SimBiology

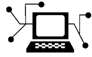
For Use with MATLAB®

- Computation
- Visualization
- Programming

Model Reference

Version 1.0

How to Contact The MathWorks:

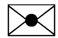


`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup

`support@mathworks.com`	Technical Support
`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 Phone

508-647-7001 Fax



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

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

SimBiology Model Reference

© COPYRIGHT 2005 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, and xPC TargetBox are registered 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` for more information.

Revision History

September 2005	Online only	New for Version 1.0 (Release 14SP3+)

Contents

1 Minimal Cascade Model for a Mitotic Oscillator

Goldbeter Model **1-2**
Graphic Representation **1-3**
Reaction Descriptions and Model Assumptions **1-4**
Mathematical Model **1-4**

SimBiology Model with Rate Rules **1-6**
SimBiology Model with Rules **1-6**
SimBiology Simulation with Rules **1-9**

Simbiology Model with Reactions **1-10**
Converting Differential Rate Equations to Reactions **1-10**
Calculating Initial Values for Reactions **1-12**
SimBiology Simulation with Reactions **1-20**

References **1-21**

2 Model of the Yeast Heterotrimeric G Protein Cycle

Background On G Protein Cycles **2-3**
G Proteins **2-3**
G Proteins and Pheromone Response **2-4**

Modeling a G Protein Cycle **2-5**
Reactions Overview **2-5**
Assumptions, Experimental Data, and Units in the G Protein Model **2-7**

Building the G Protein Cycle Model in SimBiology **2-10**

Goals of the Tutorial **2-10**
Opening the SimBiology Desktop **2-11**
Saving Your Work as a SimBiology Project File **2-12**
Adding a Reaction to the SimBiology Model **2-12**
Determining the Reaction Rate Equation **2-14**
Setting Initial Amounts of Species **2-17**

Completing the SimBiology Model **2-19**
Reactions **2-19**
Parameters **2-20**
Species **2-21**
Creating a Rule for the G Protein Model **2-25**

Simulating the G Protein Cycle Model in SimBiology .. **2-26**
Setting Conditions Before Simulating the G Protein Model **2-26**
Simulation Results for Model of the Wild-Type Strain **2-29**

Building a Model for the Mutant Strain **2-32**
Copying a Model **2-32**
Changing Model Settings **2-33**

Simulating the G Protein Cycle Model of the Mutant Strain **2-34**
Creating a Custom Plot **2-34**
Plotting Using a Custom Plot **2-35**
Simulation Results for the Model of the Mutant Strain ... **2-37**

Plotting Species from Two Different Data Sets **2-39**
Plotting Active G Protein Fraction from Model of Wild-Type Strain **2-39**
Creating a Custom Plot to Compare the Data **2-40**
Plotting Active G Protein Fraction from Model of the Mutant Strain **2-41**

Plotting Experimental Data with Simulation Data **2-43**
Creating a Custom Plot for Experimental Data **2-43**
Plotting the Data **2-44**

References **2-46**

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

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.

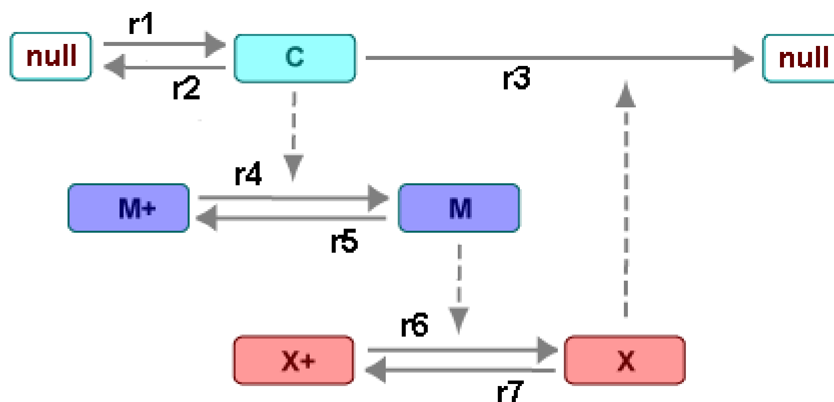
Graphic Representation (p. 1-3)	Diagram showing the reactions and relationships between species.
Reaction Descriptions and Model Assumptions (p. 1-4)	Brief descriptions of the reactions with some of the simplifying assumptions
Mathematical Model (p. 1-4)	Biochemical pathways defined with differential rate equations and algebraic equations.

Graphic Representation

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+ -> M) and protease activation (X+ -> 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 (r1) and degraded at a constant rate (r2).
- Cyclin (C) does not complex with cdc2 kinase (M).
- Cyclin (C) activates cdc2 kinase (M+ -> M) by increasing the velocity of the phosphatase that activates the kinase. Inactive cdc2 kinase (M+) is activated by removing inhibiting phosphate groups (r4).
- 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 (r5).
- The activation of cyclin protease (X+ -> X) by the active Cdc2 kinase (M) is direct without other intervening cascades. Cyclin protease (X) is activated by adding phosphate groups (r6).
- 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 (r7).
- 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).

$$\frac{dC}{dt} = vi - vdX\frac{C}{Kd + C} - kdC$$

Differential Rate Equation 2, Kinase (M)

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

$$\frac{dM}{dt} = V1\frac{(1-M)}{K1+(1-M)} - V2\frac{M}{K2+M}$$

$$V1 = \frac{VM1[C]}{Kc+[C]}$$

Differential Rate Equation 3, Protease (X)

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

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

$$V3 = VM3[M]$$

SimBiology Model with Rate 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.

SimBiology Model with Rules (p. 1-6)	Equivalent rate and algebraic rules for the equations
SimBiology Simulation with Rules (p. 1-9)	Model simulation with a deterministic solver

SimBiology Model with Rules

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 Michaelis-Menten rate as `Vm*C/Kd + C`, because `Vm*C` is divided by `Kd` 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 = vi - (vd*X*C)/(Kd + C) - kd*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 table below is a list of species in the model with their initial amounts.

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

Name	InitialAmount	ConstantAmount	BoundaryCondition
C	0.01	☐	☐
M	0.01	☐	☐
Mplus	0.99	☐	☐
Mt	1.0	☐	☐
X	0.01	☐	☐
Xplus	0.99	☐	☐
Xt	1.0	☐	☐
V1	0.0	☐	☐
V3	0.0	☐	☐

Parameters

The table below 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	☑
kd	0.01	☑
vd	0.25	☑
Kd	0.02	☑
VM1	3.0	☑
K1	0.0050	☑
Kc	0.5	☑
V2	1.5	☑
K2	0.0050	☑
VM3	1.0	☑
K3	0.0050	☑
V4	0.5	☑
K4	0.0050	☑

Rules

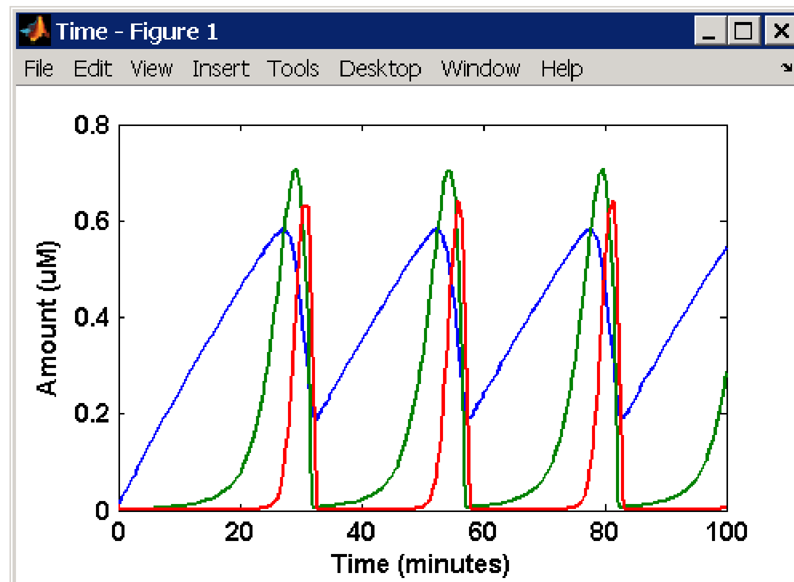
The active (`M`) and inactive (`Mplus`) forms of the kinase are assumed to be part of a conserved cycle with the total concentration (Mt) remaining constant during the simulation. You need only one differential rate equation with a mass balance algebraic equation to define the amounts of both species. 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 `dX/dt = Expression` as `X = Expression`, and algebraic rules of the form `"X = Expression"` where X is the independent variable, as `Expression - X`.

Name	Rule	RuleType
Cyclin	C = vi - (vd*X*C)/(Kd + C) - kd*C	rate
Kinase	M = (V1*Mplus)/(K1 + Mplus) - (V2*M)/(K2 + M)	rate
	Mt - Mplus - M	algebraic
	(VM1*C)/(Kc + C) - V1	algebraic
Protease	X = (V3*Xplus)/(K3 + Xplus) - (V4*X)/(K4 + X)	rate
	Xt - Xplus - X	algebraic
	VM3*M - V3	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

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

Converting Differential Rate Equations to Reactions (p. 1-10)	Simple rules for converting differential rate equations to reactions
Calculating Initial Values for Reactions (p. 1-12)	Calculations for initial values with different units within consistent dimensions
SimBiology Simulation with Reactions (p. 1-20)	Model simulation with a deterministic solver

Converting Differential Rate Equations to Reactions

Some models are defined with differential rate equations, and you might need the reactions to be compatible with your model. Some rules for converting differential rate equations to reactions are

- **For a positive term** — The species described by the equation is placed on the right as a product, 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 -> ?` with an unknown product. The positive term identifies the product and completes the reaction, `R <-> 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.

$$\frac{dC}{dt} = vi - vdX\frac{C}{Kd + C} - kdC$$

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

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

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

r3      reaction: C -> null
   reaction rate: (vd*X*C)/(Kd + 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.

$$\frac{dM}{dt} = V1\frac{(1-M)}{K1 + (1-M)} - V2\frac{M}{K2 + M}$$

$$V1 = \frac{VM1[C]}{Kc + [C]}$$

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*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*[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*Xplus]/(K3 + Xplus)
  algebraic rule: V3 = VM3*M

r7       reaction: X -> Xplus
   reaction rate: V4*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 Goldbeter 1991 paper.
- B — Units of concentration with time converted to second. When converting a to b, use `1 minute = 60 second` for parameters.

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

- 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.

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

- D — units of amount as molecules. When converting amount as moles to molecules, use `6.022e23 molecules = 1 mole`.

$$\frac{\text{Z mole}}{\text{second}} \text{ x } \frac{\text{6.022e23 molecule}}{\text{1 mole}} = \frac{\text{N molecules}}{\text{second}}$$

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 Degradataion

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

R3		Value	Units
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

R4		Value	Units
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

R4		Value	Units
		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

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

R6		Value	Units
		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

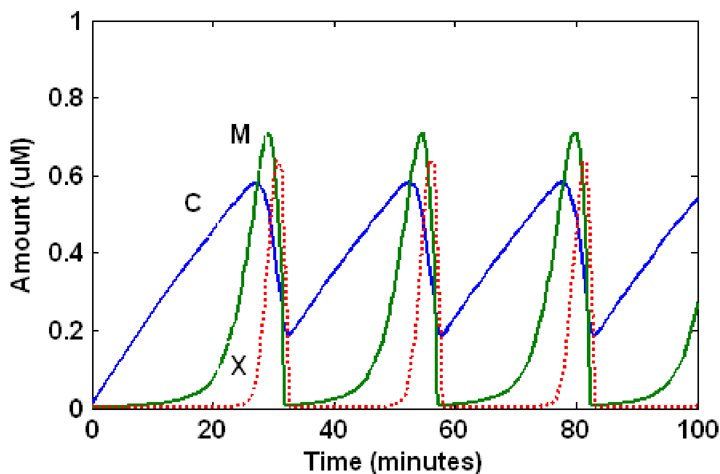
Reaction 7 Protease Deactivation

R7		Value	Units
reaction	`X -> 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 AW, Kirschner MW (1989), “Cyclin synthesis drives the early embryonic cell cycle,” Nature, 339:275-280.

2

Model of the Yeast Heterotrimeric G Protein Cycle

SimBiology enables you to 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.

Background On G Protein Cycles (p. 2-3)	Brief overview of G proteins and the G protein cycle in the yeast pheromone response pathway
Modeling a G Protein Cycle (p. 2-5)	Overview of reactions and assumptions
Building the G Protein Cycle Model in SimBiology (p. 2-10)	Describes the goals of this tutorial and explains how to build the model in SimBiology
Completing the SimBiology Model (p. 2-19)	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. 2-26)	Determine the settings for simulation and visualize the simulation results

Building a Model for the Mutant Strain (p. 2-32)	Create a model for the mutant strain, (*sst2*Δ) by copying and changing the existing model
Simulating the G Protein Cycle Model of the Mutant Strain (p. 2-34)	Plot the simulation results for the mutant strain model using a custom plot
Plotting Species from Two Different Data Sets (p. 2-39)	Plot species data from two different simulation runs
Plotting Experimental Data with Simulation Data (p. 2-43)	Store experimental data and use custom plotting to plot experimental and simulation data together
References (p. 2-46)	Reference material used in this tutorial

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.

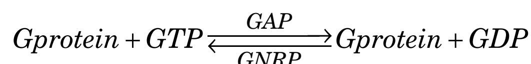
Background On G Protein Cycles

This section is a brief overview of G proteins and the G protein cycle in the yeast pheromone response pathway.

G Proteins (p. 2-3)	Introduction to G Proteins
G Proteins and Pheromone Response (p. 2-4)	Yeast G protein cycle

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)].



$$Gprotein + GTP \underset{GNRP}{\overset{GAP}{\rightleftarrows}} Gprotein + GDP$$

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 sertonin 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 α-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 α-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.

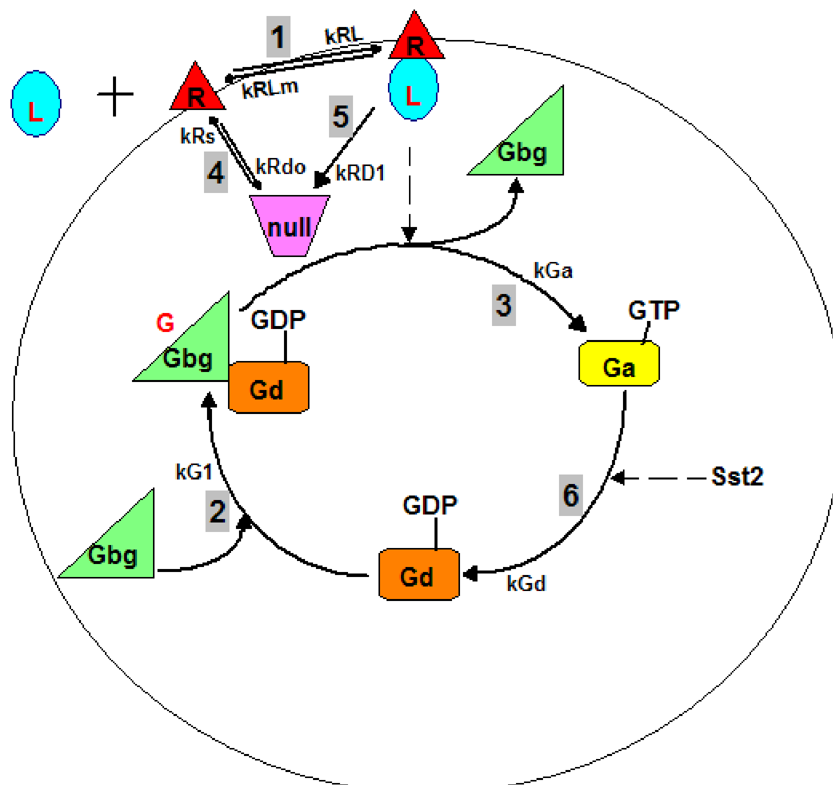
Modeling a G Protein Cycle

Systems biologists represent biological pathways and processes as reactions with reaction rates and treat the components of these pathways as individual species. In this example, α-factor, α-factor receptor, and the G protein subunits are all treated as species participating in reactions.

Reactions Overview (p. 2-5)	Table of reactions and rate parameters
Assumptions, Experimental Data, and Units in the G Protein Model (p. 2-7)	Description of assumptions, and experimental data from the reference paper

Reactions Overview

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. 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 = G-alpha-GDP, G = inactive Gbg:Gd complex, null = sink or source for degradation or synthesis

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 <-> RL	kRLm, kRL
2	Heterotrimeric G protein formation	Gd + Gbg -> G	kG1
3	G protein activation	RL + G -> Ga + Gbg + RL	kGa
4	Receptor synthesis and degradation	R <-> null	kRdo, kRs
5	Receptor-Ligand degradation	RL -> null	kRd1
6	G protein inactivation	Ga -> 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 $0.11\ s^{-1}$) in the Sst2p containing wild-type strain. The uncatalyzed reaction rate is estimated to be $0.004\ s^{-1}$ 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 Ga (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

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.

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 2-26 for the sections on the simulation and analysis of these models.

Goals of the Tutorial (p. 2-10)	Description of the salient features of the tutorial
Opening the SimBiology Desktop (p. 2-11)	Describes how to open the SimBiology graphical user interface.
Saving Your Work as a SimBiology Project File (p. 2-12)	Save your work in the SimBiology project format
Adding a Reaction to the SimBiology Model (p. 2-12)	Add a reaction and determine the reaction rate equation
Determining the Reaction Rate Equation (p. 2-14)	Assign a kinetic law and parameters to determine the reaction rate equation
Setting Initial Amounts of Species (p. 2-17)	Assign the initial values for species amounts

Goals of the Tutorial

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 (*sst2Δ* , 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, go to **Help > SimBiology Desktop Help** to access 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
   ```

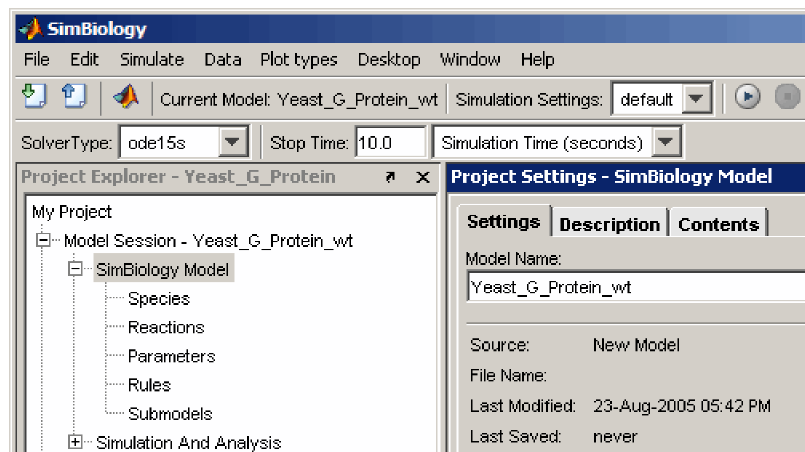
   The **SimBiology Desktop** opens. Use the **Project Explorer** in the left pane to navigate.

2. In the **Project Explorer**, click **SimBiology Model**.

3. In the **Project Settings-SimBiology Model** pane, click the **Settings** tab.

4. In the **Name** box, enter the name for your model.

   ```
   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 enable you to 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 opens.

2. Enter a name for the project file, and then click **Save**.

```
yeast_g_protein_cycle.sbproj
```

Adding a Reaction to the SimBiology Model

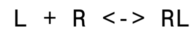
The next steps show you how to add a reaction and determine the reaction rate equation for your model.

This example shows you the first reaction.

Name	Reaction	Rate Parameters
Receptor-Ligand Interaction	L + R <-> RL	kRLm, kRL

1 In the **Project Explorer** click **Reactions**.

2 In the **Project Settings-Reactions** pane, enter the reaction in the **Enter Reaction** box, and click **Add**.



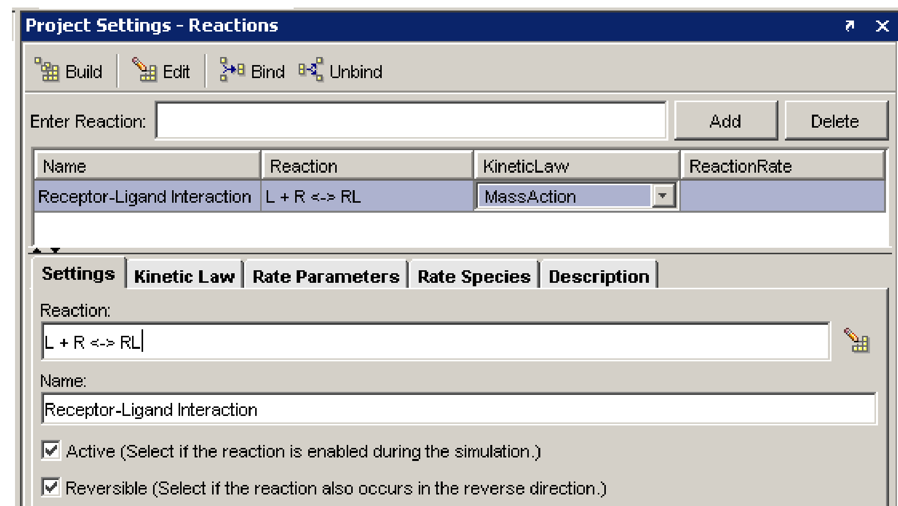
```
L + R <-> RL
```

3 In the **Name** box, enter 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:



Note 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 checked by default.

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 <-> 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

SimBiology populates the reaction rate column once 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, select 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.

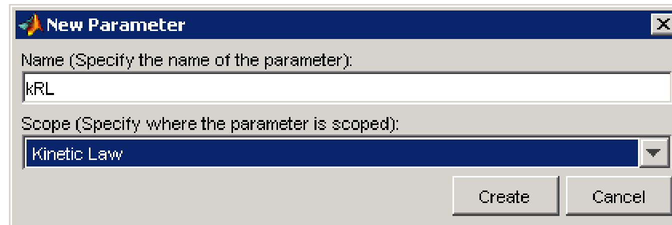
   If you set the scope of a parameter to `Model`, you can use this parameter in rules, other reactions, and submodels. All the parameters for this model are scoped to their respective kinetic laws.

   #### More about scoping

   Scoping enables you to overload parameters; you can have parameters with the same name at kinetic law and at model level. Because SimBiology 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 submodels use the parameter value at the model level.

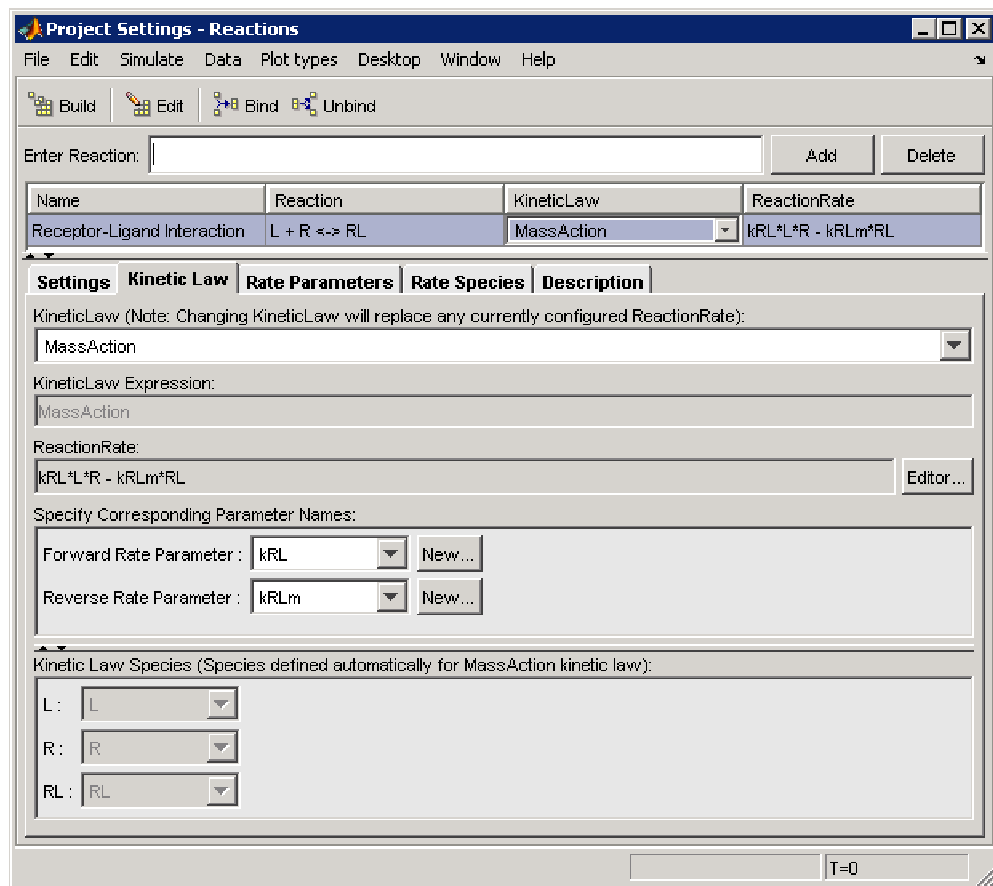
A reaction uses a parameter at the model level, only if a corresponding parameter is not defined at the kinetic law level for the reaction.



5 Click **Create**.

Repeat steps 2 through 5 to create the **Reverse Rate Parameter**. Call the parameter `kRLm`.

Notice that SimBiology populates the ReactionRate column.

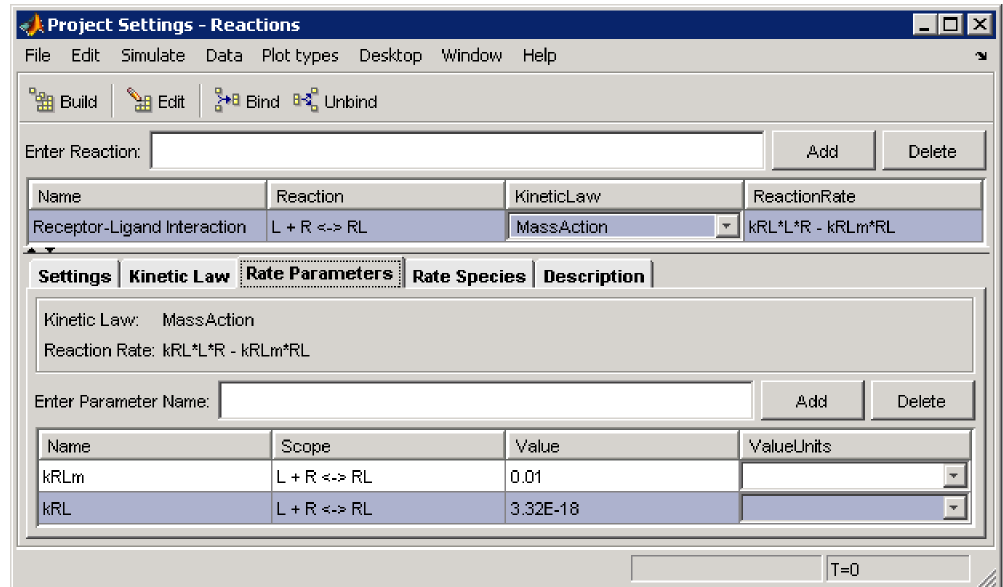


The species are automatically selected when you `MassAction` kinetic law.

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

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

`kRLm = 0.01` and `kRL = 3.32E-18`.



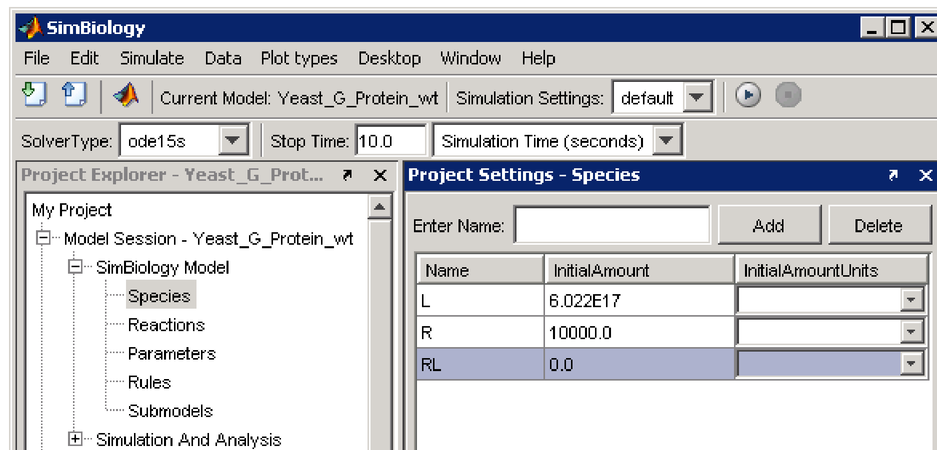
Setting Initial Amounts of Species

SimBiology automatically adds species for you when you add a reaction. You can set the initial amounts of all the model species in the **Project Settings-Species** pane.

Name	InitialAmount
L	6.022E17
R	10000.0
RL	0.0

1. Select the **Species** node in the **Project Explorer**. SimBiology updates the **Project-Settings** pane to show the species spreadsheet.

2. In the **Project Settings-Species** pane, enter the preceding values in the **Initial Amount** cell.



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

Completing the SimBiology Model

In the previous sections of this tutorial show you how to the 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].

Reactions (p. 2-19)	Table of reactions used in the model
Parameters (p. 2-20)	Table of parameters and values used in the model
Species (p. 2-21)	Table of species and species initial amounts used in the model
Creating a Rule for the G Protein Model (p. 2-25)	Create a rule in the model to specify the ratio of active G protein

Reactions

Add reactions 2 through 6 listed in the table below, 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 2-14.

No.	Name	Reaction	Rate Parameters
1	Receptor-Ligand Interaction	`L + R <-> RL`	`kRLm`, `kRL`
2	Heterotrimeric G protein formation	`Gd + Gbg -> G`	`kG1`
3	G protein activation	`RL + G -> Ga + Gbg + RL`	`kGa`
4	Receptor synthesis and degradation	`R <-> null`	`kRdo`, `kRs`

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:

Project Settings - Reactions

Build Edit Bind Unbind

Enter Reaction: Add Delete

Name	Reaction	KineticLaw	ReactionRate
Receptor-ligand interaction	L + R <-> RL	MassAction	kRL*L*R - kRLm*RL
Heterotrimeric Gprotein formation	Gd + Gbg -> G	MassAction	kG1*Gd*Gbg
Gprotein activation	RL + G -> Ga + Gbg + RL	MassAction	kGa*RL*G
Receptor synthesis-degradation	R <-> null	MassAction	kRdo*R - kRs
Receptor-ligand degradation	RL -> null	MassAction	kRD1*RL
Gprotein inactivation	Ga -> Gd	MassAction	kGd*Ga

Parameters

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

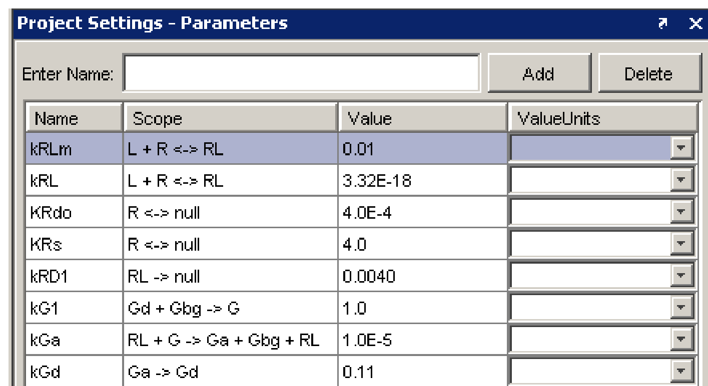
Alternatively, you can set all the parameters in the **Project Settings-Parameters** pane. Click the **Parameters** node in the **Project Explorer** to access the **Project Settings-Parameters** pane.

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).



Species

Set the species amounts in the **Project Settings-Species** pane. In the **Project Explorer**, click the **Species** node to access this pane.

- Note that the species participating in the reactions have been added automatically 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` (α-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
Ga	0.0
RL	0.0

In order to replicate the published results, SimBiology 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 (Gα-GTP) and `Gt` is the total amount of G protein in a cell. This assignment is made using a rule in SimBiology, and the procedure to create this rule is described in the next section (“Creating a Rule for the G Protein Model” on page 2-25).

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

Additional Species

Name	InitialAmount (molecule)
Ga_frac	0.0
Gt	10000.0

To add a new species,

1. In the **Project Explorer**, select the **Species** node to access the **Project Settings-Species** pane.

2. In the **Enter name** box, enter the name of a new species, and then click **Add** or press **Enter**.

   ```
   Ga_frac
   ```

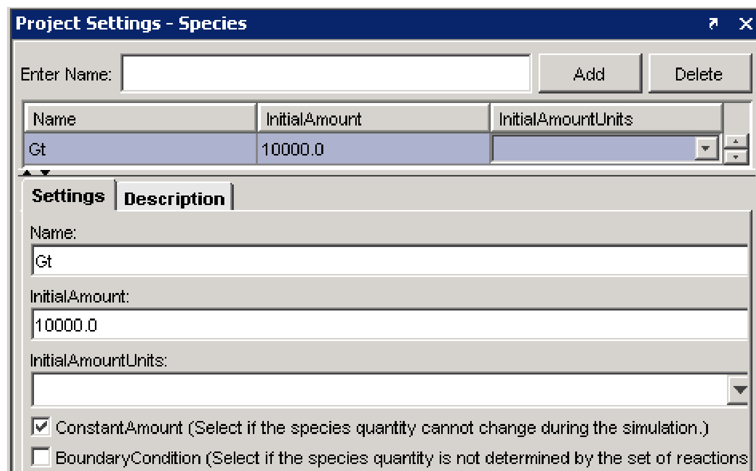
   The species table updates with the new entry and its row selected. Note that the species is now available in the **Settings** tab.

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

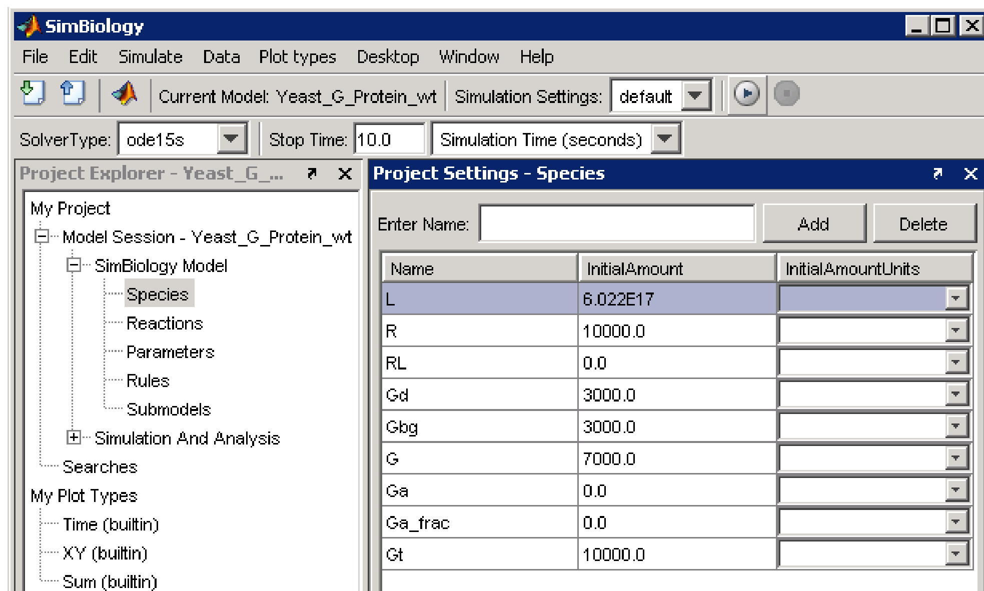
   ```
   0.0
   ```

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

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



Your species table should now look similar to this:



For additional help in each procedure, go to **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 Click the **Rules** node in the **Project Explorer** to access the **Project Settings-Rules** pane.

2 In the **Enter Rule** box, type the expression, and then click **Add** or, press **Enter**.

```
Ga_frac-Ga/Gt
```

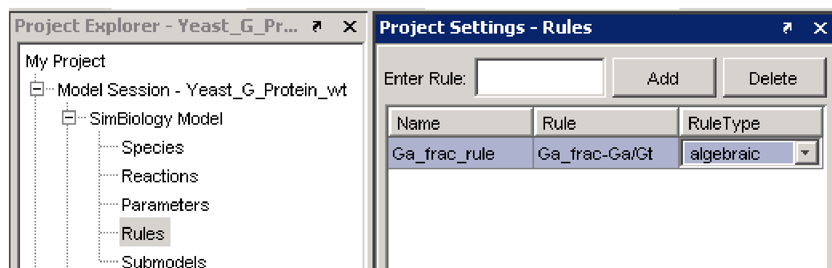
The rule is entered as an expression.

3 From the **RuleType** list, select

```
algebraic
```

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

```
Ga_frac_rule
```



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

Simulating the G Protein Cycle Model in SimBiology

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.

Setting Conditions Before Simulating the G Protein Model (p. 2-26)	Setting conditions for simulation in the **Simulation Settings** pane
Simulation Results for Model of the Wild-Type Strain (p. 2-29)	Simulation results for the G protein cycle model in the wild-type strain

Setting Conditions Before Simulating the G Protein Model

Consider the following points about simulating this model:

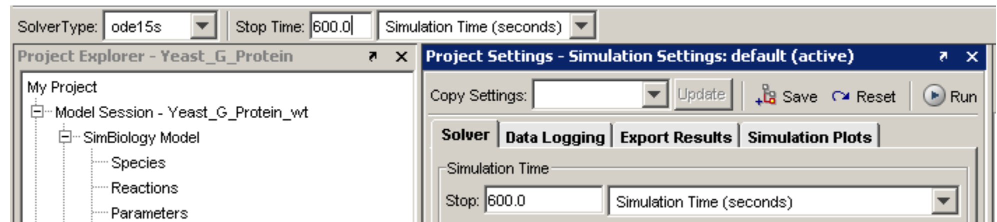
- Yi., et al. show data up to `600s` for the active G protein time course. To replicate these results, change the simulation settings from the default `10 second` to `600 second`. This change remains active for this model unless you change it back to the default.
- The ligand species `'L'` has values 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 2-28).

Changing Simulation Stop Time

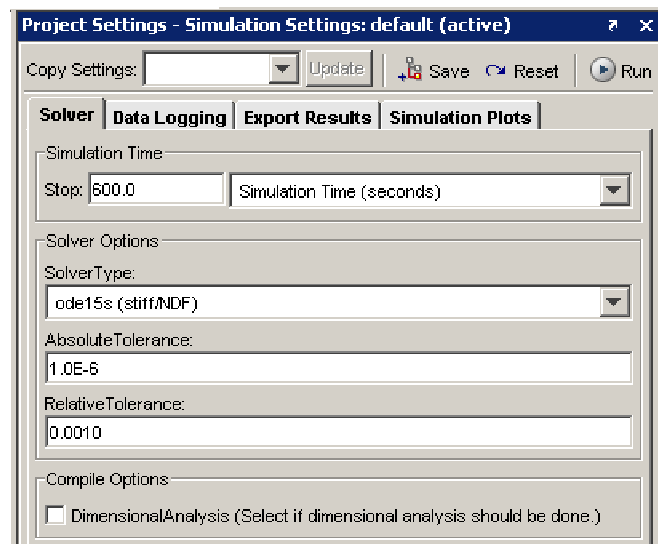
Change the stop time to replicate the simulation used in the reference paper [Yi et al.2003], and to facilitate comparison with the experimental results presented in the study.

1 Change the time in the Simulation toolbar to `600 second` and press **Enter**. SimBiology applies the new time setting.



2 Expand the **Simulation and Analysis** node in the **Project Explorer** and click **Simulation Settings**.

3 In the **Project Settings-Simulation Settings** pane, select the **Solver** tab. (You can also apply your time settings in the **Simulation Time** box.)



4 Leave the solver type as the default **SolverType**, `ode15s(stiff/NDF)`, and leave the **AbsoluteTolerance** and **RelativeTolerance** values as the default.

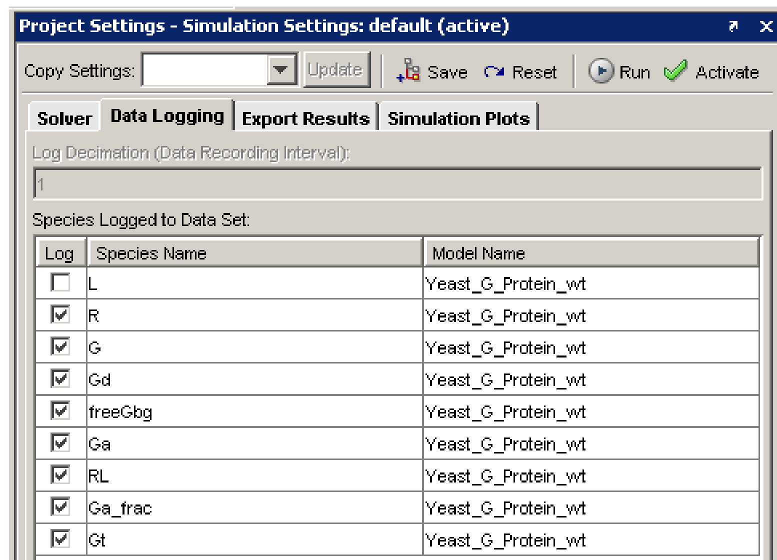
Specifying Which Data Is Recorded by SimBiology

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.

1 Expand the **Simulation and Analysis** node in the **Project Explorer**, and click **Simulation Settings** to access the **Project Settings-Simulation Settings** pane.

2 Select the **Data Logging** tab.

3 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'`.

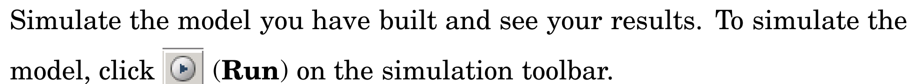


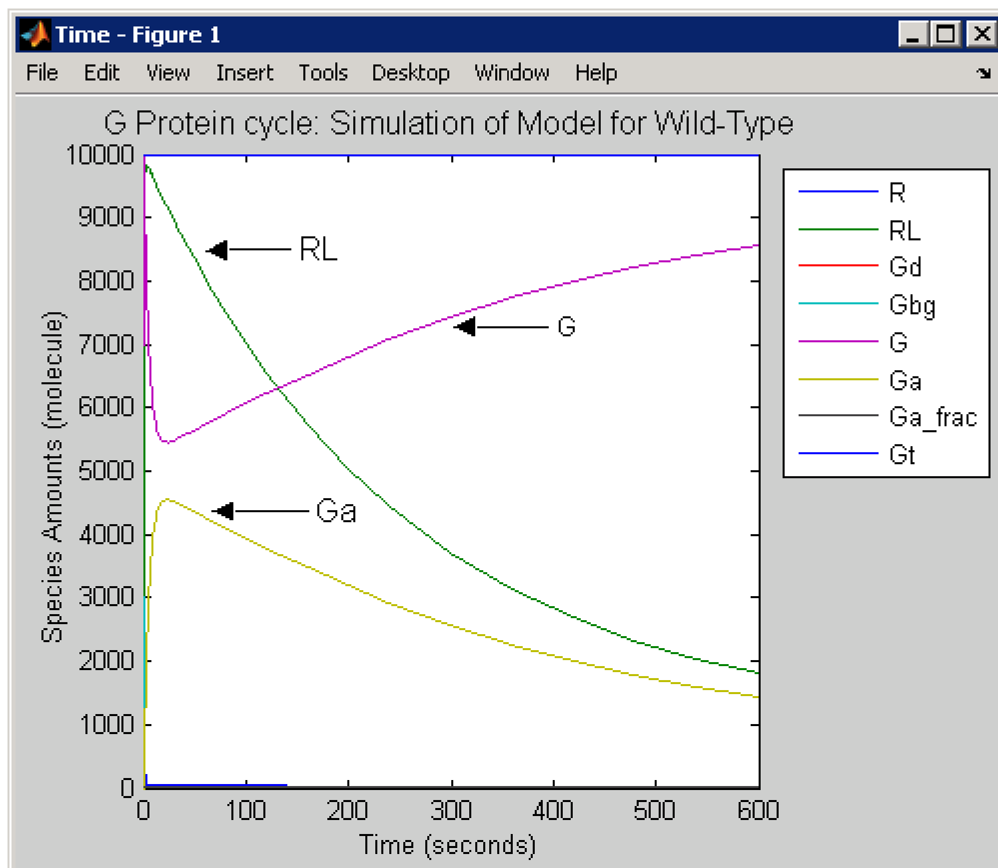
Alternatively, if you choose to log the data for all the species, you can use the **Plot Browser** to select the subset of species you want to view in your plot. This is described in "Viewing a Subset of Species in the Figure Window" on page 2-31.

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 the **Simulation Settings** node for more help on this topic. To access context-sensitive help go to **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. 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 Model of the Wild-Type Strain

Simulate the model you have built and see your results. To simulate the model, click  (**Run**) on the simulation toolbar.



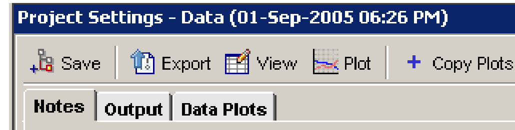
To view all species see "Viewing a Subset of Species in the Figure Window" on page 2-31 for help on using the **Plot Browser**.

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**, click the **Data** node for the wild-type model.

2 Click **Save**.



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

```
wt_model_run1
```

The saved data is added under the **Data** node in the **Project Explorer**.

Viewing a Subset of Species in the Figure Window

The **Plot Browser** enables you to view all or a subset of the species. This is useful when you log (record) the data for all the species in the model simulation, but you want to view only some of the species.

1 On the figure window, from the **View** menu, select **Plot Browser**. The figure window updates with a view of the plotted Y variables.

2 Select the variables to view.

To skip to the section on simulating the second model, see "Simulating the G Protein Cycle Model of the Mutant Strain" on page 2-34.

Building a Model for the Mutant Strain

The deletion in SST2, results in uncatalyzed G protein deactivation (Reaction 6; `Ga -> Gd`). 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 to reflect the change in the rate of the reaction, and simulate the new model.

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 (p. 2-32)	Copy a model by importing project contents
Changing Model Settings (p. 2-33)	Change the settings in the new model to reflect the pathway in the *sst2*Δ strain

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 2-12.

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 Load SimBiology Project dialog opens.

2. Browse and select the project.

   ```
   yeast_g_protein_cycle.sbproj
   ```

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

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 nodes in the *newly* created model session.

- Change the name of the model. Click the **SimBiology Model** node and select the **Settings** tab.

  ```
  Yeast_G_Protein_mut
  ```

- Change the value of `kGd` from `0.11` to `0.004`. Click the **Parameters** node and edit in the parameters table.

  ```
  0.004
  ```

Simulating the G Protein Cycle Model of the Mutant Strain

In addition to generating the plot for the model representing the G protein cycle in the *sst2Δ* strain, this section describes how to create and use a custom plot for the data generated in the model simulation.

Creating a Custom Plot (p. 2-34)	Create a custom plot to show simulation results with dashed lines
Plotting Using a Custom Plot (p. 2-35)	Add the custom plot to simulation plots, select the species to be plotted, and plot the results
Simulation Results for the Model of the Mutant Strain (p. 2-37)	Plot the simulation results for the model of the mutant (*sst2Δ*) strain

Creating a Custom Plot

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 In the **Project-Explorer**, select the **My Plot Types** node.

2 In the **Project Settings-My Plot Types** pane, point to the **Copy Plot Type** list and select `Time`.

3 Click **Update**.

The **My Plot Types** command window updates:

Plot Code with Input Arguments: (tobj, y)

```
1  % Get the simulation data associated with the species
2  % specified in y.
3  if strcmpi(y, '<all>')
4      [time, data, names] = sbiogetnamedstate(tobj);
5  else
6      [time, data, names] = sbiogetnamedstate(tobj, y);
7  end
8
9  % Error checking.
10 if size(data,2) == 0
11    error('Species specified do not exist.');
12 end
13
14 % Plot.
15 hLine = plot(time, data);
```

4 Make the following change to `hline = plot (time,data);` (line 15):

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

5 Click **Save**.

The Save Plot Type dialog box opens.

6 Enter a name for the custom plot type and click **Save**.

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

```
Time plot with dashed line
```

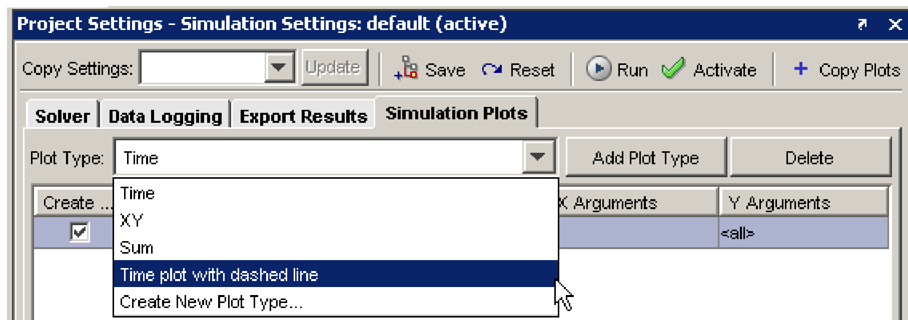
Plotting Using a Custom Plot

To plot the simulation data for the *sst2Δ* model with dashed lines,

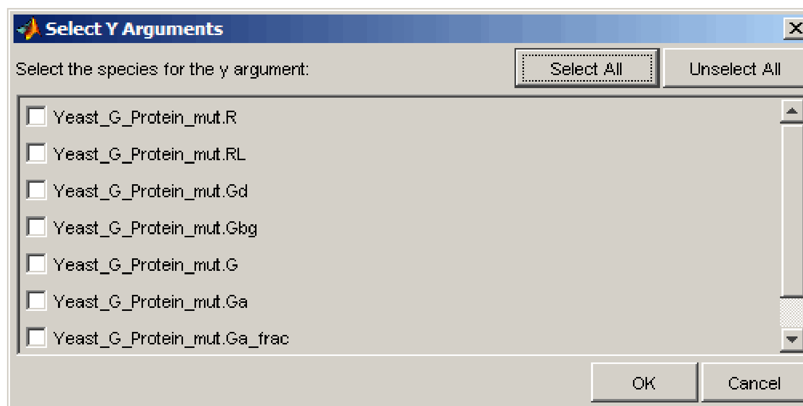
1 Expand the Simulation and Analysis node for this model and select **Simulation Settings: default (active)**.

2 In the **Project Settings-Simulation Settings: default (active)** pane, select the **Simulation Plots** tab.

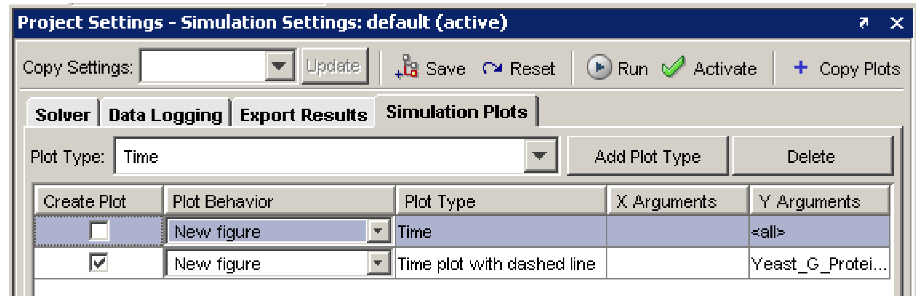
3 From the **Plot Type** list, select the new custom plot and click the **Add Plot Type** button.



4 In the plot type table, double-click the **Y Arguments** cell for the new entry. This opens the Select Y Arguments dialog box. Click the **Select All** button and then click **OK**.



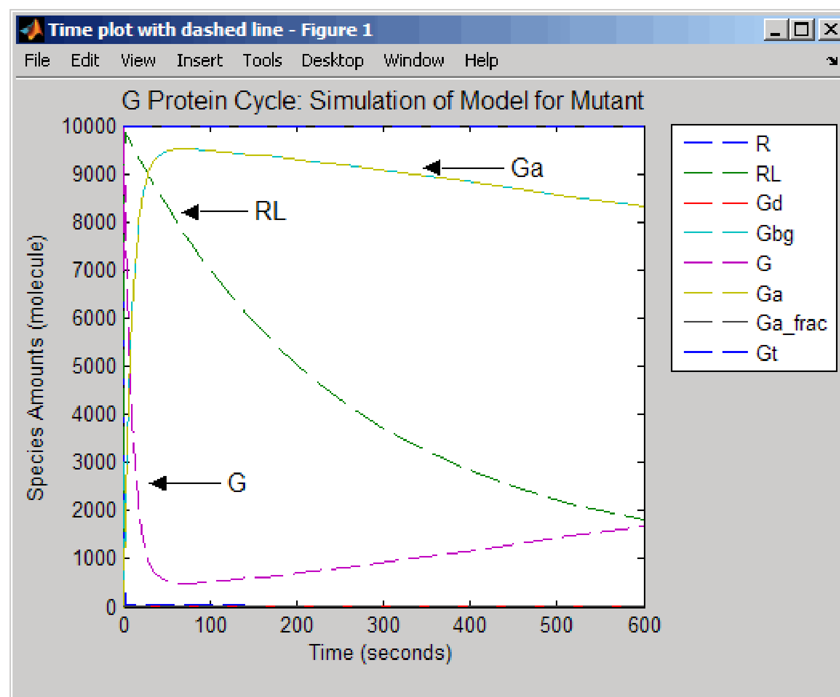
5 In the plot types table, clear the **Create Plot** check box for the `Time` plot.



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, click (**Run**) on the simulation toolbar.



To view each species in the figure window, see "Viewing a Subset of Species in the Figure Window" on page 2-31

The simulation results for the wild-type strain are described in "Simulation Results for Model of the Wild-Type Strain" on page 2-29.

Plotting Species from Two Different Data Sets

This section shows you how to compare the active G protein fractions in the two simulations. This example shows you how to plot the data without having to rerun the simulation, and more about how to generate custom plots.

The procedures in this section show you how to create a plot showing species from the two data sets in this tutorial. This example uses the data from the most recent simulation run. Start by plotting the active G protein fraction from the model of wild-type strain. Then, leave the generated figure window open and plot active G protein fraction from the model of the *sst2Δ* strain in the same plot.

Plotting Active G Protein Fraction from Model of Wild-Type Strain (p. 2-39)	Plot active G protein fraction simulation data for the wild-type strain
Creating a Custom Plot to Compare the Data (p. 2-40)	Create a custom plot specifying the species to plot
Plotting Active G Protein Fraction from Model of the Mutant Strain (p. 2-41)	Use the custom plot type to plot the *sst2Δ* `Ga_frac` data in the existing plot

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

You can find the simulation data under the **Simulation and Analysis** node for each model. You can also save the data from previous simulations, see "Saving Simulation Data" on page 2-30 for more information.

To plot data from a previous simulation run,

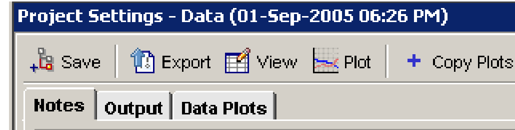
1 In the **Project Explorer**, select the **Data** node and click **Plot**.

   SimBiology plots the data from the most recent simulation run.

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

1 In the **Project Explorer**, click the **Data** node for the wild–type model.

2 In the **Project Settings-Data** pane, select the **Data 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. Select `Yeast_G_Protein_wt.Ga_frac` and click **OK**.

5 In the plot type table clear the **Create Plot** check boxes for the other plots, and click **Plot**. Leave the figure window open.

Creating a Custom Plot to Compare the Data

Create a custom plot that specifies that the species `Ga_frac` should be plotted with dashed lines for the *sst2Δ* model, and add a legend indicating `Ga_frac` from wild-type (`Ga_frac_wt`) and `Ga_frac` from *sst2Δ* (`Ga_frac_mut`)

1 In the **Project-Explorer**, select the **My Plot Types** node.

2 In the **Project Settings-My Plot Types** pane, point to the **Copy Plot Type** list and select `Time plot with dashed line`.

3 Click **Update**.

The **My Plot Types** command window is updated with the code for this plot.

4 Make the following changes to the code:

Under `% Show legend` add the `%` symbol to make these lines of code into comments.

```
%h = legend(names, 'Location', 'NorthEastOutside');
%set(h, 'Interpreter', 'none')
```

Specify the legend.

```
legend({'Ga_frac_wt','Ga_frac_mut'},'Location', 'NorthEastOutside','Interpreter','none');
```

5 Click **Save**.

A Save Plot Type dialog box opens. Enter a name for the custom plot type and click **Save**.

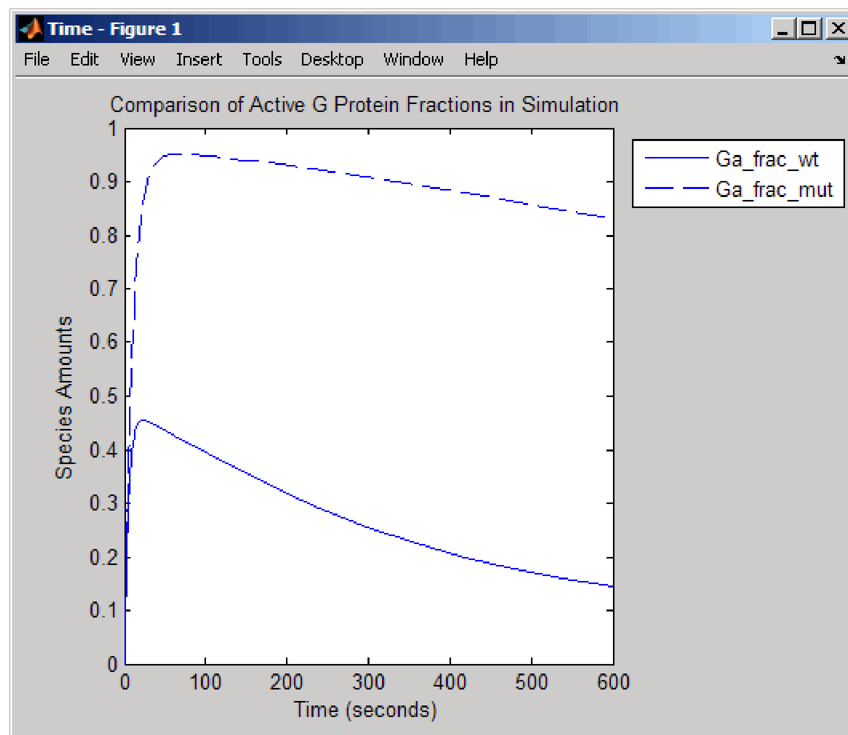
```
Time plot for Ga_frac comparisons
```

The new custom plot appears under **My Plot Types** in the **Project Explorer**. Now you can plot `Ga_frac` data from the wild-type strain with `Ga_frac` data from the *sst2Δ*.

Plotting Active G Protein Fraction from Model of the Mutant Strain

Use the new plot type to plot the *sst2Δ* `Ga_frac` data.

1. In the **Project Explorer**, click the **Data** node for the *sst2Δ* model.
2. In the **Project Settings-Data** pane, select the **Data Plots** tab.
3. From the **Plot Type** list select the `Time plot for Ga_frac comparisons` 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. Select `Yeast_G_Protein_mut.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 enables you to add data to the most recently generated plot. You must have the figure window open to exercise this option.


Time - Figure 1
File Edit View Insert Tools Desktop Window Help
Comparison of Active G Protein Fractions in Simulation
Species Amounts
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
Ga_frac_wt
Ga_frac_mut

Plotting Experimental Data with Simulation 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 (p. 2-43)	Working with experimental data in My Plot Types
Plotting the Data (p. 2-44)	Plotting the experimental data with the simulation data

Creating a Custom Plot for Experimental Data

1 In the **Project-Explorer**, select the **My Plot Types** node.

2 In the **Project Settings-My Plot Types** pane, copy the following code into the command 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 eror 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`.

3. 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 the second step.

4. Click **Save**.

   A Save Plot Type dialog box opens. Enter a name for the custom plot type and click **Save**.

   ```
   Ga_frac Experimental Data Plot
   ```

   The new custom plot appears under **My 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**, click the **Data** node for the wild–type model.

2 In the **Project Settings-Data** pane, select the **Data Plots** tab.

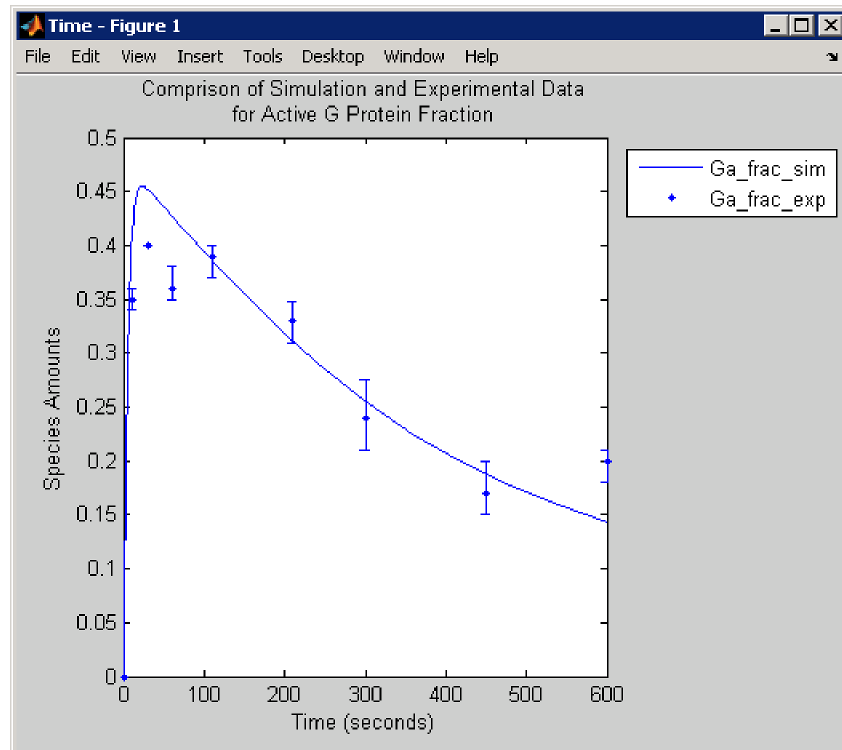
3 From the **Plot Type** list, select the `Ga_frac Experimental Data Plot` plot and click the **Add Plot Type** button.

4 Select the **Create Plot** check box for

a Time plot with the Y arguments limited to `Yeast_G_Protein_wt.Ga_frac` and choose `New Figure` from the **Plot Behavior** list for this plot.

b `Ga_frac Experimental Data Plot` and choose `Add to current axes` from the **Plot Behavior** list, for this plot.

5 In the plot type table, clear the **Create Plot** check boxes for the other plots and click **Plot**.



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.

Index

G

G Protein Cycle
- background 2-3
- in yeast 2-3
- model 2-1
- modeling of 2-5

G protein cycle model
- experimental data and assumptions 2-5
- reactions overview 2-5

G Proteins
- introduction to 2-3

M

Models
- G Protein Cycle 2-1

R

References
- yeast G protein cycle model 2-46

Y

Yeast G protein cycle model
- add reaction 2-12
- change simulation settings of 2-26
- comparison of
  - active G protein fractions 2-39
- copy and change 2-32
- creating a rule 2-25
- custom plot
  - dashed line 2-34
  - data from two models 2-40
- determine rate equation 2-14
- experimental and simulation data
  - comparison of 2-43
  - custom plot 2-43
- experimental data and assumptions 2-5
- graphical representation 2-5
- in SimBiology 2-10
- introduction 2-5
- introduction to 2-1
- mutant strain 2-34
- open SimBiology desktop 2-11
- parameters table 2-20
- project file 2-12
- reactions overview 2-5
- reactions table 2-19
- references 2-46
- setting species initial amounts 2-17
- simulation results
  - *sst2Δ* 2-37
  - wild-type 2-29
- species table 2-21
- tutorial goals 2-10
- wild-type strain 2-10