.

Leave the following properties as the default:

- **SolverType** (ode15s(stiff/NDF)).
- **AbsoluteTolerance** (1.0E-6)
- **RelativeTolerance**(0.0010)
- **MaxStep**([])

- **DimensionalAnalysis** (check box selected)
- **UnitConversion** (check box clear)

You can also change the solver to use and the stop time for the simulation in the SimBiology Toolbar. The desktop changes the setting of the Active configuration set when you change settings in the **Toolbar**. If you do not want to change the Active configuration set, you can add another configuration set and use the new settings. To add a new configuration set, double-click **Configuration Settings**, in the **Project Explorer** under **Model Variable Settings**. The **SimBiology Desktop Help** updates with more information.

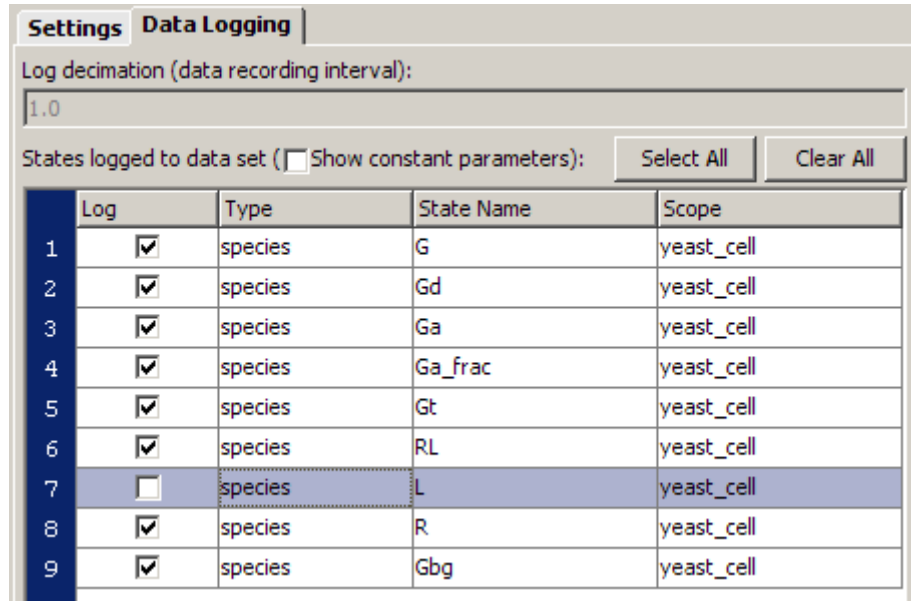
### **Specifying Which Data Is Recorded by SimBiology**

You can specify the species names for which SimBiology should *log* that is record the simulation data. The data set specified is plotted in the figure window.

As mentioned in the previous section, you will modify the default settings for this example. If you do not want to modify the default in your models, you can add another configuration set that you can then modify.

- 1** In the **Configuration Settings** pane, select the **Data Logging** tab.
- 2** Select the check boxes for the species to log. Because, all the species check boxes are selected by default, clear the check box for 'L'.





Alternatively, if you choose to log the data for all the species, you can use the **Plot Browser** to select the subset of species you want to view in your plot. This is described in .

### Tips for Use and Some Points to Consider


- You can save all these simulation settings as one custom simulation setting. See the context-sensitive help for **Simulation Settings** to find help on this topic. To access context-sensitive help select **Help > SimBiology Desktop Help**.
- The default `ode15s(stiff/NDF)`, is adequate for modeling of many biological pathways. You might, however, need a different solver for some models. For more information on choosing solver types, see [Selecting a Solver](#).
- SimBiology gives you a choice of three stochastic solvers: `stochastic` (SSA), `implicit tau`, and `explicit tau`. Try one of the stochastic solvers with this model and see how it compares with `ode15s`. For information, see [Stochastic Solvers](#).

Stochastic Solvers. You can also see how the stochastic solvers compare with each other.

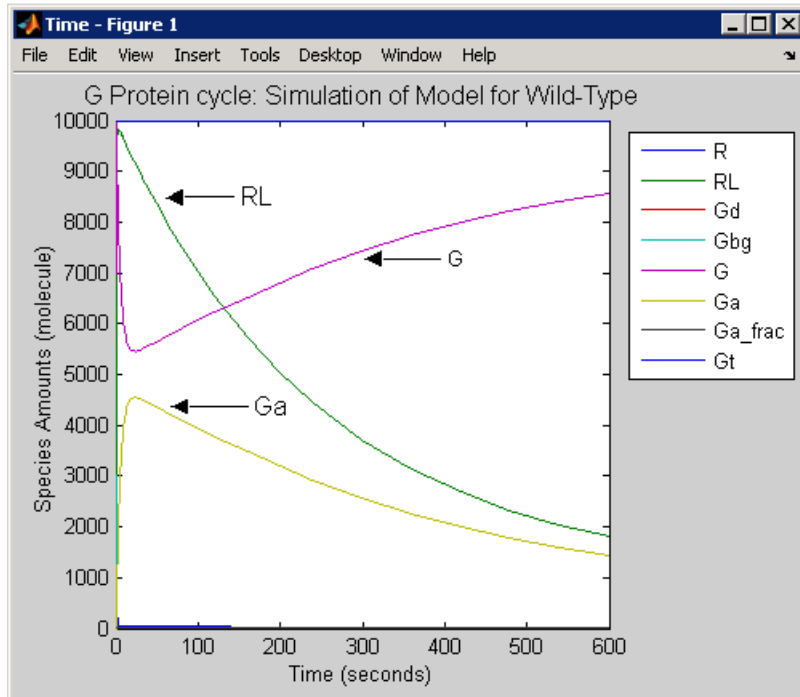
- For a counter that tracks the simulation, look in the lower-right corner of the SimBiology desktop.
- Click the following links to learn more about absolute and relative tolerance. These are links to SimBiology reference pages with definitions for `AbsoluteTolerance` and `RelativeTolerance`.

### Simulation Results for the Model of the Wild-Type Strain

Simulate the model you have built and see your results. To simulate the model,

- 1 In the **Project Explorer**, double-click the **Simulation** task, to open the **Simulation** pane.
- 2 Click  (**Run**).

Your plot should look similar to the figure below. Note that the title and the X and Y axes labels will not appear in your plot. But you can add titles and labels using the **Insert** menu in the figure window.

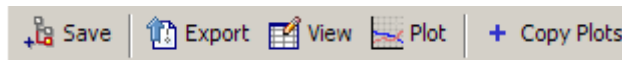


To view all species, see for help on using the **Plot Browser**.

### Saving Simulation Data

You can optionally save the data from the most recent simulation run. Unless you save the data for each simulation run, it is overwritten by the data for the next run.

- 1** In the **Project Explorer**, under **Simulation**, double-click **Data** for the wild-type model.
- 2** In the **Data** pane, click **Save**.



- 3** In the Save Data dialog box, specify a name for your data, and then click **Save**.

wt\_model\_run1

The desktop adds the saved data under **Data** in the **Project Explorer**.

## Building a Model for the Mutant Strain

### In this section...

“Modeling the Mutant Strain” on page 2-37

“Copying a Model” on page 2-37

“Changing Model Settings” on page 2-39

### Modeling the Mutant Strain

The deletion in SST2, results in uncatalyzed G protein deactivation (Reaction 6;  $G_a \rightarrow G_d$ ). From a modeling perspective this means a change in the rate of the reaction. This section shows you how to copy the previously built model into a new model session, change the parameter value to reflect the change in the rate of the reaction, and simulate the new model.

Another method to apply alternate values in the model is by using “Variants”. For more information about using variants in this example see “Applying Alternate Values Using Variants” on page 2-38.

---

**Note** An additional simplifying assumption of this model is that there are no changes in the initial amounts of species or the rate of any other reaction.

---

### Copying a Model

To copy a model, you first need to save the model to be copied as a SimBiology project. If you have not yet done so, see “Saving Your Work as a SimBiology Project File” on page 2-13.

Use the following procedure to import a model created in your SimBiology project. This is a convenient way to “copy” a model in the project and change the copied model while preserving the previous version.

- 1 From the **File** menu, select **Import Project Contents**.

The Open SimBiology Project dialog box opens.

- 2 Browse to and select the project.

```
yeast_g_protein_cycle.sbproj
```

**3 Click Open.** The imported model opens in a new model session.

This section showed you how to copy a model into a separate model session and change a component value to represent, for example a mutant phenotype. The complete example uses this method to illustrate the work flow.

The next section shows you an another method to apply alternate values to model components through variants. You can also try the same work flow using the variant. To skip the next section and continue with the example see “Changing Model Settings” on page 2-39.

### Applying Alternate Values Using Variants

If you want to use a variant to apply the change in parameter value:

**1** In the **Project Explorer**, expand **Model Variable Settings** (for the wild-type model), and double-click **Variants** to open the **Variants** pane.

**2** In the **Enter Name** box, type a name for the variant, and then click **Add** or press **Enter**. For example:

```
mut_value
```

**3** Add content to the variant:

**a** In the **Settings** tab, click **Add**. The table updates with a row containing information about a model component.

**b** From the **Type** list, select parameter.

The **Property** list updates to show the property available for storing.

**c** In the **Name** cell, type the name of the component.

```
Gprotein Inactivation.kGd
```


The parameter kGd is at the kinetic law level, and not the model level. Thus, you must specify the parameter in the format *ReactionName.ParameterName*.

**d** In the **Value** cell, type a value to apply using the variant.

0.004

If you want to use the variant for the rest of this example, you must select the variant during simulation, to represent the mutant phenotype.

To apply the variant during simulation:

- 1** In the **Project Explorer**, double-click the **Simulation** task, to open the **Simulation** pane.
- 2** In the **Variants** table, select the **Use In Task** check box for mut\_value.
- 3** Click  (**Run**).

## Changing Model Settings

The copied model has the simulation settings used for the previous model. Therefore, the only settings to change are the name of the model and the value of the parameter kGd. Remember to select the appropriate items in the *newly* created model session.

- 1** Rename the model. In the **Project Explorer**, double-click **SimBiology Model** to open the **SimBiology Model**.
- 2** Click the **Settings** tab.
- 3** In the **Model Name** box, enter the following:

Yeast\_G\_Protein\_mut

- 4** Locate the parameter kGd for the Yeast\_G\_Protein\_mut using **Find** in the top-right corner of the SimBiology desktop.
  - a** In the **Find** box, enter kGd and press **Enter**. SimBiology opens the **Output** pane with the results of the search.
  - b** Locate the parameter kGd for the mutant strain. Note that SimBiology organizes the results by model and lists the model component.
  - c** Double-click the row containing the parameter kGd under the model Yeast\_G\_Protein\_mut. The desktop opens the **Parameters** pane with the row containing kGd selected.

- d** Change the value of  $k_{Gd}$  from 0.11 to 0.004.



## Simulating the G Protein Cycle Model of the Mutant Strain

### In this section...

“Creating a Custom Plot for Simulation Results” on page 2-41

“Plotting Using a Custom Plot” on page 2-42

“Simulation Results for the Model of the Mutant Strain” on page 2-43

### Creating a Custom Plot for Simulation Results

To keep the data plots from each of the model simulations distinct and to facilitate comparison, you can customize one of the plots. This example shows you how to create and save a custom plot to plot the simulation data with dashed lines.

- 1** Select **Plot Types > New**. A new **Plot Types** pane opens and the desktop adds the new plot type in the **Project Explorer** named `untitled1`
- 2** In the **Project Explorer**, right-click `untitled1` and select **Rename Plot Type**.
- 3** In the Rename Plot Type dialog box, specify a name for the plot type. For example,  

```
Time plot with dashed line
```
- 4** In the **Plot Types** pane from the **Copy Plot Type** list select `Time`.
- 5** Click **Update**.

The **Plot Types** command window updates:

```
Plot code with input arguments: (tobj, y)
1 if (length(tobj) > 1)
2     sbioplot(tobj, @timeplotdata, [], y);
3 else
4     timeplotdata(tobj, [], y);
5 end
6
7 %-----
8 function [handles, names] = timeplotdata(tobj, x, y)
9
10 handles = [];
11 for i=1:length(tobj)
12     % Get the data from the next run.
13     nexttobj = tobj(i);
14
15     % Get the simulation data associated with the species
16     % specified in y.
17     if strcmpi(y, '<all>')
18         [time, data, names] = getdata(nexttobj);
19     else
20         [time, data, names] = selectbyname(nexttobj, y);
21     end
22
23     % Error checking.
24     if size(data,2) == 0
25         error('Species specified do not exist.');
```

6 Make the following change to `hline = plot (time,data);` (line 29):

```
hLine = plot (time,data,'LineStyle','--');
```

This plot type is now available for use with any task.

### Plotting Using a Custom Plot


Plot the simulation data for the `Yeast_G_Protein_mut` model with dashed lines.

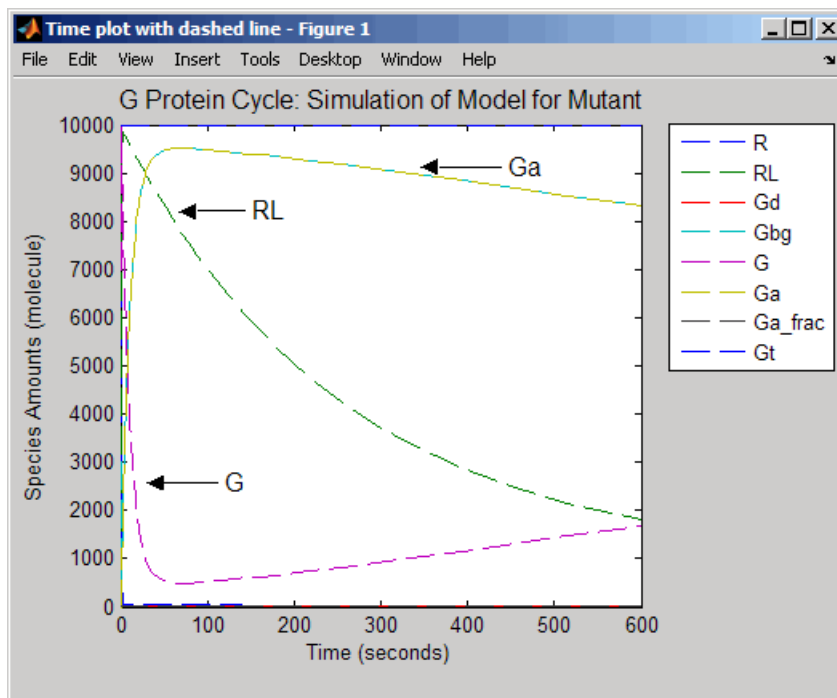
- 1** In the **Project Explorer**, double-click **Simulation** for the Yeast\_G\_Protein\_mut model. The **Simulation** pane opens
- 2** In the **Simulation** pane, click the **Plot Results** tab.
- 3** From the **Plot Type** list, select the new plot type (Time plot with dashed line) and click **Add Plot Type**.
- 4** In the plot type table, double-click the **Y Arguments** cell for the new entry. This opens the Select Y Arguments dialog box. Click **Select All** and then click **OK**.
- 5** In the plot types table, clear the **Create Plot** check box for the Time plot.

When you simulate the model, the plot is generated with dashed lines.

## Simulation Results for the Model of the Mutant Strain

To simulate the model,

- In the **Simulation** pane, click  (**Run**)



The simulation results for the wild-type strain are described in “Simulation Results for the Model of the Wild-Type Strain” on page 2-34.

## Plotting Species from Two Different Data Sets

### In this section...

“Procedures Described in this Section” on page 2-45

“Plotting the Active G Protein Fraction from the Model of the Wild-Type Strain” on page 2-45

“Plotting the Active G Protein Fraction from the Model of the Mutant Strain” on page 2-46

### Procedures Described in this Section

This section shows you the following:

- How to compare the active G protein fractions in the two simulations. The procedures in this section show you how to create a plot showing species from the two data sets in this tutorial.
- How to plot the data without having to rerun the simulation, and more about how to generate custom plots.

### Plotting the Active G Protein Fraction from the Model of the Wild-Type Strain

You can find the simulation data under **Simulation and Analysis** for each model. You can also save the data from previous simulations, see “Saving Simulation Data” on page 2-35 for more information.

Specify that the species `Ga_frac` should be plotted for the wild-type model.

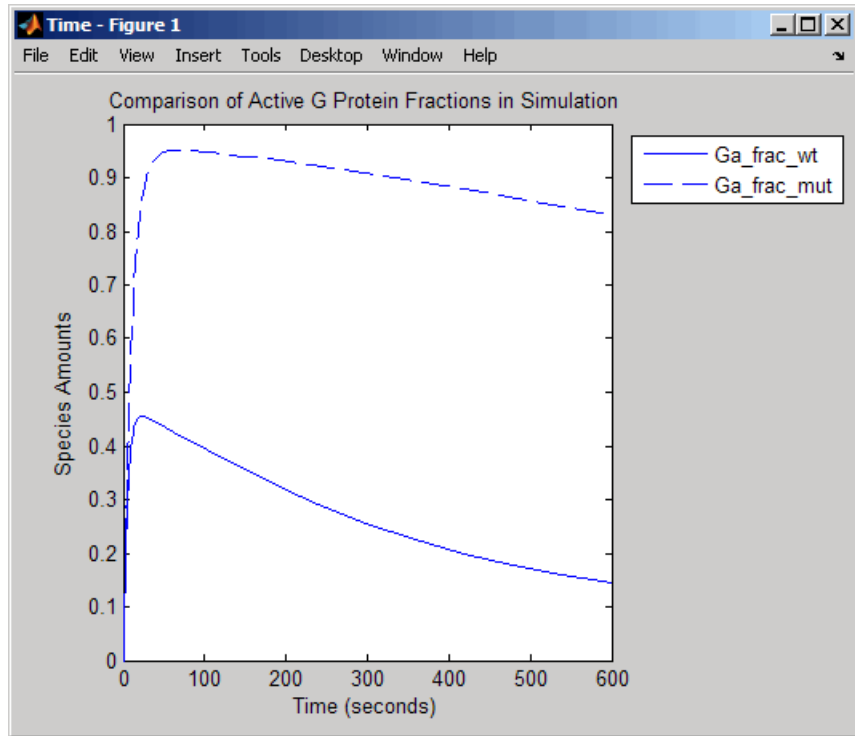
- 1 In the **Project Explorer**, double-click **Data** for the wild-type model.
- 2 In the **Data** pane, select the **Plots** tab.
- 3 From the **Plot Type** list, select the Time plot and click the **Add Plot Type** button.
- 4 In the plot type table, double-click the **Y Arguments** cell for the new entry. This opens the Select Y Arguments dialog box.

- 5 Select `yeast_cell.Ga_frac` and click **OK**.
- 6 In the plot type table, clear the **Create Plot** check boxes for the other plots, and click **Plot**. Leave the figure window open.

### Plotting the Active G Protein Fraction from the Model of the Mutant Strain

Plot the *sst2Δ* `Ga_frac` data using dashed lines.

- 1 In the **Project Explorer**, double-click **Data** for the `Yeast_G_Protein_mut (sst2Δ)` model.
- 2 In the **Data** pane, select the **Plots** tab.
- 3 From the **Plot Type** list, select the Time plot with dashed line plot and click **Add Plot Type**.
- 4 In the plot type table, double-click the **Y Arguments** cell for the new entry. This opens the Select Y Arguments dialog box. Select `yeast_cell.Ga_frac` and click **OK**.
- 5 In the plot type table, clear the **Create Plot** check box for the other plots.
- 6 From the **Plot Behavior** list select Add to current axes and click **Plot**. This option lets you add data to the most recently generated plot. You must have the figure window open to exercise this option.



## Plotting Experimental Data with Simulation Data

In this section...
“About the Experimental Data” on page 2-48
“Creating a Custom Plot for Experimental Data” on page 2-48
“Plotting the Data” on page 2-50

### About the Experimental Data

SimBiology enables you to work with your experimental data and plot your data from the SimBiology desktop. The section describes how to store the experimental data and use the custom plotting features to plot the data with your simulation data.

This example uses the yeast G protein model built in this tutorial, using a reference paper published by Yi and colleagues (Yi et al. 2003). The experimental data used here are also from the same reference paper.

### Creating a Custom Plot for Experimental Data

- 1** Select **Plot Types > New**. A new **Plot Types** pane opens and the desktop adds the new plot type in the **Project Explorer** named untitled1
- 2** In the **Project Explorer**, right-click untitled1 and select **Rename Plot Type**.
- 3** In the Rename Plot Type dialog box specify a name for the plot type. For example,

Ga\_frac Experimental Data Plot

- 4** In the **Plot Types** pane, copy the following code into the plot types editor window:

```
% 1. Store the time and state data
%(Obtained from Fig. 5 of reference paper.)
x = [0 10 30 60 110 210 300 450 600];
y = [0 0.35 0.4 0.36 0.39 0.33 0.24 0.17 0.2];
```



```

% 2. Store the estimated error values.
%(Obtained from Fig. 5 of reference paper.)
L = [0 0.0100 0 0.0100 0.0200 0.0200 0.0300 0.0200 0.0200];
U = [0 0.0100 0 0.0200 0.0100 0.0180 0.0350 0.0300 0.0100];

% 3. Plot the experimental data.
errorbar(x,y,L,U,'LineStyle','none','Marker','.');
legH3 = legend('Ga_frac_sim','Ga_frac_exp','Location','NorthEastOutside');
set(legH3,'Interpreter','none')

% To make a better picture,
axis([0 600 0 0.5]);

```

### Explanation of the code

In step 1 you store the data as vectors in two variables, one for the experimental values of active G protein fractions ( $x$ ), and the other for the time points ( $y$ ). By writing scripts in this command window, you can store and process your experimental data before plotting.

In step 2 you store the estimated upper ( $U$ ) and lower ( $L$ ) bounds of the error values of each data point.

In step 3, the function `errorbar` enables you to plot  $x$  versus  $y$  with error bars  $L(i)+U(i)$  long.  $x$ ,  $y$ ,  $L$ , and  $U$  must be the same size. When they are vectors, each error bar is a distance of  $L(i)$  below and  $U(i)$  above the point defined by  $(x(i),y(i))$ . For more information, see `errorbar`.

The code also specifies legend location, marker style, line style, and axis scale. For more information, see `legend` and `axis`.

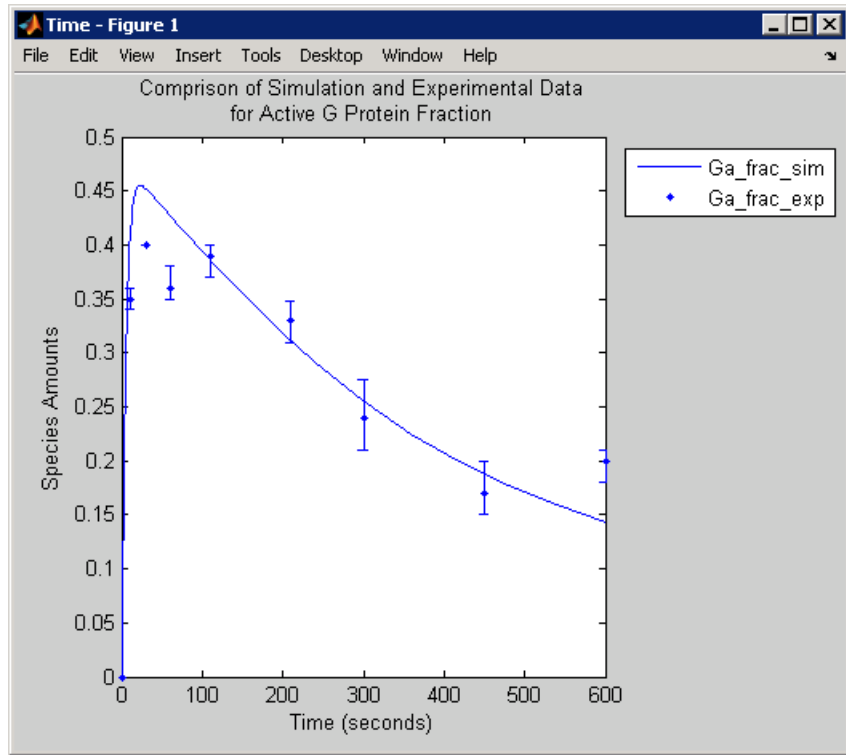
- 5** Clear the **X Arguments must be specified** and **Y Arguments must be specified** check boxes. The  $x$  and  $y$  arguments are already specified in the code that you entered in step 4.

The new custom plot appears under **Plot Types** in the **Project Explorer**.

### Plotting the Data

You can add the new custom plot to the data plot for the model of the wild-type strain and plot the simulation data for Ga\_frac with the experimental data.

- 1** In the **Project Explorer**, double-click **Data** for the wild-type model.
- 2** In the **Data** pane, click the **Plots** tab.
- 3** From the **Plot Type** list, select the Ga\_frac Experimental Data Plot plot and click **Add Plot Type**.
- 4** Select the **Create Plot** check box for the following:
  - a** For the Time plot in which only Ga is specified in the **Y Arguments** list, select New Figure from the **Plot Behavior** list.
  - b** For the Ga\_frac Experimental Data Plot, select Add to current axes from the **Plot Behavior** list.
- 5** In the plot type table, clear the **Create Plot** check boxes for the other plots and click **Plot**.



For more information on the plot used in step 4a see “Plotting the Active G Protein Fraction from the Model of the Wild-Type Strain” on page 2-45

## References

- [1] Tau-Mu Yi, Hiroaki Kitano, and Melvin I. Simon. A quantitative characterization of the yeast heterotrimeric G protein cycle. PNAS (2003) vol. 100, 10764-10769.
- [2] Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., and Watson, J.D. Molecular Biology of the Cell, 3rd edition, Garland Publishing, 1994.

# M-Phase Control in *Xenopus* Oocyte Extracts

---

John Tyson's Computational Cell Biology Lab created a mathematical model for M-phase control in *Xenopus* oocyte (frog egg) extracts (Marlovits et. al. 1998). The M-phase control model shows principles by which you can apply phosphorylation and regulatory loops in your own models. Publications typically list systems of ordinary differential equations (ODEs) that represent a model system. This example shows you how to interpret these ODEs in the form of reaction pathways that are easier to represent and visualize in SimBiology.

The model is centered around M-phase promoting factor (MPF). There are two positive feedback loops where MPF increases its synthesis and a negative feedback loop where MPF decreases its amount by increasing its degradation.

M-Phase Control Model (p. 3-2)	Diagram of reaction pathways and model assumptions
M-Phase Control Equations (p. 3-4)	Differential rate equations for each varying species and supporting algebraic equations
SimBiology Model with Rate and Algebraic Rules (p. 3-14)	Enter differential rate equations directly from the literature
SimBiology Model with Reactions and Algebraic Rules (p. 3-22)	Convert differential rate equations to reactions and reaction rates
References (p. 3-40)	Literature references and Web links for additional information

## M-Phase Control Model

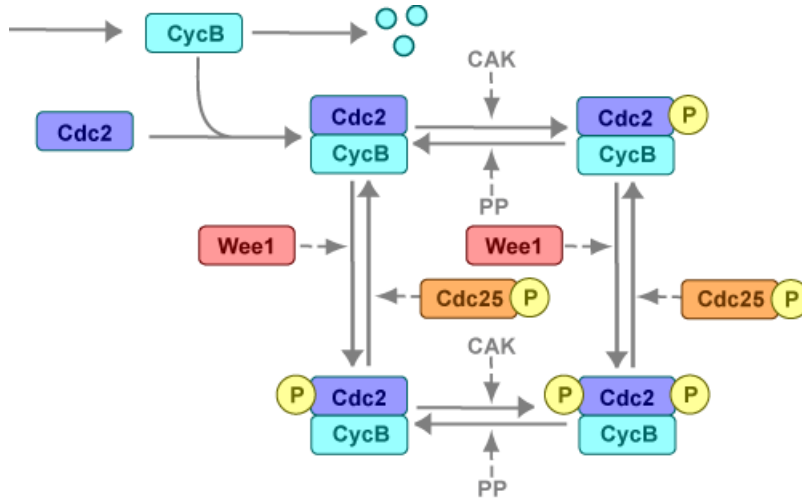
### In this section...

“Synthesis Reactions” on page 3-2

“Regulation Reactions with Active MPF” on page 3-2

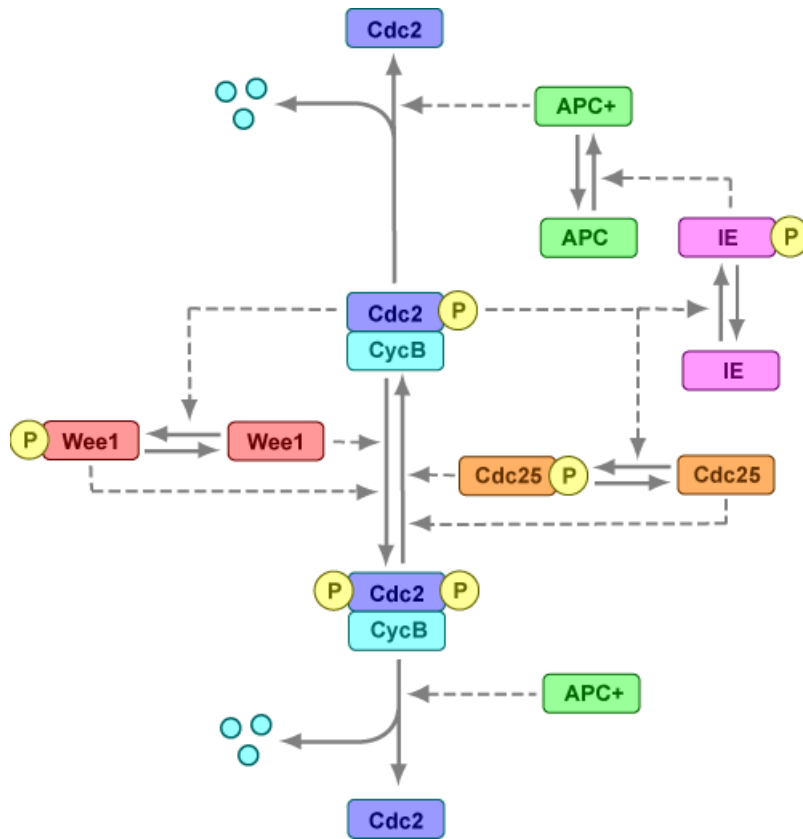
### Synthesis Reactions

Cyclin B (CycB) dimerizes with Cdc2 kinase (Cdc2) to form M-phase promoting factor (MPF).



### Regulation Reactions with Active MPF

Positive feedback loops with M-phase promoting factor (MPF) activate the Cdc25 phosphatase and deactivate the Wee1 kinase. A negative feedback loop with MPF activates anaphase-promoting complex (APC) that regulates the degradation of the Cyclin B subunit.



## M-Phase Control Equations

### In this section...

“About the Rate Equations Described in this Example” on page 3-4

“Converting Differential Equations to Reactions” on page 3-5

“Equation 1 Cyclin B” on page 3-6

“Equation 2 M-Phase Promoting Factor (MPF)” on page 3-6

“Equation 3 Inhibited M-Phase Promoting Factor (pMPF)” on page 3-7

“Equation 4 Inhibited and Activated M-Phase Promoting Factor (pMPFp)” on page 3-8

“Equation 5 Activated M-Phase Promoting Factor (MPFp)” on page 3-8

“Equation 11 Cell Division Control 25 (Cdc25)” on page 3-9

“Equation 12 Wee1 Activation/Deactivation” on page 3-10

“Equation 13 Intermediate Enzyme (IE) Activation/Deactivation” on page 3-10

“Equation 14 APC Activation/Deactivation” on page 3-11

“Equation 17 Rate Parameter K2” on page 3-11

“Equation 18 Rate Parameter Kcdc25” on page 3-12

“Equation 19 Rate Parameter Kwee1” on page 3-12

### About the Rate Equations Described in this Example

Models in systems biology are commonly described in the literature with differential rate equations. However, SimBiology defines a model using reactions. This section shows you how to convert models published in the literature to a SimBiology format. The equation numbers match the published paper for this model (Marlovits et. al. 1998). Equations that are missing in the sequence involve the Cdk inhibitor protein (CKI), which is not currently modeled in the SimBiology version.



## Converting Differential Equations to Reactions

The rules for writing reaction and reaction rate equations from differential rate equations include not only the equations but also an understanding of the reactions.  $dx/dt$  refers to the species the differential rate equation is defining. *kinetics* refers to the species in the reaction rate.

- Positive terms: rate species are placed on right side of the reactions; reaction rate equation species are placed on the right.

$$\text{kinetics} \rightarrow \frac{dx}{dt}$$

- Negative terms: rate species are placed on the left side of the reaction because the species are being used up in some way; reaction rate equation species are placed on left. You need to deduce the products from additional information about the model.

$$\text{kinetics or } \left(\frac{dx}{dt}\right) \rightarrow \text{products?}$$

The following table will help you to deduce the products for a reaction. In this example, by convention, phosphate groups on the right side of a species name are activating while phosphate groups on left are inhibiting.

Enzyme	Description	Reaction
wee1	Kinase, add inhibiting phosphate group	MPF $\rightarrow$ P-MPF
cdc25	Phosphatase, remove inhibitory phosphate group	P-MPF $\rightarrow$ MPF + P
kcak	Kinase, add activating phosphate group	MPF $\rightarrow$ MPF <sub>p</sub>
kpp	Phosphatase, remove activating phosphate group	MPF-P $\rightarrow$ MPF + P
MPF	Kinase, add activating or inhibiting phosphate group	Wee1/Cdc25/IE $\rightarrow$ X-P or P-X

Enzyme	Description	Reaction
ki	Add inhibiting Cki	Cki + MPF → Cki:MPF
kir	Remove inhibiting Cki	Cki:MPF → Cki + MPF

### Equation 1 Cyclin B

Differential rate equation for Cyclin B (Marlovits et. al. 1998).

$$\frac{d[\text{CycB}]}{dt} = +k_1 - k_2[\text{CycB}] - k_3[\text{Cdc2}][\text{CycB}]$$

Rate rule using SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\text{Rule 1 } [\text{CycB}] = k_1 - K_2 * [\text{CycB}] - k_3 * [\text{Cdc2}] * [\text{CycB}]$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

Reaction 1	AA -> CycB	v = k1
Reaction 2	CycB -> AA	v = K2*[CycB]
Reaction 3	Cdc2 + CycB -> MPF	v = k3*[Cdc2]*[CycB]

### Equation 2 M-Phase Promoting Factor (MPF)

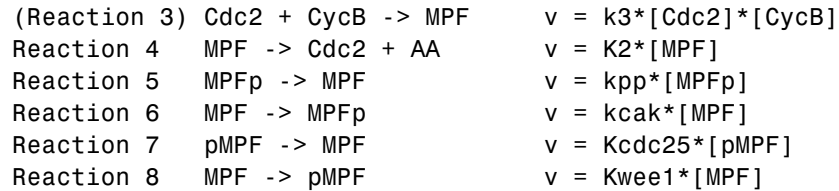
Differential rate equation for M-phase promoting factor (MPF) (Marlovits 1998). Note, the parameter name kcakr (Marlovits 1998) is changed to kpp (Borisuk 1998) in the following reaction equations. MPF is a heterodimer of Cdc2 kinase and Cyclin B.

$$\begin{aligned} \frac{d[\text{MPF}]}{dt} = & +k_3[\text{Cdc2}][\text{CycB}] - K_2[\text{MPF}] \\ & +k_{pp}[\text{MPF}_p] - k_{cak}[\text{MPF}] \\ & +K_{cdc25}[\text{pMPF}] - K_{wee1}[\text{MPF}] \\ & +k_{ir}[\text{Cki:MPF}] - k_i[\text{MPF}][\text{Cki}] \end{aligned}$$

Rate rule using SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\text{Rule 2 MPF} = k_{pp} \cdot \text{MPFp} - (K_{wee1} + k_{cak} + K_2) \cdot \text{MPF} + K_{cdc25} \cdot \text{pMPF} + k_3 \cdot \text{Cdc2} \cdot \text{CycB}$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22. A reaction name in parentheses denotes a reaction repeated in another differential rate equation.



### Equation 3 Inhibited M-Phase Promoting Factor (pMPF)

Differential rate equation for inhibited M-phase promoting factor (pMPF) (Marlovits 1998).

$$\begin{aligned} \frac{d[\text{pMPF}]}{dt} = & -K_2[\text{pMPF}] \\ & +k_{pp}[\text{pMPFp}] -k_{cak}[\text{pMPF}] \\ & +K_{wee1}[\text{MPF}] -K_{cdc25}[\text{pMPF}] \\ & +k_d[\text{Cki:pMPF}] \end{aligned}$$

Rate rule using SimBiology format for the differential rate equation 3. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\text{Rule 3 pMPF} = K_{wee1} \cdot \text{MPF} - (K_{cdc25} + k_{cak} + K_2) \cdot \text{pMPF} + k_{pp} \cdot \text{pMPFp}$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

Reaction 11	pMPF -> Cdc2 + AA	v = K2*[pMPF]
Reaction 12	pMPFp -> pMPF	v = kpp*[pMPFp]
Reaction 13	pMPF -> pMPFp	v = kcak*[pMPF]
(Reaction 8)	MPF -> pMPF	v = Kwee1*[MPF]
(Reaction 7)	pMPF -> MPF	v = Kcdc25*[pMPF]

### Equation 4 Inhibited and Activated M-Phase Promoting Factor (pMPFp)

Differential rate equation for inhibited and activated M-phase promoting Factor (pMPFp) (Marlovits 1998).

$$\frac{d[pMPFp]}{dt} = -K2[pMPFp] + kcak[pMPF] - kpp[pMPFp] + Kwee1[MPFp] - Kcdc25[pMPFp] + kd[Cki:pMPFp]$$

Rate rule using SimBiology format for the differential rate equation. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\text{Rule 4 } pMPFp = Kwee1*MPFp - (kpp + Kcdc25 + K2)*pMPFp + kcak*pMPF$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

Reaction 15	pMPFp -> Cdc2 + AA	v = K2*[pMPFp]
(Reaction 13)	pMPF -> pMPFp	v = kcak*[pMPF]
(Reaction 12)	pMPFp -> pMPF	v = kpp*[pMPFp]
Reaction 16	MPFp -> pMPFp	v = Kwee1*[MPFp]
Reaction 17	pMPFp -> MPFp	v = Kcdc25*[pMPFp]

### Equation 5 Activated M-Phase Promoting Factor (MPFp)

Differential rate equation for activated M-phase promoting factor (MPFp) (Marlovits 1998).

$$\begin{aligned} \frac{d[\text{MPFp}]}{dt} = & -K2[\text{MPFp}] \\ & +k_{\text{cak}}[\text{MPF}] -k_{\text{pp}}[\text{MPFp}] \\ & +K_{\text{cdc25}}[\text{pMPFp}] -K_{\text{wee1}}[\text{MPFp}] \\ & +k_{\text{ir}}[\text{CKI:MPFp}] -k_{\text{i}}[\text{CKI}][\text{MPFp}] \end{aligned}$$

Rate rule using SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\begin{aligned} \text{Rule 5 MPFp} = & k_{\text{cak}}*\text{MPF} - (k_{\text{pp}} + K_{\text{wee1}} + K2)*\text{MPFp} \\ & + K_{\text{cdc25}}*\text{pMPFp} \end{aligned}$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

Reaction 19	MPFp -> MPF + AA	$v = K2*[\text{MPFp}]$
(Reaction 6)	MPF -> MPFp	$v = k_{\text{cak}}*[\text{MPF}]$
(Reaction 5)	MPFp -> MPF	$v = k_{\text{pp}}*[\text{MPFp}]$
(Reaction 17)	pMPFp -> MPFp	$v = K_{\text{cdc25}}*[\text{pMPFp}]$
(Reaction 16)	MPFp -> pMPFp	$v = K_{\text{wee1}}*[\text{MPFp}]$

## Equation 11 Cell Division Control 25 (Cdc25)

Differential rate equation for activating and deactivating Cdc25 (Marlovits 1998).

$$\frac{d[\text{Cdc25p}]}{dt} = + \frac{k_{25}[\text{MPFp}][\text{Cdc25}]}{K_{m25} + [\text{Cdc25}]} - \frac{k_{25r}[\text{Cdc25p}]}{K_{m25r} + [\text{Cdc25p}]}$$

Rate rule in SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14. Note that, since there isn’t a rate rule for Cdc25, its amount is written as (TotalCdc25 - Cdc25p).

$$\text{Rule 11 } \text{Cdc25p} = (k_{25} \cdot \text{MPFp} \cdot (\text{TotalCdc25} - \text{Cdc25p})) / (K_{m25} + (\text{TotalCdc25} - \text{Cdc25p})) - (k_{25r} \cdot \text{PPase} \cdot \text{Cdc25p}) / (K_{m25r} + \text{Cdc25p})$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

$$\begin{aligned} \text{Reaction 36 } \text{Cdc25} &\rightarrow \text{Cdc25p}, & v &= k_{25} \cdot [\text{MPFp}] \cdot [\text{Cdc25}] / (K_{m25} + [\text{Cdc25}]) \\ \text{Reaction 37 } \text{Cdc25p} &\rightarrow \text{Cdc25}, & v &= k_{25r} \cdot [\text{Cdc25p}] / (K_{m25r} + [\text{Cdc25p}]) \end{aligned}$$

### Equation 12 Wee1 Activation/Deactivation

Differential rate equation for activating and deactivating Wee1 kinase (Marlovits 1998). The kinase (MPFp) phosphorylates active Wee1 (Wee1) to its inactive form (Wee1p). The dephosphorylation of inactive Wee1 (Wee1p) is by an unknown phosphatase.

$$\frac{d[\text{Wee1}]}{dt} = -\frac{k_w[\text{MPFp}][\text{Wee1}]}{K_{mw} + [\text{Wee1}]} + \frac{k_{wr}[\text{Wee1P}]}{K_{mwr} + [\text{Wee1P}]}$$

Rate rule in SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\begin{aligned} \text{Rule 12 } \text{Wee1p} &= (k_w \cdot \text{MPFp} \cdot (\text{TotalWee1} - \text{Wee1p})) / (K_{mw} + (\text{TotalWee1} - \text{Wee1p})) \\ &\quad - (k_{wr} \cdot \text{Wee1p}) / (K_{mwr} + \text{Wee1p}) \end{aligned}$$

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

$$\begin{aligned} \text{reaction 38 } \text{Wee1} &\rightarrow \text{Wee1p}, & v &= (k_w \cdot [\text{MPFp}] \cdot [\text{Wee1}]) / (K_{mw} + [\text{Wee1}]) \\ \text{reaction 39 } \text{Wee1p} &\rightarrow \text{Wee1}, & v &= (k_{wr} \cdot [\text{Wee1p}]) / (K_{mwr} + [\text{Wee1p}]) \end{aligned}$$

### Equation 13 Intermediate Enzyme (IE) Activation/Deactivation

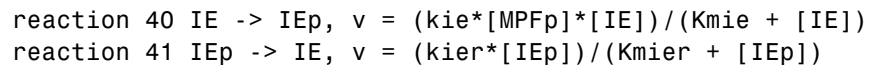
Differential rate equation for activating and deactivating the intermediate enzyme (IE) (Marlovits 1998). The active kinase (MPFp) phosphorylates the inactive intermediate enzyme (IE) to its active form (IEp).

$$\frac{d[IEp]}{dt} = + \frac{k_{ie}[MPFp][IE]}{K_{mie} + [IE]} - \frac{k_{ier}[IEp]}{K_{mier} + [IEp]}$$

Rate rule in SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

```
Rule 13 IEp = (kie*MPFp*(TotalIE - IEp))/(Kmie + (TotalIE - IEp))
             - (kier*IEp)/(Kmier + IEp)
```

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.



## Equation 14 APC Activation/Deactivation

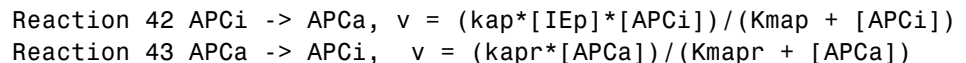
Differential rate equation for (Marlovits 1998).

$$\frac{d[APCa]}{dt} = + \frac{k_{ap}[IEP][APCi]}{K_{map} + [APCi]} - \frac{k_{apr}[APCa]}{K_{mapr} + [APCa]}$$

Rate rule in SimBiology format for the differential rate equation 1. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

```
Rule 14 APCa = (kap*IEp*(TotalAPC - APCa))/(Kmap + (TotalAPC - APCa))
             - (kapr*APCa)/(Kmapr + APCa)
```

Reaction and reaction rate equations derived from the differential rate equation. For a model using these reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.



## Equation 17 Rate Parameter K2

Algebraic equation to define the rate parameter K2 (Marlovits 1998). Inactive APC (APCi) is catalyzed by IE (intermediate enzyme) to active APC (APCa).

$$k2 = V2'[APC] + V2''[APC']$$

Algebraic rule in SimBiology format for the algebraic equation 17. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\text{Algebraic Rule 17 } V2i*(TotalAPC - APCa) + V2a*APCa - K2$$

Algebraic rule when simulating with reactions. For a model using this rule with reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.  $V2'$  is renamed to  $V2i$  and  $V2''$  is renamed to  $V2a$ .  $APCi$  ( $APC$ ) is the inactive form of the enzyme while  $APCa$  ( $APC'$ ) is the active form.  $K2$  is the independent variable.

$$\text{Algebraic Rule 1 } (V2i*APCi) + (V2a*APCa) - K2$$

### **Equation 18 Rate Parameter Kcdc25**

Algebraic equation to define the rate parameter  $Kcdc25$  (Marlovits 1998). Inactive  $Cdc25$  ( $Cdc25$ ) is phosphorylated by MPF to active  $Cdc25$  ( $Cdc25p$ ).

$$kcdc25 = V25'[Cdc25] + V25''[Cdc25p]$$

Algebraic rule in SimBiology format for the algebraic equation 18. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

$$\text{Algebraic Rule 18 } V25i*(TotalCdc25 - Cdc25p) + V25a*Cdc25p - Kcdc25$$

Algebraic rule when simulating with reactions.  $Kcdc25$  is the independent variable. For a model using this rule with reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

$$\text{Algebraic Rule 2 } (V25i*Cdc25) + (V25a*Cdc25p) - Kcdc25$$

### **Equation 19 Rate Parameter Kwee1**

Algebraic equation to define the rate parameter (Marlovits 1998). Active  $Wee1$  ( $Wee1$ ) is phosphorylated by MPF to inactive  $Wee1$  ( $Wee1p$ ).

$$kwee1 = Vwee1'[Wee1p] + Vwee1''[Wee1]$$



Algebraic rule in SimBiology format for rate parameter equation 19. For a model using this rule, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.

Algebraic Rule 19  $V_{wee1i} * Wee1p + V_{wee1a} * (TotalWee1 - Wee1p) - K_{wee1}$

Algebraic rule when simulating with reactions.  $K_{wee1}$  is the independent variable. For a model using this rule with reactions, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

Algebraic Rule 3  $(V_{wee1i} * Wee1p) + (V_{wee1a} * Wee1) - K_{wee1}$

## SimBiology Model with Rate and Algebraic Rules

### In this section...

“Overview” on page 3-14

“Writing Differential Rate Equations as Rate Rules” on page 3-15

“Species” on page 3-15

“Parameters” on page 3-16

“Rate Rule 1 Cyclin B (CycB)” on page 3-17

“Rate Rule 2 M-Phase Promoting Factor (MPF)” on page 3-18

“Rate Rule 3 Inhibited M-Phase Promoting Factor (pMPF)” on page 3-18

“Rate Rule 4 Activated but Inhibited M-Phase Promoting Factor (pMPFp)” on page 3-19

“Rate Rule 5 Activated M-Phase Promoting Factor (MPFp)” on page 3-19

“Rate Rule 11 Activated Cdc25 (Cdc25p)” on page 3-19

“Rate Rule 12 Inhibited Wee1 (Wee1p)” on page 3-19

“Rate Rule 13 Activated Intermediate Enzyme (IEp)” on page 3-19

“Rate Rule 14 Activated APC (APCa)” on page 3-19

“Algebraic Rule 17 Rate Parameter K2” on page 3-20

“Algebraic Rule 18 Rate Parameter Kcdc25” on page 3-20

“Algebraic Rule 19 Rate Parameter Kwee1” on page 3-20

“Simulation of the M-Phase Control Model with Rules” on page 3-20

### Overview

There is one rate rule for each equation defining a species and one algebraic rule for each variable parameter in the M-phase control model (Marlovits 1998). For a list and description of the equations, see “M-Phase Control Equations” on page 3-4.

A basic model includes rate rules 1 through 5 and 11 through 14 with algebraic rules 17, 18, and 19.

## Writing Differential Rate Equations as Rate Rules

Writing differential rate equations in an unambiguous format that a software program can understand is a simple process when you follow the syntax rules for programming languages.

- Use an asterisk to indicate multiplication. For example,  $k[A]$  is written  $k*A$  or  $k*[A]$ . The brackets around the species  $A$  do not indicate concentration.
- SimBiology uses square brackets around species and parameter name to allow names that are not valid MATLAB® variable names. For example, you could have a species named `glucose-6-phosphate dehydrogenase` but you need to add brackets around the name in reaction rate and rule equations.

`[glucose-6-phosphate dehydrogenase]`

- Use parentheses to clarify the order of evaluation for mathematical operations. For example, do not write Henri-Michaelis-Menten reaction rates as  $V_m*C/K_d + C$ , because  $V_m*C$  is divided by  $K_d$  before adding  $C$  to the result. Instead, write this reaction rate as  $(V_m*C)/(K_d + C)$ .

## Species

The following table lists species in the model with their initial amounts. There are three variable parameters modeled as species ( $K_2$ ,  $K_{cdc25}$ , and  $K_{wee1}$ ). You could also model the variable parameters as parameters with the property **ConstantAmount** cleared.

Name	InitialAmount ▲	InitialAmountUnits	ConstantAmount
CycB	0.0		<input type="checkbox"/>
MPF	0.0		<input type="checkbox"/>
pMPF	0.0		<input type="checkbox"/>
pMPFp	0.0		<input type="checkbox"/>
MPFp	0.0		<input type="checkbox"/>
Cdc25p	0.0		<input type="checkbox"/>
Wee1p	0.0		<input type="checkbox"/>
IEp	0.0		<input type="checkbox"/>
APCa	0.0		<input type="checkbox"/>
Kcdc25	0.0		<input type="checkbox"/>
Kwee1	0.0		<input type="checkbox"/>
K2	0.0		<input type="checkbox"/>
Wee1	0.0		<input type="checkbox"/>
TotalCdc25	1.0		<input checked="" type="checkbox"/>
TotalWee1	1.0		<input checked="" type="checkbox"/>
TotalAPC	1.0		<input checked="" type="checkbox"/>
TotalIE	1.0		<input type="checkbox"/>
PPase	1.0		<input checked="" type="checkbox"/>
AntiAPC	1.0		<input checked="" type="checkbox"/>
Cdc2	100.0		<input checked="" type="checkbox"/>

### Parameters

The following table lists parameters in the model with their initial values. The property **ConstantValue** is selected for all of the parameters.

Name	Value ▾	ValueUnits	ConstantValue
Vweea	1.0		<input checked="" type="checkbox"/>
k1	1.0		<input checked="" type="checkbox"/>
Kmwr	1.0		<input checked="" type="checkbox"/>
Kmapr	1.0		<input checked="" type="checkbox"/>
Km25r	1.0		<input checked="" type="checkbox"/>
kcak	0.64		<input checked="" type="checkbox"/>
V2a	0.25		<input checked="" type="checkbox"/>
V25a	0.17		<input checked="" type="checkbox"/>
kier	0.15		<input checked="" type="checkbox"/>
kapr	0.13		<input checked="" type="checkbox"/>
kap	0.13		<input checked="" type="checkbox"/>
kwr	0.1		<input checked="" type="checkbox"/>
k25r	0.1		<input checked="" type="checkbox"/>
Kmw	0.1		<input checked="" type="checkbox"/>
Km25	0.1		<input checked="" type="checkbox"/>
kie	0.02		<input checked="" type="checkbox"/>
kw	0.02		<input checked="" type="checkbox"/>
k25	0.02		<input checked="" type="checkbox"/>
V25i	0.017		<input checked="" type="checkbox"/>
Vweei	0.01		<input checked="" type="checkbox"/>
Kmier	0.01		<input checked="" type="checkbox"/>
Kmie	0.01		<input checked="" type="checkbox"/>
Kmap	0.01		<input checked="" type="checkbox"/>
V2i	0.0050		<input checked="" type="checkbox"/>
k3	0.0050		<input checked="" type="checkbox"/>
kpp	0.0040		<input checked="" type="checkbox"/>

### Rate Rule 1 Cyclin B (CycB)

The rate rule is from “Equation 1 Cyclin B” on page 3-6.

$$\text{rate rule: CycB} = k1 - K2 * \text{CycB} - k3 * \text{Cdc2} * \text{CycB}$$

```
species: CycB = 0 nM
         Cdc2 = 100 nM, [x]constant
parameters: k1 = 1 nM/minute
            K2 = 0 1/minute, []constant
            k3 = 0.005 1/(nM*minute)
```

K2 is a variable rate parameter whose value is defined by an algebraic rule. See “Algebraic Rule 17 Rate Parameter K2” on page 3-20. Its value varies from 0.005 to 0.25 1/minute.

### **Rate Rule 2 M-Phase Promoting Factor (MPF)**

The rate rule is from “Equation 2 M-Phase Promoting Factor (MPF)” on page 3-6.

```
rate rule: MPF = kpp*MPFp - (Kwee1 + kcak + K2)*MPF + Kcdc25*pMPF
           + k3*Cdc2*CycB
species: MPF = 0 nM
        MPFp = 0 nM
        pMPF = 0 nM
parameters: kpp = 0.004 1/minute
            kcak = 0.64 1/minute
            k3 = 0.005 1/(nM*minute)
            K2 = 0 1/minute
            Kcdc25 = 0 1/minute
            Kwee1 = 0 1/minute
```

K2, Kcdc25, and Kwee1 are variable rate parameters whose values are defined by algebraic rules. See “Algebraic Rule 17 Rate Parameter K2” on page 3-20, “Algebraic Rule 18 Rate Parameter Kcdc25” on page 3-20, and “Algebraic Rule 19 Rate Parameter Kwee1” on page 3-20.

### **Rate Rule 3 Inhibited M-Phase Promoting Factor (pMPF)**

The rate rule is from “Equation 3 Inhibited M-Phase Promoting Factor (pMPF)” on page 3-7.

```
rate rule: pMPF = Kwee1*MPF - (Kcdc25 + kcak + K2)*pMPF + kpp*pMPFp
```

### Rate Rule 4 Activated but Inhibited M-Phase Promoting Factor (pMPFp)

The rate rule is from “Equation 4 Inhibited and Activated M-Phase Promoting Factor (pMPFp)” on page 3-8.

$$\text{rate rule: pMPFp} = \text{Kwee1} * \text{MPFp} - (\text{kpp} + \text{Kcdc25} + \text{K2}) * \text{pMPFp} + \text{kca} * \text{MPF}$$

### Rate Rule 5 Activated M-Phase Promoting Factor (MPFp)

The rate rule is from “Equation 5 Activated M-Phase Promoting Factor (MPFp)” on page 3-8.

$$\text{rate rule: MPFp} = \text{kca} * \text{MPF} - (\text{kpp} + \text{Kwee1} + \text{K2}) * \text{MPFp} + \text{Kcdc25} * \text{pMPFp}$$

### Rate Rule 11 Activated Cdc25 (Cdc25p)

The rate rule is from “Equation 11 Cell Division Control 25 (Cdc25)” on page 3-9.

$$\text{rate rule: Cdc25p} = (\text{k25} * \text{MPFp} * (\text{TotalCdc25} - \text{Cdc25p})) / (\text{Km25} + (\text{TotalCdc25} - \text{Cdc25p})) - (\text{k25r} * \text{PPase} * \text{Cdc25p}) / (\text{Km25r} + \text{Cdc25p})$$

### Rate Rule 12 Inhibited Wee1 (Wee1p)

The rate rule is from “Equation 12 Wee1 Activation/Deactivation” on page 3-10.

$$\text{rate rule: Wee1p} = (\text{kw} * \text{MPFp} * (\text{TotalWee1} - \text{Wee1p})) / (\text{Kmw} + (\text{TotalWee1} - \text{Wee1p})) - (\text{kwr} * \text{PPase} * \text{Wee1p}) / (\text{Kmwr} + \text{Wee1p})$$

### Rate Rule 13 Activated Intermediate Enzyme (IEp)

The rate rule is from “Equation 13 Intermediate Enzyme (IE) Activation/Deactivation” on page 3-10.

$$\text{rate rule: IEp} = (\text{kie} * \text{MPFp} * (\text{TotalIE} - \text{IEp})) / (\text{Kmie} + (\text{TotalIE} - \text{IEp})) - (\text{kier} * \text{PPase} * \text{IEp}) / (\text{Kmier} + \text{IEp})$$

### Rate Rule 14 Activated APC (APCa)

The rate rule is from “Equation 14 APC Activation/Deactivation” on page 3-11.

rate rule:  $APCa = (\text{kap} * IEp * (\text{TotalAPC} - APCa)) / (\text{Kmap} + (\text{TotalAPC} - APCa)) - (\text{kapr} * \text{AntiAPC} * APCa) / (\text{Kmapr} + APCa)$

### **Algebraic Rule 17 Rate Parameter K2**

K2 is a variable rate parameter whose value is determined by the amount of active and inactive APC. The algebraic rule is from “Equation 17 Rate Parameter K2” on page 3-11.

algebraic rule:  $V2i * (\text{TotalAPC} - APCa) + V2a * APCa - K2$   
 species:  $APCi = 1 \text{ nM}$   
 $APCa = 0 \text{ nM}$   
 $\text{TotalAPC} = 1 \text{ nM [x]constant}$   
 parameters:  $K2 = 0 \text{ or } 0.25 \text{ 1/minute, []constant}$   
 $V2i = 0.005 \text{ 1/(nM*minute)}$   
 $V2a = 0.25 \text{ 1/(nM*minute)}$

### **Algebraic Rule 18 Rate Parameter Kcdc25**

Kcdc25 is a variable rate parameter whose value is determined by the amount of active and inactive Cdc25. The algebraic rule is from “Algebraic Rule 18 Rate Parameter Kcdc25” on page 3-20.

algebraic rule:  $V25i * (\text{TotalCdc25} - Cdc25p) + V25a * Cdc25p - Kcdc25$

### **Algebraic Rule 19 Rate Parameter Kwee1**

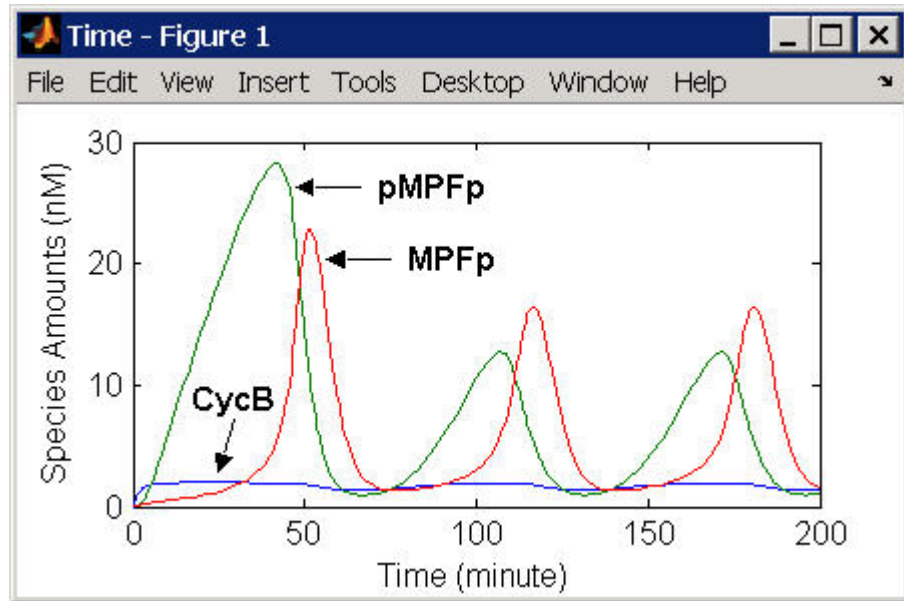
Kwee1 is a variable rate parameter whose value is determined by the amount of active and inactive Wee1. The algebraic rule is from “Equation 19 Rate Parameter Kwee1” on page 3-12.

algebraic rule:  $Vweei * Wee1p + Vweea * (\text{TotalWee1} - Wee1p) - Kwee1$

### **Simulation of the M-Phase Control Model with Rules**

This is a simulation of the M-phase control model using rate and algebraic rules. Simulate with the ode15s solver and plot species CycB, pMPFp, and MPFp. For a description of the model, see “SimBiology Model with Rate and Algebraic Rules” on page 3-14.





If you want to run the simulation, you can open the model within the SimBiology desktop.

- 1** Open the SimBiology desktop using the function `sbiodesktop`.
- 2** From the **File** menu, select **Open Project**.
- 3** Browse to the file `m_phase_xenopus.sbproj` in the following directory:

```
matlabroot/toolbox/simbio/simbiodemos/m_phase_xenopus.sbproj
```

For a model using reactions and algebraic rules, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

## SimBiology Model with Reactions and Algebraic Rules

### In this section...

“Overview” on page 3-23

“Reaction 1 Synthesis of Cyclin B ” on page 3-23

“Reaction 2 Degradation of Cyclin B ” on page 3-24

“Reaction 3 Dimerization of Cyclin B with Cdc2 Kinase” on page 3-25

“Reaction 4 Degradation of Cyclin B on MPF” on page 3-26

“Reaction 5 Deactivation of Active MPF” on page 3-27

“Reaction 6 Activation of MPF” on page 3-28

“Reaction 7 Remove Inhibiting Phosphate from Inhibited MPF” on page 3-29

“Reaction 8 Inhibition of MPF by Phosphorylation” on page 3-30

“Reaction 11 Degradation of Cyclin B on Inhibited MPF” on page 3-32

“Reaction 12 Deactivation of MPF to Inhibited MPF” on page 3-32

“Reaction 13 Activation of Inhibited MPF” on page 3-32

“Reaction 15 Degradation of Cyclin B on Active but Inhibited MPF” on page 3-33

“Reaction 16 Inhibit MPF by Phosphorylation” on page 3-33

“Reaction 17 Remove Inhibiting Phosphate from Activated MPF” on page 3-34

“Reaction 19 Degradation of Cyclin B on Activated MPF” on page 3-34

“Reaction 36 Activation of Cdc25 by Activated MPF” on page 3-34

“Reaction 37 Deactivation of Cdc25” on page 3-35

“Reaction 38 Deactivation of Wee1 by Active MPF” on page 3-35

“Reaction 39 Activation of Wee1” on page 3-35

“Reaction 40 Activation of Intermediate Enzyme by Active MPF ” on page 3-36

“Reaction 41 Deactivation of IE” on page 3-36

“Reaction 42 APC Activation by IEp” on page 3-36

**In this section...**

“Reaction 43 APC Deactivation” on page 3-36

“Block Diagram of the M-Phase Control Model with Reactions” on page 3-37

“Simulation of the M-Phase Control Model with Reactions” on page 3-39

**Overview**

There can be one or more reactions for an equation defining a species and one algebraic rule for each variable parameter in the M-phase control model (Marlovits 1998). For a list and description of the equations, see “M-Phase Control Equations” on page 3-4.

A basic model includes reactions 1 through 8, 11 through 13, 15 through 17, 19, and 36 through 43 with algebraic rules from equations 17, 18, and 19.

**Reaction 1 Synthesis of Cyclin B**

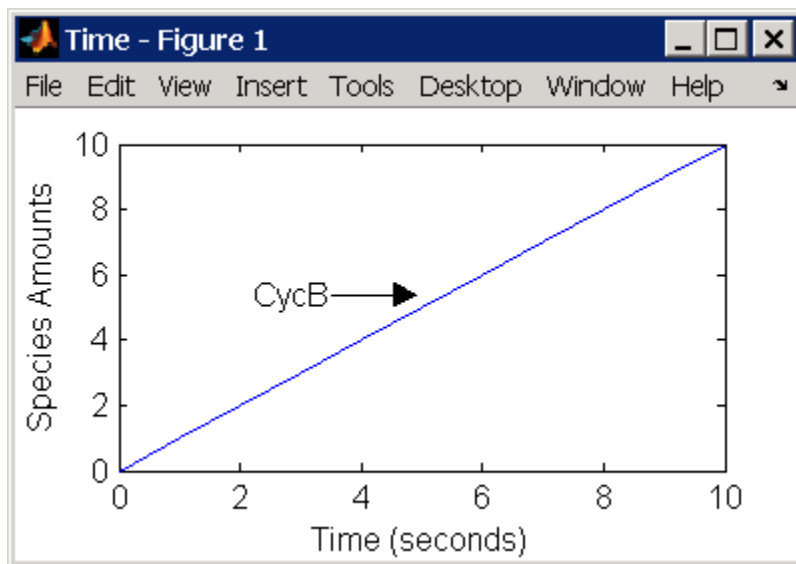
Cyclin B is synthesized at a constant rate.

```

reaction: AA -> CycB
reaction rate: k1 nM/minute
parameter: k1 = 1 nM/minute
species: CycB = 0 nM
          AA = 100 nM [x]constant [x]boundary

```

Simulate reaction 1 with the ode15s solver.



## Reaction 2 Degradation of Cyclin B

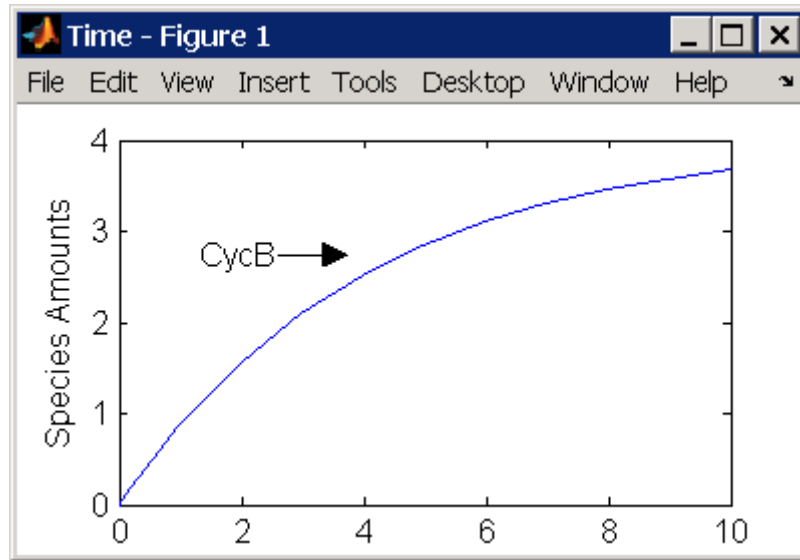
Cyclin B is degraded at the end of the M-phase.

```

reaction: CycB -> AA
reaction rate: K2*CycB nM/minute
parameters: K2 = 0 1/minute, []constant, variable by rule
            V2i = 0.005 1/nM*minute
            V2a = 0.25 1/nM*minute
species: CycB = 0 nM
         APCi = 1 nM
         APCa = 0 nM
         AA = 100 nM [x]constant [x]boundary
algebraic rule: (V2i*APCi) + (V2a*APCa) - K2
    
```

Initially, Cyclin B degradation is low. This implies the amount of active APC (APCa) = 0 and inactive APC (APCi) = APCtotal = 1 nM.

Test the algebraic rule by simulating reactions 1 and 2 with APCi = 0 and APCa = 1.



### Reaction 3 Dimerization of Cyclin B with Cdc2 Kinase

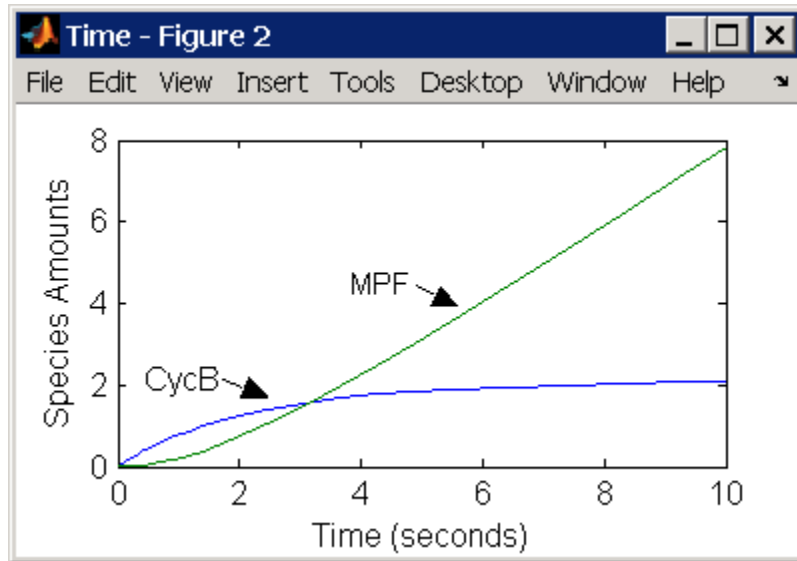
Cyclin B dimerizes with Cdc2 kinase to form M-phase promoting factor (MPF).

```

reaction: Cdc2 + CycB -> MPF
reaction rate: k3*Cdc2*CycB nM/minute
parameters: k3 = 0.005 1/(nM*minute)
species: Cdc2 = 100 nM
          CycB = 0 nM
          MPF = 0 nM

```

Test the model by simulating with  $K2 = 0.25$ .



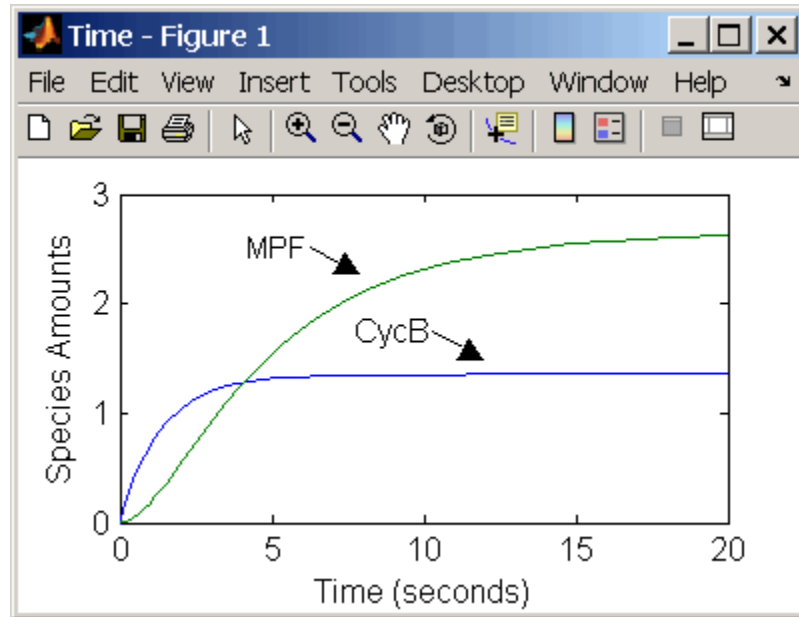
### Reaction 4 Degradation of Cyclin B on MPF

Cyclin B is tagged with ubiquitin groups and degrades while bound to Cdc2.

```

reaction: MPF -> Cdc2 + AA
reaction rate: K2*[MPF]
parameters: K2 = 0 or 0.25 1/minute, variable by rule
             v2i = 0.005 1/(nM*minute)
             v2a = 0.25 1/(nM*minute)
species: MPF = 0 nM
          APCi = 1 nM
          APCa = 0 nM
          AA = 100 nM [x]constant [x]boundary
algebraic rule: (v2i*APCi) + (v2a*APCa) - K2
    
```

Test the simulation with  $APCa = 1$  and  $APCi = 0$ . Because the amount of  $APCa$  (active) is high,  $K2$  increases and the degradation starts to balance the synthesis of MPF.



### Reaction 5 Deactivation of Active MPF

Active MPF (MPFp) is dephosphorylated on Thr-161 by an unknown phosphatase (PP) to inactive MPF (MPF).

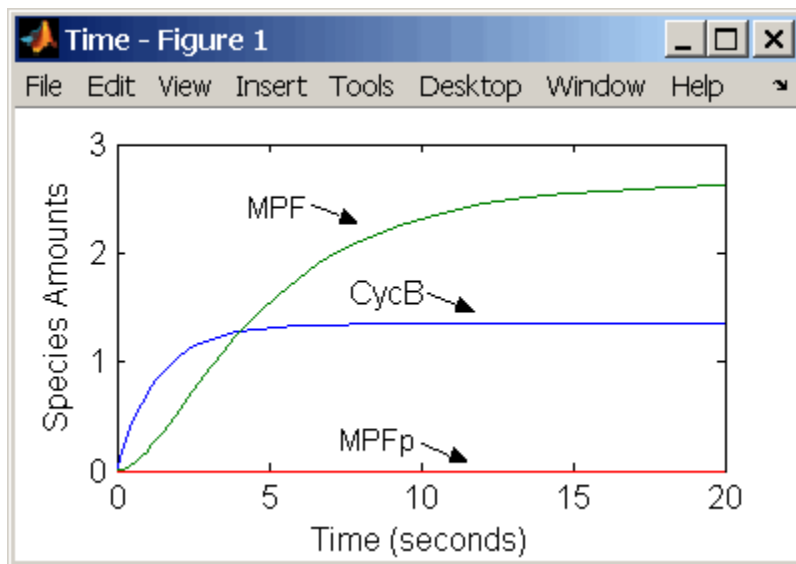
```

reaction: MPFp -> MPF
reaction rate: kpp*[MPFp]
parameters: kpp = 0.004 1/minute
species: MPFp = 0 nM
          MPF = 0 nM

```

$k_{cacr} = 0.004$  1/minute (Marlovits 1998, p 175), but is renamed to  $k_{pp}$  (Borisuk 1998).

Test simulation with  $APCa = 1$  and  $APCi = 0$ . MPF increases without reaching steady-state.



### Reaction 6 Activation of MPF

Inactive MPF (MPF) is phosphorylated on Thr-161 by an unknown cyclin activating kinase (CAK).

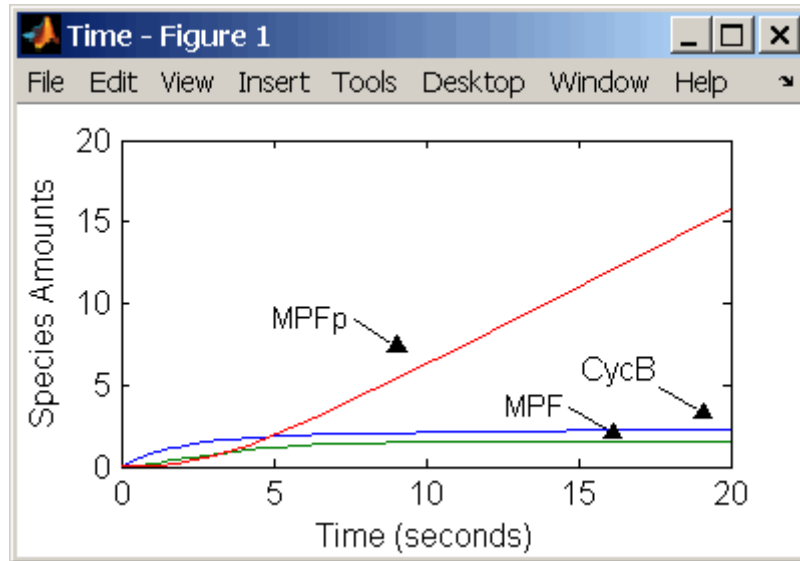
```

reaction: MPF -> MPFp
reaction rate: kcak*[MPF]
parameters: kcak = 0.64 1/minute
species: MPF = 0 nM
          MPFp = 0 nM
    
```

The kinase reaction that phosphorylates MPF to the active form is 160 times faster than the phosphatase reaction that dephosphorylates active MPF.

Simulate the model with reactions 1 through 6. Notice that after adding reaction 6, most of the product goes to active MPF (MPFp).





## Reaction 7 Remove Inhibiting Phosphate from Inhibited MPF

Cdc25 phosphatase removes the inhibiting phosphate groups at the threonine 14 and tyrosine 15 residues on Cdc2 kinase.

```

reaction: pMPF -> MPF
reaction rate: Kcdc25*[pMPF]
parameters: Kcdc25 = 0.0 1/minute or 0.017 1/minute, variable by
              algebraic rule
              V25i = 0.017 1/(mM*minute)
              V25a = 0.17 1/mM*minute
species: pMPF = 0 nM
         MPF = 0 nM
         Cdc25 = 1 nM (inactive)
         Cdc25p = 0 nM (active)
algebraic rule: (V25i*Cdc25) + (V25a*Cdc25p) - Kcdc25

```

Initially, all of the Cdc25 phosphatase is in the inactive form (Cdc25).

Enter the initial value for Kcdc25 as 0.0 and let the first time step calculate the value from the rule, or enter an initial value using the rule.

Initially, set **ConstantAmount** for Cdc25 and Cdc25p to test reactions 1 through 7. Then after you can add the reactions to regulate the Cdc25 phosphatase by clearing the **ConstantAmount** property.

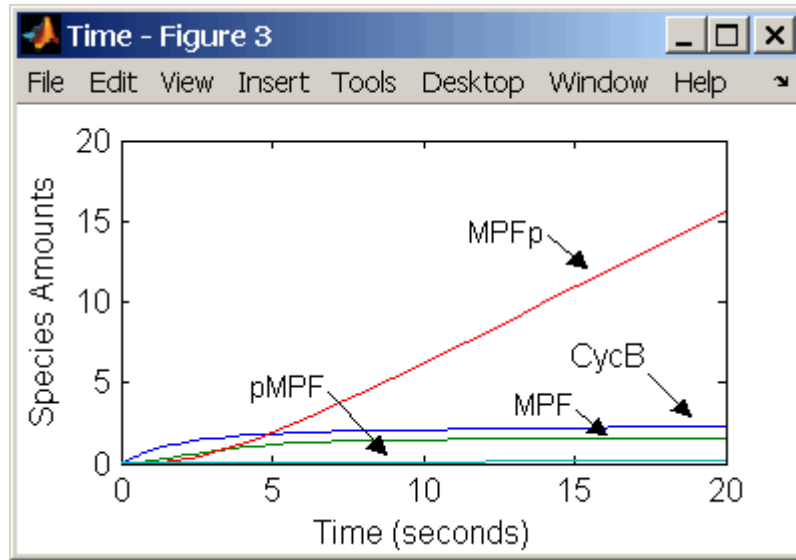
### **Reaction 8 Inhibition of MPF by Phosphorylation**

Addition of inhibiting phosphate groups by Wee1 kinase to inhibit active M-phase promoting factor (MPF). Myt1 kinase is also involved with the phosphorylation, but its contribution is grouped with Wee1.

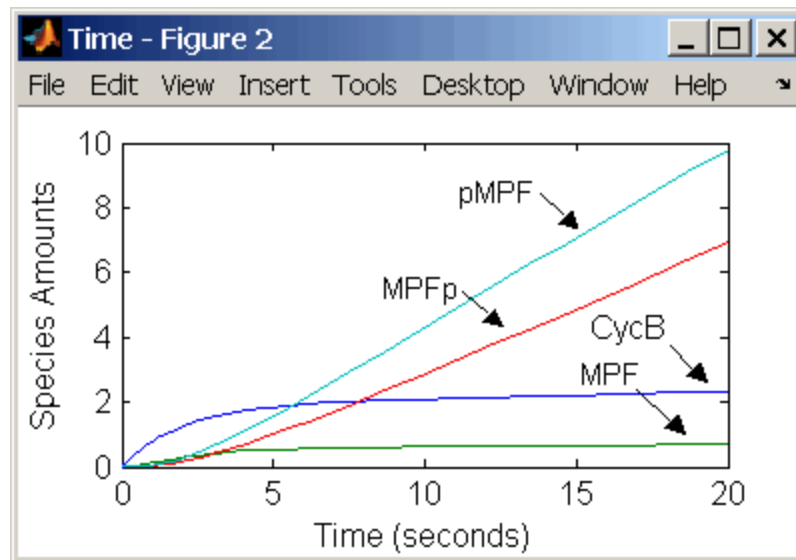
```
reaction: MPF -> pMPF
reaction rate: Kwee1*[MPF]
parameters: Kwee1 = 0.0 1/minute or 0.01 1/minute, variable by
              algebraic rule
              Vwee1i = 0.01 1/(nM*minute)
              Vwee1a = 1.0 1/(nM*minute)
species: MPF = 0 nM
         pMPF = 0 nM
         Wee1p = 1 nM (inactive)
         Wee1 = 0 nM (active)
algebraic rule: (Vwee1i*Wee1p) + (Vwee1a*Wee1) - Kwee1
```

The initial capitalization for the parameter Kwee1 is a convention to indicate that this value changes during the simulation.

Test the simulation for reactions 1 through 8 with Wee1p (inactive) = 1 and Wee1 (active) = 0.



Test the simulation with Wee1p (inactive) = 0 and Wee1 (active) = 1.



### Reaction 11 Degradation of Cyclin B on Inhibited MPF

Degradation of Cyclin B (CycB) on inhibited MPF (pMPF). Cyclin B is tagged with ubiquitin groups and degrades while bound to Cdc2.

```

reaction: pMPF -> Cdc2 + AA
reaction rate: K2*[pMPF] nM/minute
parameters: K2 = 0 or 0.25 1/minute, variable by rule
            V2i = 0.005 1/nM*minute
            V2a = 0.25 1/nM*minute
species: MPF = 0 nM
         APCi = 1 nM
         APCa = 0 nM
         AA = 100 nM [x]constant [x]boundary
         Cdc2 = 100 nm
algebraic rule: (V2i*APCi) + (V2a*APCa) - K2
    
```

Test the simulation with Wee1 active (Wee1 = 1) and APC active (APCi = 1).

### Reaction 12 Deactivation of MPF to Inhibited MPF

Inhibited/Active MPF (pMPFp) is dephosphorylated on Thr-161 by an unknown phosphatase (PP) to inhibited MPF (pMPF). Compare reaction 12 with reaction 5.

```

reaction: pMPFp -> pMPF
reaction rate: kpp*[pMPFp]
parameters: kpp = 0.004 1/minute
species: pMPFp = 0 nM
         pMPF = 0 nM
    
```

### Reaction 13 Activation of Inhibited MPF

Inhibited MPF (pMPF) is phosphorylated on Thr - 161 by an unknown cyclin activating kinase (CAK). Compare reaction 13 with reaction 6.

```

reaction: pMPF -> pMPFp
reaction rate: kcak*[pMPF] nM/minute
parameters: kcak = 0.64 1/minute
species: pMPF = 0 nM
         pMPFp = 0 nM
    
```

Test the simulation with Wee1p = 1 (inactive)/ Wee1 = 0 and then test with Wee1p = 0 (inactive)/ Wee1 = 1.

## Reaction 15 Degradation of Cyclin B on Active but Inhibited MPF

Degradation of Cyclin B (CycB) on inhibited MPF (pMPF). Cyclin B is tagged with ubiquitin groups and degrades while bound to Cdc2 kinase.

```

reaction: pMPFp -> Cdc2 + AA
reaction rate: K2*[pMPFp] nM/minute
parameters: K2 = 0 or 0.25 1/minute, variable by rule
             v2i = 0.005 1/nM*minute
             v2a = 0.25 1/nM*minute
species: MPF = 0 nM
          APCi = 1 nM
          APCa = 0 nM
          AA = 100 nM [x]constant [x]boundary
          Cdc2 = 100 nm
algebraic rule: (V2i*APCi) + (V2a*APCa) - K2
    
```

## Reaction 16 Inhibit MPF by Phosphorylation

Addition of inhibiting phosphate groups by Wee1 kinase to inhibit active M-phase promoting factor (MPF). Myt1 kinase is also involved with the phosphorylation, but its contribution is grouped with Wee1.

```

reaction: MPFp -> pMPFp
reaction rate: Kwee1*[MPFp] nM/minute
parameters: Kwee1 = 1/minute [x]constant, variable by rule
             Vweei = 0.01 1/nM*minute
             Vweea = 1 1/nM*minute
species: MPFp = 0 nM
          pMPFp = 0 nM
          Wee1p = 1 nM (inactive)
          Wee1 = 0 nM (active)
algebraic rule: (Vweei*Wee1p) + (Vweea*Wee1) - Kwee1
    
```

### Reaction 17 Remove Inhibiting Phosphate from Activated MPF

Remove the inhibiting phosphate group from pMPFp with Cdc25 phosphatase.

```

reaction: pMPFp -> MPFp
reaction rate: Kcdc25*[pMPFp]
parameters: Kcdc25 = 0 1/minute, [x]constant, variable by rule
            V25i = 0.017 1/nM*minute
            V25a = 0.17 1/nM*minute
species: pMPFp = 0 nM
         MPFp = 0 nM
algebraic rule: (V25i*Cdc25) + (V25a*Cdc25p) - Kcdc25
    
```

### Reaction 19 Degradation of Cyclin B on Activated MPF

Degradation of Cyclin B (CycB) on inhibited MPF (pMPF). Cyclin B is tagged with ubiquitin groups and degrades while bound to Cdc2 kinase.

```

reaction: MPFp -> MPF + AA
reaction rate: K2*[MPFp] nM/minute
parameters: K2 = 0 or 0.25 1/minute, variable by rule
            V2i = 0.005 1/nM*minute
            V2a = 0.25 1/nM*minute
species: MPF = 0 nM
         MPFp = 0 nM
         APCi = 1 nM
         APCa = 0 nM
         AA = 100 nM [x]constant [x]boundary
         Cdc2 = 100 nm
algebraic rule: (V2i*APCi) + (V2a*APCa) - K2
    
```

### Reaction 36 Activation of Cdc25 by Activated MPF

Activation of Cdc25 phosphatase by phosphorylation with active M-phase promoting factor (MPFp).

```

reaction: Cdc25 + (MPFp) -> Cdc25p + (MPFp)
reaction rate: (k25*[MPFp]*[Cdc25])/(Km25 + [Cdc25])
parameters: k25 = 0.02 1/minute
    
```

```

Km25 = 0.1 nM
species: Cdc25 = 1 nM (inactive)
        Cdc25p = 0 nM (active)

```

Initially MPF is inhibited (MPF\* reacts to pMPF\*).

### Reaction 37 Deactivation of Cdc25

Deactivation of Cdc25 phosphatase by dephosphorylation with an unknown phosphatase.

```

reaction: Cdc25p -> Cdc25
reaction rate: (k25r*[Cdc25p])/(Km25r + [Cdc25p])
parameters: k25r = 0.1 nM/minute
            Km25r = 1 nM
species: Cdc25 = 1 nM (inactive)
        Cdc25p = 0 nM (active)

```

### Reaction 38 Deactivation of Wee1 by Active MPF

Deactivation of Wee1 kinase by phosphorylation with active M-phase promoting factor (MPFp).

```

reaction: Wee1 + (MPFp) -> Wee1p + (MPFp)
reaction rate: (kw*[MPFp]*[Wee1])/(Kmw + [Wee1]) nM/minute
parameters: kw = 0.02 1/minute
            Kmw = 0.1 nM
species: Wee1p = 1 nM (inactive)
        Wee1 = 0 nM (active)

```

Initially MPF is inhibited (MPF\* reacts to pMPF\*).

### Reaction 39 Activation of Wee1

Activation of Wee1 kinase by dephosphorylation with an unknown kinase.

```

reaction: Wee1p -> Wee1
reaction rate: (kwr*[Wee1p])/(Kmwr + [Wee1p]) nM/minute
parameters: kwr = 0.1 nM/minute
            Kmwr = 1 nM
species: Wee1p = 1 nM (inactive)

```

Wee1 = 0 nM (active)

### **Reaction 40 Activation of Intermediate Enzyme by Active MPF**

The inactive intermediate enzyme (IE) is activated by phosphorylation with active M-phase promoting factor (MPFp).

```
reaction: IE + (MPFp) -> IEp + (MPFp)
reaction rate: (kie*[MPFp]*[IE])/(Kmie + [IE])
parameters: kie = 0.02 1/minute
             Kmie = 0.01nM
species: IE = 1 nM (inactive)
         IEp = 0 nM (active)
```

### **Reaction 41 Deactivation of IE**

The active intermediate enzyme (IE) is deactivated by dephosphorylation.

```
reaction: IEp -> IE
reaction rate: (kier*[IEp])/(Kmier + [IEp])
parameters: kier = 0.15 nM/minute
             Kmier = 0.01 nM
species: IE = 1 nM (inactive)
         IEp = 0 nM (active)
```

### **Reaction 42 APC Activation by IEp**

Anaphase promoting complex (APC) is activated by an active intermediate enzyme (IEp).

```
reaction: APCi + IEp -> APCa + IEp
reaction rate: (kap*[IEp]*[APCi])/(Kmap + [APCi])
parameters: kap = 0.13 1/minute
             Kmap = 0.01 nM
species : APCi = 1 nM
         APCa = 0 nM
```

### **Reaction 43 APC Deactivation**

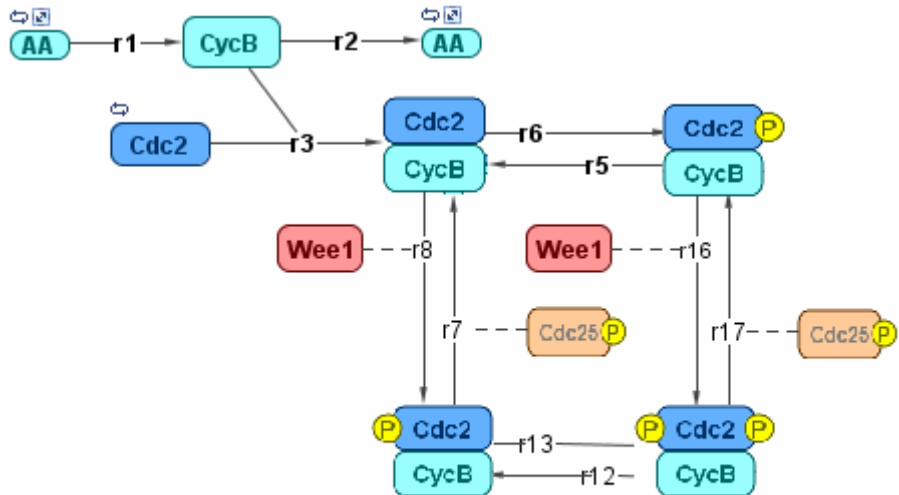
Anaphase promoting complex (APC) is deactivated.

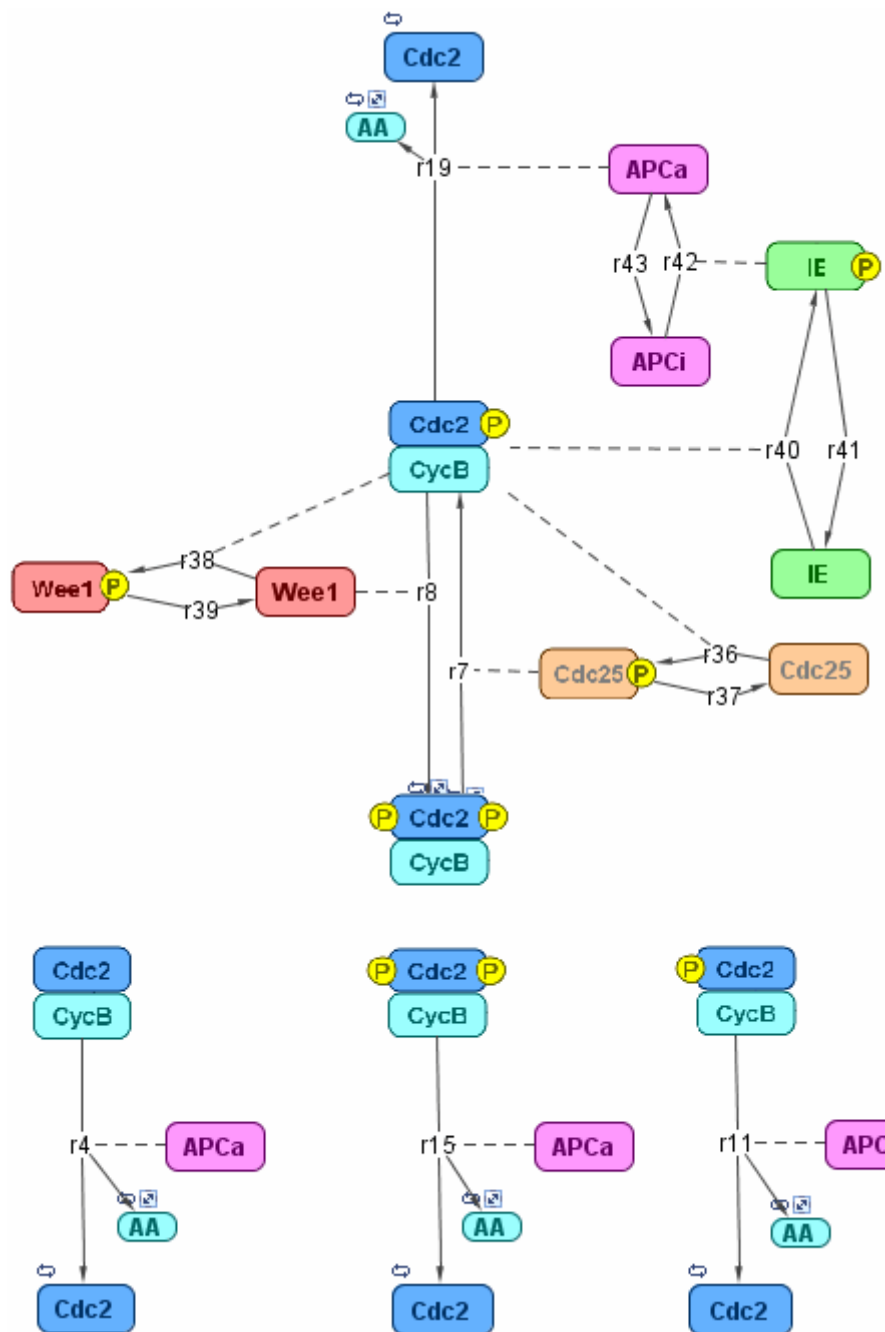
```
reaction: APCa -> APCi
```



reaction rate:  $(k_{apr} * [APCa]) / (K_{mapr} + [APCa])$   
 parameters:  $k_{apr} = 0.13$  nM/minute  
 $K_{mapr} = 1$  nM  
 species :  $APCi = 1$  nM  
 $APCa = 0$  nM

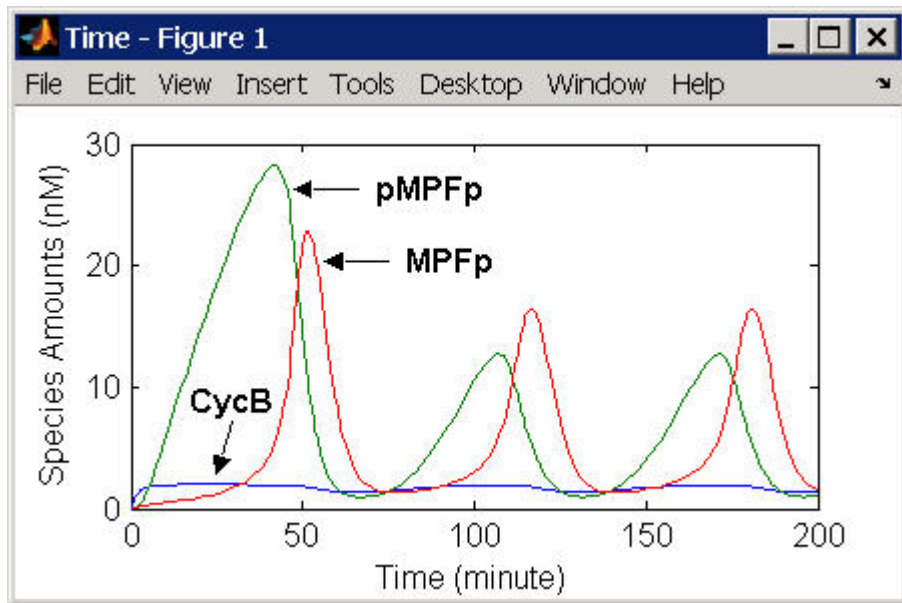
### Block Diagram of the M-Phase Control Model with Reactions





## Simulation of the M-Phase Control Model with Reactions

This is a simulation of the M-phase control model using reactions and algebraic rules. Simulate with the `ode15s` solver and plot species `CycB`, `pMPFp`, and `MPFp`. For a description of the model, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.



If you want to run the simulation, you can open the model within the SimBiology Desktop.

- 1 Open the Desktop using the function `sbiodesktop`.
- 2 From the **File** menu, select **Open Project**.
- 3 Browse to the file `m_phase_xenopus.sbproj` in the following directory

```
matlabroot/toolbox/simbio/simbiodemodemos/m_phase_xenopus.sbproj
```

For a model using reactions and algebraic rules, see “SimBiology Model with Reactions and Algebraic Rules” on page 3-22.

## References

[1] Borisuk M, Tyson J (1998), “Bifurcation analysis of a model of mitotic control in frog eggs,” *Journal of Theoretical Biology*, 195(1):69–85, PubMed 9802951.

[2] Marlovits G, Tyson C, Novak B, Tyson J (1998), “Modeling M-phase control in *Xenopus* oocyte extracts: the surveillance mechanism for unreplicated DNA,” *Biophysical Chemistry*, 72(1-2):169–184, PubMed 9652093.

[3] Novák B, Tyson J (1993), “Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos,” *Journal of Cell Science*, 106(4):1153–1168, PubMed 8126097.

## G

- G protein cycle
  - background 2-4
  - in yeast 2-4
  - model 2-1
  - modeling of 2-6
- G protein cycle model
  - experimental data and assumptions 2-6
  - reactions overview 2-6
- G proteins
  - introduction 2-4

## M

- models
  - G protein cycle 2-1

## R

- references
  - yeast G protein cycle model 2-52

## Y

- Yeast G protein cycle model
  - add reaction 2-13
  - change simulation settings of 2-30
  - comparison of
    - active G protein fractions 2-45

- copy and change 2-37
- creating a rule 2-28
- custom plot
  - dashed line 2-41
- determine rate equation 2-15
- experimental and simulation data
  - comparison of 2-48
  - custom plot 2-48
- experimental data and assumptions 2-6
- graphical representation 2-6
- in SimBiology 2-11
- introduction 2-1 2-6
- mutant strain 2-41
- open SimBiology desktop 2-12
- parameters table 2-24
- project file 2-13
- reactions overview 2-6
- reactions table 2-21
- references 2-52
- setting species initial amounts 2-19
- simulation results
  - sst2*  $\Delta$  2-43
  - wild-type 2-34
- species table 2-25
- tutorial goals 2-11
- wild-type strain 2-11