

SimBiology

For Use with **MATLAB**[®]

- Computation
- Visualization
- Programming

Getting Started

Version 2



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.

Getting Started with SimBiology

© COPYRIGHT 2005–2006 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+)
March 2006	Online only	Updated for Version 1.0.1 (Release 2006a)
May 2006	Online only	Updated for Version 2.0 (Release 2006a+)

Introduction

1

What Is SimBiology?	1-2
Required Software	1-3
Optional Software	1-3
Related Software	1-4
Features and Functions	1-5
Graphical User Interface	1-6
Customizing Desktop Views	1-8
Command-Line Interface	1-9
Data Formats and Projects	1-9
Modeling	1-10
Simulation	1-11
Further Analysis with MATLAB	1-12
SimBiology and SBML	1-13
What Is SBML?	1-13
SBML Support in SimBiology	1-13
SimBiology Objects	1-16
Organization of SimBiology Objects	1-17
Model	1-17
Submodels	1-18
Reaction	1-18
Kinetic Law	1-18
Species	1-19
Parameters	1-21
Rules	1-23

Gene Regulation Model	2-2
Model Diagram	2-2
Model Reactions	2-3
Modeling Using the SimBiology Graphical User	
Interface	2-7
Creating a SimBiology Model	2-7
Saving Your Work as a SimBiology Project File	2-9
Adding Reactions to a Model	2-9
Verifying the Model	2-13
Simulating a Model	2-14
Modeling Using the SimBiology Diagram	2-16
Creating a SimBiology Model	2-16
Saving Your Work as a SimBiology Project File	2-19
Adding Reactions to a Model Diagram	2-19
Simulating a Model	2-23
Changing Block Appearance	2-24
Modeling Using Command-Line Interface	2-35
Creating a SimBiology Model	2-35
Adding the Reaction for Transcription	2-37
Adding the Reaction for Translation	2-38
Adding the Reactions for Gene Regulation	2-38
Adding the Reactions for mRNA and Protein	
Degradation	2-39
Simulating the Model with an ODE Solver	2-39
Simulating the Model with a Stochastic Solver	2-41

Introduction

This chapter introduces the functions and features in SimBiology. An introduction to these features will help you to develop a conceptual model for working with the software and your biochemical data.

What Is SimBiology? (p. 1-2)

Description of this software product

Features and Functions (p. 1-5)

Functions grouped into categories that support systems biology tasks

SimBiology and SBML (p. 1-13)

Support for the Systems Biology Markup Language (SBML) specification and extended features in SimBiology

SimBiology Objects (p. 1-16)

Definition of the SimBiology object structure

What Is SimBiology?

SimBiology is an integrated environment for modeling biological processes, simulating the dynamic behavior of these processes, and analyzing the model with simulation and experimental data. Biological processes include metabolic, genetic, and signaling pathways with transform, binding, and transport reactions.

- **Model** — Design and build models by entering reactions, species, parameters, kinetic laws, rules, and submodels with a graphical user interface, a block diagram editor, or using the MATLAB® Command Window. Verify that the model can be simulated, and use the verification results to fix any incompatibilities in the model.

Import SBML models created with SimBiology or other modeling software that is compliant with the Systems Biology Markup Language (SBML) standard.

Export models created with the SimBiology desktop to MATLAB and continue your simulations and analysis with command-line functions. For more information, see “Modeling” on page 1-10.

- **Simulate** — Observe changes in species amounts and parameter values over time. Convert your model to a system of differential equations and simulate the model numerically with various differential equation solvers. The deterministic solvers include stiff and nonstiff ordinary differential equation (ODE) solvers. The stochastic solvers include a stochastic simulation algorithm with implicit and explicit tau variations. Perform multiple stochastic ensemble runs. For more information, see “Simulation” on page 1-11.
- **Analyze** — Save data from a simulation and export that data to the MATLAB workspace. Compare simulation and experimental data, perform sensitivity analysis, parameter estimation, and search for conserved moieties. For more information, see “Further Analysis with MATLAB” on page 1-12.

Required Software

To use SimBiology, you must first install the following product from The MathWorks:

MATLAB

Provides a command-line interface and an integrated software environment. For instructions, see the MATLAB installation documentation for your platform.

If you have installed MATLAB and want to check which other MathWorks products are installed, enter `ver` in the MATLAB Command Window.

Optional Software

Genetic Algorithm and Direct Search Toolbox

Solve optimization problems using genetic and direct search algorithms. If this toolbox is installed, you can use various genetic and direct search algorithms for parameter estimation in SimBiology. If the this toolbox is not installed, SimBiology uses the optimization algorithms available in MATLAB.

Optimization Toolbox

Solve standard and large-scale optimization problems. If the Optimization Toolbox is installed, you can use some algorithms included in the Optimization Toolbox for parameter estimation in SimBiology. If the Optimization Toolbox is not installed, SimBiology uses the optimization algorithms available in MATLAB.

Related Software
Bioinformatics Toolbox

Read, analyze, and visualize
genomic, proteomic, and microarray
data.

Features and Functions

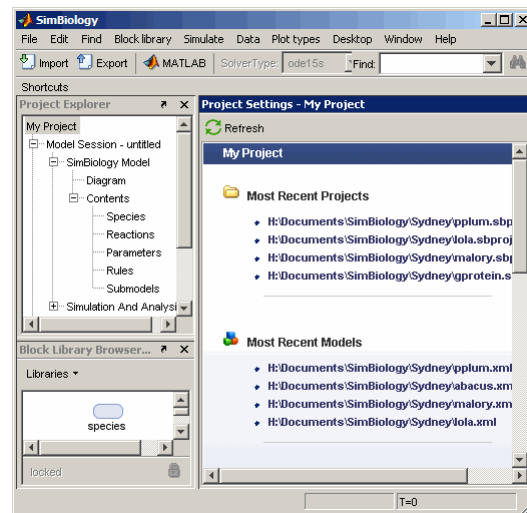
SimBiology includes many features to help you model mathematically biological processes, simulate those processes, and analyze the results.

Graphical User Interface (p. 1-6)	Enter reactions and kinetic data using a familiar spreadsheet format, or use the Diagram to work with block representation.
Customizing Desktop Views (p. 1-8)	Arrange the desktop according to your preferences.
Command-Line Interface (p. 1-9)	Create and simulate model objects interactively or by using M-code scripts.
Data Formats and Projects (p. 1-9)	Group related models and simulation data together into project files.
Modeling (p. 1-10)	Model metabolic, genetic, and signaling pathways.
Simulation (p. 1-11)	Select deterministic or stochastic solvers, simulate models, save data, and plot results. Perform multiple stochastic ensemble runs.
Further Analysis with MATLAB (p. 1-12)	Use MATLAB functions to analyze model properties, compare simulation data with experimental data; perform sensitivity analysis, parameter estimation, search for conserved moieties, and plot results.

Graphical User Interface

SimBiology provides access to command-line functionality through a graphical user interface.

The interface lets you model, simulate, and analyze biological pathways and reactions. Export your model to the MATLAB workspace and perform further analysis through the command-line.



The graphical user interface (SimBiology desktop) is organized with the following set of integrated tools:

- **Project Explorer** — View projects and the organization of models, and navigate to components and subcomponents using a hierarchical tree view. The project branch is divided into separate model sessions. A model session includes model and simulation branches. The model branch is further divided into components while the simulation branch is divided into simulation settings and simulation data.
- **Model Objects** — Enter model data using a familiar spreadsheet format, forms, comma-separated files, or previously saved SBML models. Model and simulate biological processes as reactions with reactants, products, and reaction rates. Reactions include processes that transform species, and processes such as transport, diffusion, and binding.
- **Diagram** — Enter model data using block representations for species, reactions, and submodels. Use the Plot block to visualize simulation data during a simulation.

- **Simulation** — Simulate a model with deterministic or stochastic solvers, save simulation data, export data to the MATLAB workspace.
- **Plotting** — Generate simulation plots, create and save custom plots
- **Block Library Browser** — Use the basic library included in SimBiology for blocks in the diagram. Create and store your own custom block library.
- **Diagram Overview** — Pan through large models using the miniaturized view of the model in this tool.
- **Diagram Table View** — Tools like hiding, pinning, splitting, and cloning for model layout in the Diagram and generate histogram for degree of connectivity of species in a model.
- **Bookmarks** — Create custom searches for object properties with specific values within a project, save these criteria in bookmarks.
- **Find** — Use the find feature for simple searches within a project.
- **Output** — Shows you the results of Find, error or warnings generated by Verify (model verification and validation), and errors or warnings generated by Simulation.
- **Embedded Help** — The embedded help pane is context sensitive to the tasks you are performing while you build, simulate and analyze your model. As you select component branches in the Project Explorer pane or optional panes, the topics that support your current tasks appear in the embedded help pane on the right.

The Help browser supplements the embedded help information with detailed reference material for command-line functions; object methods and properties; tutorials to help you get started; and more details about modeling, simulation, and analysis.

- **Kinetic Law Manager** — View built-in kinetic laws and define your own kinetic laws.
- **Unit Manager** — View built-in units or define your own units composed from basic physical quantities.
- **Table Column Manager** — Customize the desktop spreadsheets using the Table Column Manager to display or hide the properties you are interested in viewing or editing.

- **M-code capture** — Convert interactive GUI actions with the SimBiology desktop to M-code scripts, and load them into MATLAB.

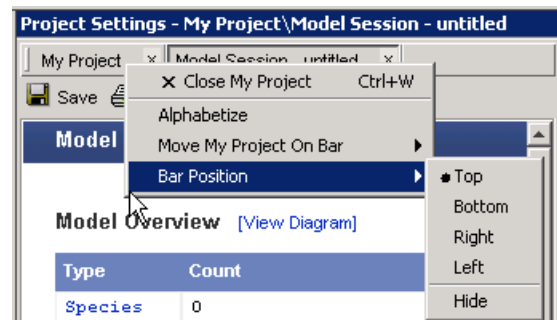
Customizing Desktop Views

You can arrange the desktop layout according to your preference.

- **Changing pane location** – Click the title bar of a pane, drag to the desired location, and drop. SimBiology shows you an outline of the pane during drag and drop. If the dropped location is on top of another pane, click the tabs at the bottom to navigate between the panes.
- **Obtaining default layout** – From the **Desktop Menu**, point to **Desktop Layout** and select **Default Layout**.
- **Organizing tables** – You can also add or delete the properties listed in the table columns on each pane using the **Table Column Manager**.

If the **Table Column Manager** does not appear on your SimBiology desktop, from the **Desktop** menu, select **Table Column Manager**. The **Table Column Manager** opens in the desktop, and if the context-sensitive help pane is open the help updates with more information.

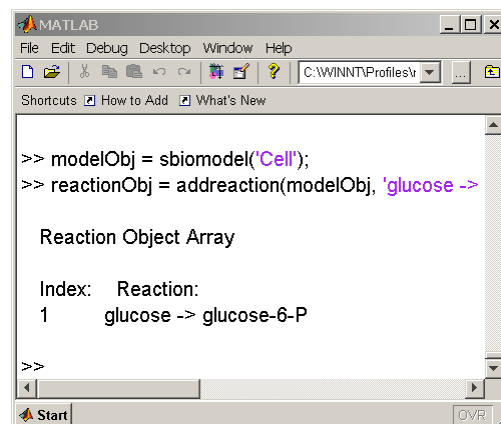
- **Changing pane title bar position** – SimBiology lets you open multiple panes from the **Project Explorer**. Each pane has a title bar in the **Project Settings** pane. You can change the location of the title bar. Right-click the title bar and then select **Alphabetize** or point to **Bar Position** and select the location.



Command-Line Interface

All the features in SimBiology are accessible and executable interactively from the MATLAB Command Window or programmatically using M-code scripts.

You can either create your models in the SimBiology desktop and export to the workspace or create your models directly in MATLAB.



```

MATLAB
File Edit Debug Desktop Window Help
C:\WINNT\Profiles\...
Shortcuts How to Add What's New

>> modelObj = sbiomodel('Cell');
>> reactionObj = addreaction(modelObj, 'glucose ->

Reaction Object Array

Index: Reaction:
1      glucose -> glucose-6-P

>>
Start OVR
  
```

- **Functions** — Use functions to create projects, manage the unit and kinetic law libraries, and simulate models.
- **Methods** — Methods create objects and specify object property values.
- **M-code capture** — In the SimBiology desktop, you can record and save your actions as an M-code script. After loading the script into MATLAB, you can then run the script programmatically and observe what you created interactively.

Data Formats and Projects

SimBiology creates and stores all your models in project files. Projects group related models and simulation data into project files.

- **Projects** — Organize projects with related models, simulation data, and analysis results.

A *project* combines models, simulations, analysis, and results. Each project is divided into one or more *model sessions*.

- **Model session** — In the SimBiology desktop, a project can contain one or more model sessions. Each model session contains a model, any included submodels, all its configuration settings, and any saved data.

The configuration settings include simulation settings, units, kinetic laws, tags, notes, and annotations.

- **Kinetic Law library** — Use built-in kinetic laws or create your own kinetic laws for defining reaction rates.
- **Units library** — Use built-in units or create your own units for species and parameters.

In addition, SimBiology supports SBML Level 2 Version 1. Systems Biology Markup Language (SBML) is becoming the standard language for documenting and sharing mathematical models for biological systems. You can save your models to an SBML-formatted file or read an SBML file created with SimBiology or other modeling software. For more information, see “SimBiology and SBML” on page 1-13.

When you save a project, the model definition is saved with the project. However, you can independently export a SimBiology model to an SBML file. Use this SBML file for importing your model into SBML Level 2 Version 1-compliant modeling and simulation software.

When you import an SBML file created with other software, SimBiology translates the model definition into SimBiology objects.

Modeling

SimBiology is an environment for designing and building biochemical models.

- **Model objects** — A SimBiology model is a collection of interrelated reactions and rules that transform, transport, and bind species. The model components are species, reactions, parameters, rules, and submodels. After creating a model object, you can add reaction objects to the model, and then add kinetic law objects and parameter objects to the reaction object. Other objects define simulation settings, and rules.
- **Model properties** — The properties define the characteristics of an object. For example, a reaction object includes properties for reactions, reaction rates, and links to associated species, parameters, and kinetic laws. You can interactively change object properties in the graphical user interface. Use the MATLAB `get` and `set` commands to list object properties and change their values at the command line.

- **Submodel objects** — SimBiology extends the current version of SBML by hierarchically organizing a model with submodels. A submodel can be a group of reactions.

By partitioning your model into functional units, you can easily understand the model and the interactions between the submodels. You also can reuse the functional units in other modeling projects.

- **Notes** — SimBiology extends the usefulness of the SBML **Notes** property by allowing HTML formatted text. This property lets you link to journal articles in PDF and other formats, Web pages, databases, graphic files, experimental data, and verbose project notes that do not fit into a text box.

Simulation

SimBiology includes both deterministic and stochastic solvers.

- **Stochastic simulation** — Genetic and signaling pathways often have a small number of molecules that are simulated more accurately with stochastic algorithms. The stochastic solvers are based on Daniel Gillespie's exact solution algorithm with variations for implicit and explicit tau leaping.

Gillespie, D., (1977), "Exact Stochastic Simulation of Coupled Chemical Reactions," *The Journal of Physical Chemistry*, 8(1):2340–2361.

The implicit tau-leaping solver works optimally with large, numerically stiff systems. When solving numerically stiff systems, this solver remains stable for large time intervals (tau) where the explicit tau-leaping solver might not.

While stochastic simulations might accurately reflect the actual processes in a cell, they might not be practical for models with a large number of reactions or higher species amounts. Metabolic and signaling transduction pathways fall into this category. In this case, mass-action kinetics and ODE solvers can be a reasonable approximation.

- **Deterministic simulation** — You can choose from the list of stiff and nonstiff deterministic simulation solvers that are included with MATLAB. These solvers are reliable and robust.
- **Dimensional analysis and unit conversion** — Dimensional analysis compares the physical quantities for parameters and species in mathematical expressions (reaction rates, algebraic rules, assignment

rules, and rate rules) and verifies that they are in consistent dimensions. Unit conversion scales the parameter values and species amounts to have the same units before simulation.

- **Plotting simulation results** — At the end of the simulation, SimBiology plots the results in a figure window, or you can change plot settings and replot selected species yourself. In the SimBiology desktop, insert and configure a Plot block in the Diagram to view the results in a plot while the simulation runs.
- **Simulations settings** — Configuration set objects define the properties needed for simulation. These properties include the simulation time, absolute and relative tolerances, and controls for dimensional analysis and unit conversion.

Further Analysis with MATLAB

SimBiology is closely integrated with MATLAB.

- **Getting simulation data into MATLAB** — During a simulation, SimBiology saves (logs) the values for parameters and the amounts of species to the MATLAB workspace.
- **Plotting simulation data and analyzing results** — Using the functions in MATLAB, you can select and scale species for plotting.
- **Parameter estimation and optimization** — Estimate parameters that are unknown in the model.
- **Sensitivity analysis** — Calculate sensitivities of species with respect to species and parameter input factors.
- **Moiety conservation** — Calculate a complete set of linear conservation relations for the species in a SimBiology model.
- **Ensemble runs** — Perform iterative simulation runs to compare and analyze simulation results from stochastic solvers. Find the mean and variance as a function of time.

SimBiology and SBML

SimBiology supports SBML Level 2 Version 1. You can import from and export to SBML-compatible .sbml files.

What Is SBML? (p. 1-13)

Language for mathematically defining biological models and reaction pathways

SBML Support in SimBiology (p. 1-13)

Description of SBML features and functions supported in SimBiology

What Is SBML?

Systems Biology Markup Language (SBML) is an information standard for sharing, evaluating, and developing systems biology models cooperatively.

Its purpose is to provide a structure for creating models and sharing those models among various modeling and simulation software tools. The current specification is available on the Web at <http://sbml.org/documents/>.

SBML Support in SimBiology

SimBiology supports a subset of the SBML Level 2 Version 1 specification. Unsupported features include

- Compartments — Model compartments are not supported. If an imported SBML model has a single compartment, the model is loaded as a top-level model. If the model has multiple compartments, SimBiology returns a warning and does not load the SBML file.
- Volume — Volume is not supported and cannot be specified.
- Events — Events in an SBML file are ignored when you are importing into a project.
- Piecewise kinetics — Models with piecewise kinetics are loaded in, but SimBiology ignores the definitions for piecewise kinetics.
- Function definitions — Models containing functional definitions are loaded, but SimBiology generates a warning and ignores the function definitions.

- MATLAB incompatible variable names in `UnitDefinition` — Models that have variable names incompatible with MATLAB in `UnitDefinition`, are not loaded and you see an error message.

The following features in SimBiology are not supported by SBML:

- Projects — SimBiology creates and stores all your models in project files with the `.sbproj` extension. You can store multiple models in one project file.
- Hierarchical models — Organize your models with hierarchies within the model including submodels.
- Kinetic Law — You can store Kinetic law information such as the kinetic law name and an abstract kinetic law expression.

When you export a SimBiology model to an SBML file, you save the model as an SBML-compatible `.xml` file. The following information is included in this file:

- Species, reactions, parameters, and rules that are defined in the model
- All unit definitions in SBML-compliant form
- The reaction rate equation, but not the kinetic law

Example

In SimBiology, the Michaelis-Menten equation,

$$V_{\max} * S / K_m + S$$

is a kinetic law stored in the kinetic law library.

Specifically, for a one-substrate enzyme-catalyzed reaction, if you assign $V_{\max} = v$, $K_m = K_a$, and $S = A$, the reaction rate equation is

$$V_a * A / K_a + A$$

- Model component properties with SBML equivalents, such as **Annotation**, species, and parameter unit values will be exported. Features and

properties specific to SimBiology, such as submodels, **Tag**, and **Active** cannot be exported.

SimBiology Objects

SimBiology is organized as a hierarchical list of objects under one root object.

Organization of SimBiology Objects (p. 1-17)	Software implementation of SimBiology objects and definitions.
Model (p. 1-17)	Definition of the SimBiology model object
Submodels (p. 1-18)	Definition of the SimBiology submodel object
Reaction (p. 1-18)	Definition of the SimBiology reaction object
Kinetic Law (p. 1-18)	Definition of the SimBiology kinetic law object
Species (p. 1-19)	Definition of the SimBiology species object
Parameters (p. 1-21)	Definition of the SimBiology parameter object
Rules (p. 1-23)	Definition of the SimBiology rule object

Organization of SimBiology Objects

This is the hierarchical organization of SimBiology objects:

```
Model
  Species
  Parameters
  Reaction
    Kinetic Law
      Parameters
  Rules
  Submodel
    Species
    Parameters
    Reaction
      Kinetic Law
        Parameters
    Submodel
      . . .
  Submodel
    . . .
```

Parameters can be scoped to a kinetic law or the model.

The recommended method to store your models in SimBiology is to create and store project files.

Model

A SimBiology *model* is a collection of interrelated reactions and rules that transform, transport, and bind species. The model components are species, reactions, parameters, rules, and submodels. After creating a model object, you can add reaction objects to the model, and then add kinetic law objects and parameter objects to the reaction object. Other objects define simulation settings and rules.

Properties define the characteristics of an object. For example, a reaction object includes properties for reactions, reaction rates, and links to associated species, parameters, and kinetic laws. You can interactively change object properties in the graphical user interface. Use the MATLAB `get` and `set`

commands to list object properties and change their values at the command line.

Submodels

A *submodel* is a collection of reactions and rules that make up a distinct module in the model.

Submodels can reference parameters in the submodel and model-level parameters in successively higher parent models. Submodels cannot, however, reference species that are at the parent model level.

Use submodels to keep parts of your model separate. For example consider the citric acid or Krebs cycle, which, apart from being a major generator of ATP, is also the source of intermediates that are precursors in the biosynthesis of key building blocks such as amino acids (proteins) and porphyrins (heme, chlorophyll). In SimBiology, the Krebs cycle can be modeled in a top-level model, and the porphyrin and amino acid synthesis pathways can be modeled as two separate submodels.

Reaction

A *reaction* describes a transformation, transport, or binding process that changes one or more species. Typically, the change is the amount of a species.

`Creatine + ATP <-> ADP + phosphocreatine`

Enter a reaction in the form

`glucose + 2 ADP + 2 Pi -> 2 lactic acid + 2 ATP + 2 H2O`

Spaces are required before and after species names and stoichiometric values.

Kinetic Law

A *kinetic law* provides a mechanism for applying a specific rate law to multiple reactions. It acts as a template for the reaction rate. The kinetic law is defined by an expression, and the species and parameter variables used in the expression.

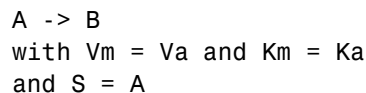
The ReactionRate is the result of defining the names of parameters and species variables in a rate law. To get ReactionRate, the species variables and parameter variables that participate in the reaction rate should be clearly assigned in the **Project Settings-Reactions** pane **Kinetic Law** tab.

For example, the kinetic law Henri-Michaelis-Menten is expressed as

$$V_{\max} * S / K_m + S$$

where V_m and K_m are defined as parameters and S is defined as a species. (To see parameter and species definitions for a kinetic law, access the **Kinetic Law Manager** from the **Desktop** menu).

By applying the kinetic law Henri-Michaelis-Menten to a reaction



the rate equation is

$$V_a * A / K_a + A$$

A , V_a , and K_a are defined in the **Kinetic Law** tab.

The *reaction rate equation* is a mathematical expression that describes the rate at which the reactants combine to form products.

For a reaction following Mass Action kinetics,



the rate equation is written as

$$K * \text{Creatine} * \text{ATP} - K_{\text{rev}} * \text{phosphocreatine} * \text{ADP}$$

Species

A *species* is a chemical or entity that participates in reactions, for example, DNA, ATP, Pi, creatine, G-Protein, or Mitogen-Activated Protein Kinase (MAPK). Species amounts can vary or remain constant during a simulation.

Specifying Species Names in SimBiology

SimBiology species names can have any number, letter, or character, for example, N-acetyl-D-glucosamine. The names of species are case-sensitive.

Note the following constraints,

- Name cannot be left empty.
- Name cannot be the literal words `null` and `time`. Note that you could specify species names with these words contained within the name, for example, `nullaminoacids` or `nullnucleotides`.
- If you are using a species name that is not a valid MATLAB variable name, do the following:
 - Enclose the name in square brackets when writing a reaction rate equation or a rule. For example, enclose `[DNA polymerase+]` within brackets in reaction rates and rules.
 - Enter the name without brackets when you are creating the species or when you are adding the reaction. For example, enter `DNA polymerase+` when specifying the name of the species or while writing the reaction.
- Name cannot be the characters `i`, or `j`, or contain `->`, `<>`, `[`, or `]`. The names `i` and `j` are reserved MATLAB characters. Because expressions in abstract kinetic laws, reaction rates, and rules are considered to be MATLAB code, SimBiology evaluates `i` or `j` as an imaginary number and not as the value of species `i` or `j`. For example, an expression `V*S*i/K` is interpreted to have three names, `V`, `S`, and `K`, instead of four. Use brackets to protect such variables.
- If a variable in a reaction rate equation or rule has the same name as a MATLAB function, SimBiology evaluates the expression as a call to the MATLAB function. In general, when creating variable names, you should avoid using MATLAB function names or variable names that are invalid in MATLAB.

If you change the Name of a species you must configure all applicable elements, such as rules that use the species, any user-specified `ReactionRate`, or the kinetic law object property `SpeciesVariableNames`. Use the method `setspecies` to configure `SpeciesVariableNames`.

To update species names in the SimBiology graphical user interface, access each appropriate pane through the **Project Explorer**. You can also use the **Find** feature to locate the names that you want to update. The **Output** pane opens with the results of **Find**. Double-click a result row to go to the location of the model component.

SimBiology automatically updates species names for reactions that use `MassAction` kinetic law.

Parameters

A *parameter* is a quantity that can change or can be constant. In SimBiology, parameters are generally used to define rate constants.

Scope refers to the definition of a parameter at the model (global) level or kinetic law (local) level.

If the scope of the parameter is global in the model, it can be used by any rule, submodel, or by any reaction rate in the model. If the scope of the parameter is at the kinetic law level, it can be used only by the kinetic law in the reaction for which it was defined.

How SimBiology Uses Scope

For reaction rate, SimBiology hierarchically uses the value of the parameter with scope at the kinetic law level first. If there is no such parameter at the kinetic law level, SimBiology looks for the parameter with scope at the model level.

If there are two parameters with the same name, one at the model level and the other at the kinetic law level, SimBiology uses the value of the parameter at the kinetic law level for the reaction rate. SimBiology uses the value of the parameter at the model level for any rules or submodels that reference the parameter.

For more details on scope and further links to procedures on changing scope, see [Parameter and Scope](#) in the SimBiology User's Guide.

Specifying Parameter Names in SimBiology

Consider the following while naming parameters in SimBiology:

- Names are case-sensitive.
- Name cannot be empty or time.
- If you are using a parameter name that is not a valid MATLAB variable name, do the following:
 - Enclose the name in square brackets when writing a reaction rate equation or a rule.
 - Enter the name without brackets when you are creating the parameter.

For example, enclose `[K_DNA polymerase+]` within brackets in reaction rates and rules, but, enter `K_DNA polymerase+` when creating a parameter.

- Name cannot be the characters `i`, or `j`, or contain `[`, or `]`. The names `i` and `j` are reserved MATLAB characters. Because expressions in abstract kinetic laws, reaction rates, and rules are considered to be MATLAB code, SimBiology evaluates `i` or `j` as an imaginary number and not as the value of species `i` or `j`. For example, an expression `V*S*i/K` is interpreted to have three names, `V`, `S`, and `K`, instead of four. Use brackets to protect such variables.
- If a variable in a reaction rate equation or rule has the same name as a MATLAB function, SimBiology evaluates the expression as a call to the MATLAB function. In general, when creating variable names, you should avoid using MATLAB function names or variable names that are invalid in MATLAB.

If you change the Name of a parameter you must configure all applicable elements, such as rules that use the parameter, any user-specified `ReactionRate`, or the kinetic law object property `ParameterVariableNames`. Use the method `setParameter` to configure `ParameterVariableNames`.

To update parameter names in the SimBiology graphical user interface, access each appropriate pane through the **Project Explorer**. You can also use the **Find** feature to locate the names that you want to update. The **Output** pane opens with the results of **Find**. Double-click a result row to go to the location of the model component.

Rules

A *rule* is a mathematical expression that modifies a species amount or a parameter value. There are three types of rules in SimBiology:

- **Algebraic** — Algebraic rules are evaluated continuously during a simulation. An algebraic rule takes the form $0 = \text{Expression}$ and the rule is specified as the Expression. For example, you could write a mass conservation expression such as $\text{species_total} = \text{species1} + \text{species2}$ where species_total is the independent variable, write the rule as

```
species1 + species2 - species_total
```

- **Assignment** — Assignment rules are evaluated once at the beginning of a simulation. Assignment rules are expressed as $\text{Variable} = \text{Expression}$. For example you could write an assignment rule to set the amount of species1 to be proportional to species2 .

```
species1 = k/species2  
(where k is a known constant with units = concentration^2)
```

- **Rate** — Rate rules are evaluated continuously during a simulation. Rate rules are determined by $d\text{Variable}/dt = \text{Expression}$ which is expressed in SimBiology as $\text{Variable} = \text{Expression}$. For example, to define the rate of change in the quantity of species3 ($d\text{species3}/dt$), write the rule in SimBiology as

```
species3 = k * (species1 + species2)
```

One example case for a rate rule is when Species1 is at the boundary of the system, but the rate of input of species1 to the system can be determined by a rate rule.

Note If you plan to use a parameter in a rule, remember to set the **Scope** of the parameter to `model`. Further, if you are using an algebraic or rate rule to vary the value of a parameter during the simulation, clear the **ConstantValue** check box for the parameter in the **Parameters** pane, **Settings** tab.

Tutorial Model

This chapter will get you started with creating and simulating dynamic models with SimBiology. Models follow the structure described in the Systems Biology Markup Language (SBML Level 2 Version 1) specification, but they include features that extend the current release of the standard.

Gene Regulation Model (p. 2-2)

Description of a simple gene regulation model with transcription, translation, and negative feedback to suppress transcription

Modeling Using the SimBiology Graphical User Interface (p. 2-7)

Model created interactively with the SimBiology desktop

Modeling Using the SimBiology Diagram (p. 2-16)

Model created interactively with the SimBiology diagram interface

Modeling Using Command-Line Interface (p. 2-35)

Model created interactively with the MATLAB command-line interface

Gene Regulation Model

This model is an example of simple gene regulation, where the protein product from translation controls transcription. You could create a more complex model by adding the enzymes, coenzymes, cofactors, nucleotides, and amino acids that are not included in this model.

Model Diagram (p. 2-2)

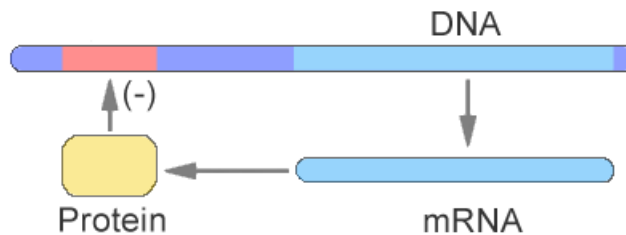
Visualize a systems biology model for gene regulation with different levels of detail.

Model Reactions (p. 2-3)

Describe a systems biology model for gene regulation with a series of reaction and rate equations.

Model Diagram

You can visualize a systems biology model with various levels of detail. One view sketches only the major species and processes. The gene regulation example simplifies the regulatory mechanism by not showing the contributions of RNA polymerase and any cofactors. The protein product from gene expression binds to a regulatory region on the DNA and represses transcription.



Another way of looking at a systems biology model is to list the reactions in a model with the processes they represent.

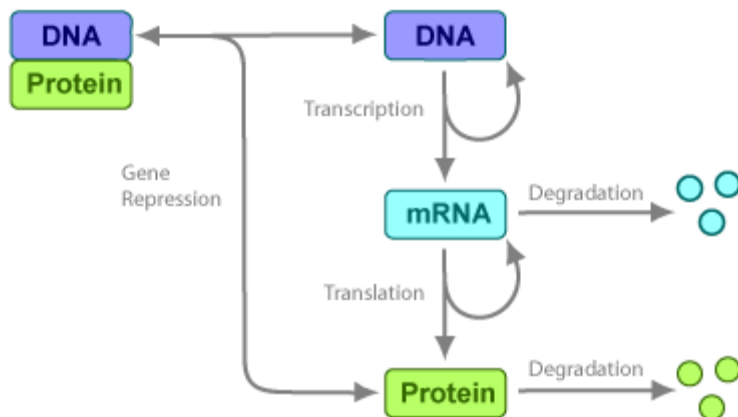
```

DNA -> DNA + mRNA           (transcription)
mRNA -> mRNA + protein       (translation)
DNA + protein -> DNA_protein (binding)
DNA_protein -> DNA + protein (unbinding)

```

mRNA -> null (degradation)
 protein -> null (degradation)

Drawing the reaction pathways will help you to visualize the relationships between reactions and species. In the gene regulation example, as the amount of a protein increases, the protein forms a complex with the gene responsible for its expression, and slows down protein production.



Model Reactions

Reaction equations define a systems biology model at the level of detail needed for a software program to simulate the dynamic behavior of the model. The following reactions for transcription, translation, binding, and degradation describe a simple gene-regulating mechanism. For additional information about this model, see “Gene Regulation Model” on page 2-2.

Transcription

Transcription is where RNAPolymerase and cofactors bind with a DNA molecule. The RNAPolymerase then moves along the DNA and combines nucleotides to create mRNA. A simple model of transcription shows only the DNA and mRNA.



This model simplifies the transcription and the synthesis of mRNA by using a single reaction.

```
Reaction: DNA -> DNA + mRNA
Reaction rate: v = k1*DNA   molecule/second
Species: DNA = 50   molecule
          mRNA = 0   molecule
Parameters: k1 = 0.2 1/second
```

Translation

After the mRNA moves from the nucleus to the cytoplasm, it can bind with ribosomes. The ribosomes move along the mRNA and create proteins with the help of tRNAs bound to amino acids. A simple model of translation shows only the mRNA and protein product.



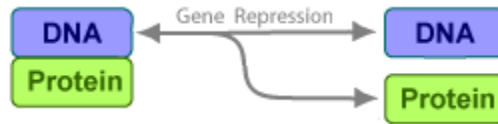
The synthesis of the protein is modeled as a single reaction.

```
Reaction: mRNA -> mRNA + protein
Reaction rate: v = k2*mRNA   molecule/second
Species: mRNA = 0   molecule
          protein = 0   molecule
Parameters: k2 = 20 1/second
```

Gene Repression

Transcription of DNA to mRNA is regulated by the binding of the protein product from translation to the DNA. As more protein is produced, the DNA is

bound with the protein more often and less time is available for transcription with the unbound DNA.



Forward Reaction (Binding)

Reaction: DNA + protein -> DNA_protein

Reaction rate: $v = k_3 \cdot \text{DNA} \cdot \text{protein}$ molecule/second

Species: DNA = 50 molecule

protein = 0 molecule

Parameters: $k_3 = 0.2$ 1/(molecule*second)

Notice how the units are described for the parameter k_3 . In the literature, some of the common ways to display second-order rate constants are

$\text{mole}^{-1} \text{second}^{-1}$ or $1/\text{mole}\cdot\text{second}$. However, software evaluations of these common ways are sometimes ambiguous.

Reverse Reaction (Unbinding)

Reaction: DNA_protein -> DNA + protein

Reaction rate: $k_{3r} \cdot \text{DNA_protein}$ molecule/second

Species: DNA_protein = 0 molecule

Parameters: $k_{3r} = 1$ 1/second

For this tutorial, model the binding and unbinding reactions as one reversible reaction.

Reaction: DNA + protein <-> DNA_protein

Reaction rate: $v = k_3 \cdot \text{DNA} \cdot \text{protein} - k_{3r} \cdot \text{DNA_protein}$ molecule/second

Species: DNA = 50 molecule

protein = 0 molecule

Parameters: $k_3 = 0.2$ 1/(molecule*second)

$k_{3r} = 1$ 1/second

Degradation

Protein and mRNA degradation are important reactions for regulating gene expression. The steady-state level of these compounds is maintained by a balance between synthesis and degradation reactions. Proteins are hydrolyzed to amino acids with the help of proteases, and nucleic acids are degraded to nucleotides.



mRNA degradation is modeled as a transformation to the null species.

```
Reaction: mRNA -> null
Reaction rate:  $v = k_4 * \text{mRNA}$  molecule/second
Species: mRNA = 0 molecule
Parameters:  $k_4 = 1.5$  1/second
```

Likewise, protein degradation is also modeled as a transformation to the null species. The species null is a predefined species name in SimBiology.



```
Reaction: protein -> null
Reaction rate:  $v = k_5 * \text{protein}$  molecule/second
Species: protein = 0.0 molecule
Parameters:  $k_5 = 1.0$  1/second
```

Modeling Using the SimBiology Graphical User Interface

This section shows how to mathematically model a simple gene-regulation pathway using the SimBiology desktop. For more information about this model, see “Gene Regulation Model” on page 2-2. For information on building a model with the GUT’s Diagram interface, see “Modeling Using the SimBiology Diagram” on page 2-16.

Alternatively, you can use M-code and the MATLAB command-line interface (see “Modeling Using Command-Line Interface” on page 2-35). A comparison of these methods will help you understand the model objects that MATLAB creates behind the SimBiology desktop, and how to access and edit a model through the MATLAB Command Window.

Creating a SimBiology Model (p. 2-7)	Use the SimBiology desktop to create a structure for entering reaction and kinetic equations.
Saving Your Work as a SimBiology Project File (p. 2-9)	Save your work in the SimBiology project format.
Adding Reactions to a Model (p. 2-9)	Add reactions with species, reaction rate equations, and parameter values to your model.
Verifying the Model (p. 2-13)	Verify the model to check if it is ready for simulation
Simulating a Model (p. 2-14)	Simulate your model and plot the change in amount of species over time.

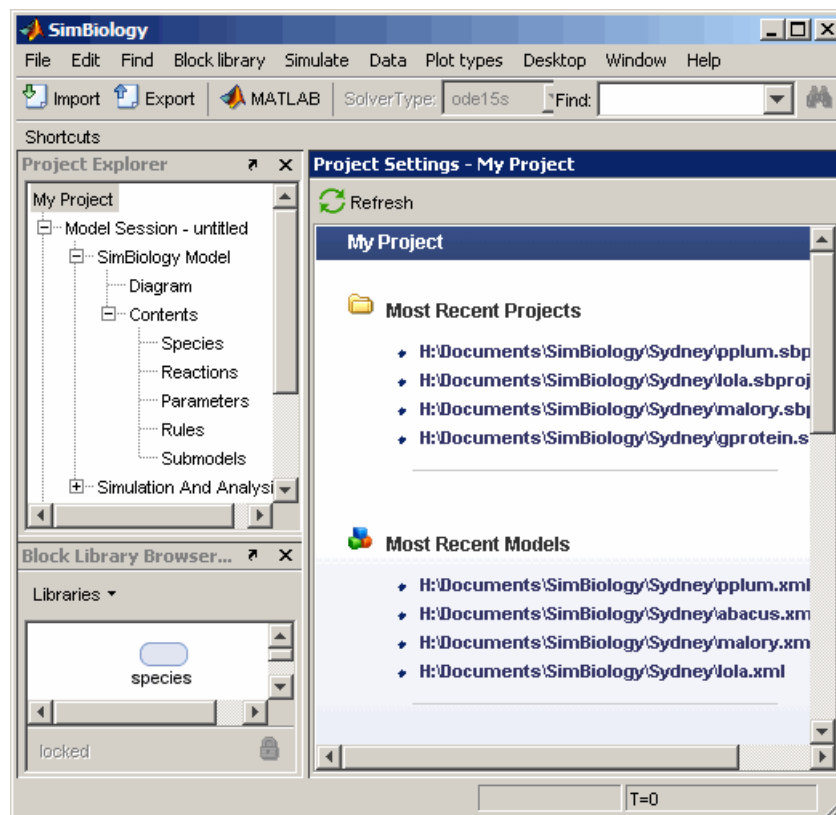
Creating a SimBiology Model

A SimBiology model is a collection of objects that are structured hierarchically. At least one model object is needed to contain all the other objects, including reaction and submodel objects.

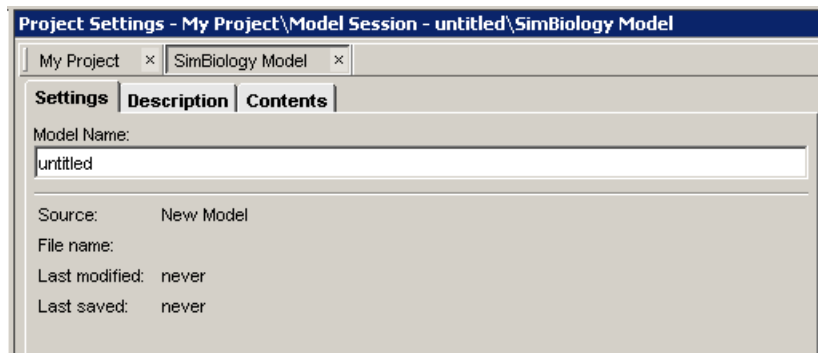
- 1 Open the SimBiology desktop window. In the MATLAB Command Window, type

```
sbiodesktop
```

The SimBiology desktop window opens with a new but empty model.



2 In the **Project Explorer**, double-click **SimBiology Model** to open the **SimBiology Model** pane.



- 3 In the **Model Name** box, enter a name for your model and press **Enter**.
For example, enter `cell`.

Saving Your Work as a SimBiology Project File

Project is the file format that SimBiology uses to save one or more model sessions; the file extension is `.sbproj`. Projects lets you save custom settings, notes, and data associated with your models. For more information on projects, see “Data Formats and Projects” on page 1-9.

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**.

`gene_regulation.sbproj`

Adding Reactions to a Model

SimBiology models are defined by connected reactions representing transformation, binding, and transport.

- 1 In the **Project Explorer**, double-click **Reactions** to open the **Reactions** pane.

- 2** In the **Enter Reaction** box, enter the reactants and products, separated by a hyphen and right angle bracket. For example, enter the following simple reaction for transcription.

DNA -> DNA + mRNA

- 3** Click **Add**.

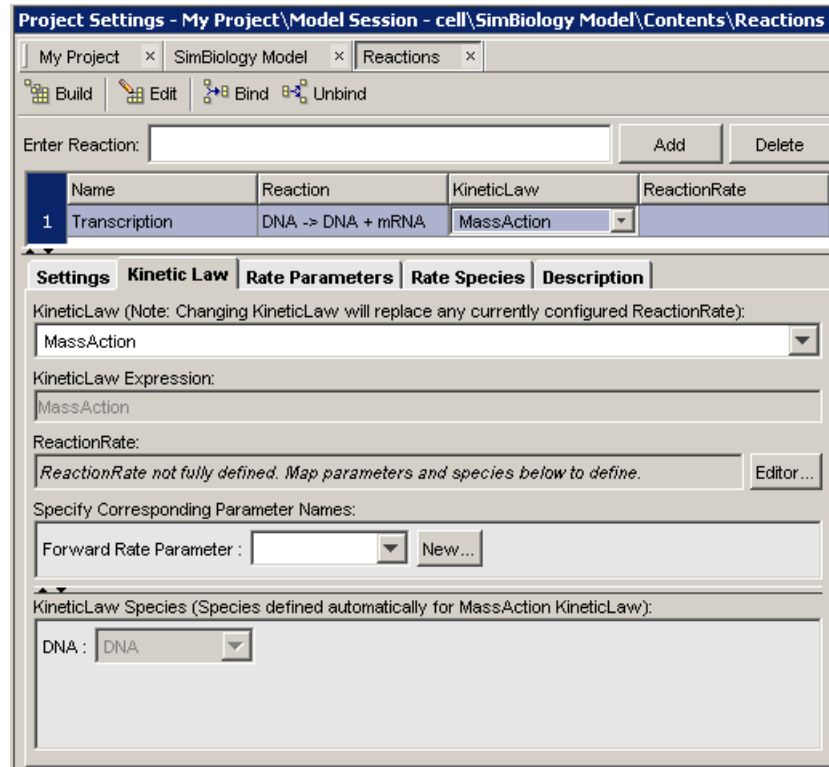
- 4** Double-click the **Name** box and enter a name for the reaction. For example, enter

Transcription

- 5** From the **KineticLaw** list, select MassAction.

Name	Reaction	ReactionRate
transcription	DNA -> DNA + mRNA	k1*DNA

- 6** Click the **Kinetic Law** tab.



7 In the **Specify Corresponding Parameter Names** box, click **New**. The New Parameter dialog box opens.

- a** Specify name of the parameter in the **Name** box; for **Forward Rate Parameter**, , enter **k1**.
- b** From the **Scope** list select **Kinetic Law**.

When you choose this option, SimBiology makes this parameter available only to the reaction for which it was created.

In this case, it is the $\text{DNA} \rightarrow \text{DNA} + \text{mRNA}$ reaction.

- c** Click **Create**.

Note that SimBiology updates **ReactionRate** for the reaction to show $k1 \cdot \text{DNA}$.

Your reaction table should resemble the following.

Name	Reaction	KineticLaw	ReactionRate
Transcription	DNA -> DNA + mRNA	MassAction	k1*DNA

- 8 Click the **Rate Parameters** tab. Notice that SimBiology lists k1 with a default value of 1.0 for the parameter.
- 9 Double-click the **Value** cell and enter a new value. From the **ValueUnits** list, select the units for this parameter. The rate constant k1 for transcription is 0.2 1/second.

Your parameters table should resemble the following:

Name	Scope	Value	ValueUnits	ConstantValue
k1	cell	1.0	1/second	<input checked="" type="checkbox"/>

- 10 In the **Project Explorer**, double-click **Species**. The species you entered in the reaction are automatically added to the species table.

The default value for a species is 0.0.

- 11 Double-click the **InitialAmount** cell and enter a new value. From the **InitialAmountUnits** list, select the units for this species. The initial value for the species DNA is 50 molecule and 0.0 molecule for mRNA.
- 12 Enter the remaining reactions. For a list of reactions and parameters, see “Model Reactions” on page 2-3.

Notice that the ValueUnits for the second-order rate constant k3 is not a default unit in the list. However, molecule and second are default units and you can enter this unit directly by typing

$$1/(\text{molecule*second})$$

Without the parentheses, molecule would be divided into 1, and then the result multiplied (not divided) by second.

Your reaction, species, and parameter panes should look similar to the following ones.

Name	Reaction	KineticLaw	ReactionRate
Transcription	DNA -> DNA + mRNA	MassAction	k1*DNA
Translation	mRNA -> mRNA + protein	MassAction	k2*mRNA
Binding/Unbinding	DNA + protein <-> DNA_protein	MassAction	k3*DNA*protein - k3r*DNA_protein
mRNA Degradation	mRNA -> null	MassAction	k4*mRNA
Protein Degradation	protein -> null	MassAction	k5*protein

Name	InitialAmount	InitialAmountUnits
DNA	50.0	molecule
mRNA	0.0	molecule
protein	0.0	molecule
DNA_protein	0.0	molecule

Name	Scope	Value	ValueUnits
k1	DNA -> DNA + mRNA	0.2	1/second
k2	mRNA -> mRNA + protein	20.0	1/second
k3	DNA + protein <-> DNA_protein	0.2	1/(molecule*second)
k3r	DNA + protein <-> DNA_protein	1.0	1/second
k4	mRNA -> null	1.5	1/second
k5	protein -> null	1.0	1/second

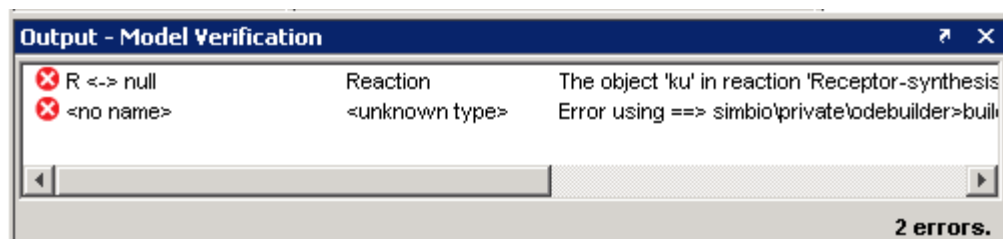
For more detailed information about the reactions in this model, see “Gene Regulation Model” on page 2-2.

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.


Simulating a Model

After you enter the reactions for your model, you can simulate its dynamic behavior and plot the results. If you have more than one model session in your project, then you must first make the model you want to simulate active. SimBiology shows the name of the active model in bold in the **Project Explorer**. To make a model active,

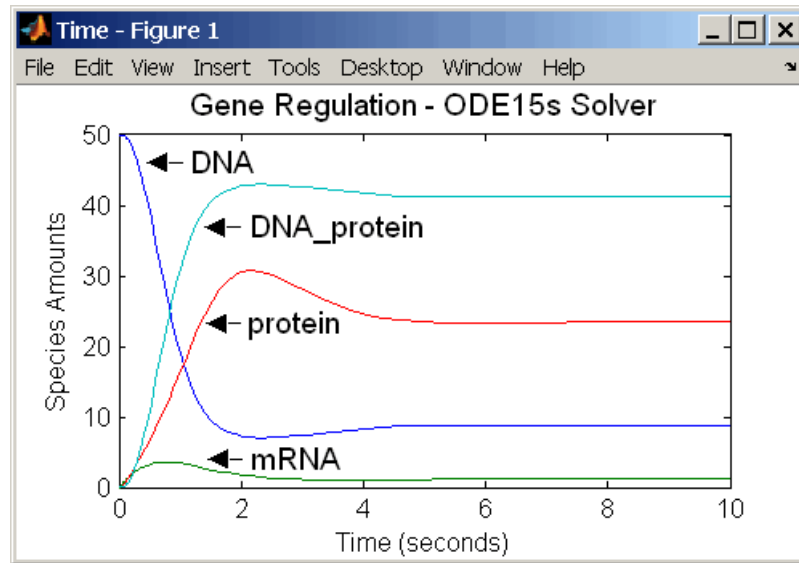
- 1 In the **Project Explorer**, right-click the model and select **Activate (model name)**.

The model name is bold in the **Project Explorer** indicating that this model is the active model in the project.

To simulate a model,

- 1 From the simulation toolbar, in the **Solver Type** list, select a stochastic or ordinary differential equation (ODE) solver. For example, select ode15s.
- 2 In the **Stop time** box, enter the length of time for the simulation. For example, enter 10 seconds.
- 3 From the **Simulate** menu, click **Start**, or from the simulation toolbar, click  (**Run**).

The simulation runs to completion and then draws the results in a figure window. You can annotate your plot in the figure window.



Modeling Using the SimBiology Diagram

This section shows how to model a simple gene-regulation pathway using the SimBiology Diagram interface. For more information about this model, see “Gene Regulation Model” on page 2-2.

Creating a SimBiology Model
(p. 2-16)

Use the SimBiology Diagram to interact with a graphical view of the model while entering reactions and kinetic equations.

Saving Your Work as a SimBiology Project File (p. 2-19)

Save your work in the SimBiology project format.

Adding Reactions to a Model Diagram (p. 2-19)

Add reactions with species, reaction rate equations, and parameter values to your model.

Simulating a Model (p. 2-23)

Simulate your model and plot the change in amount of species over time.

Changing Block Appearance (p. 2-24)

Change the appearance of blocks, text, lines, and indicators.

Creating a SimBiology Model

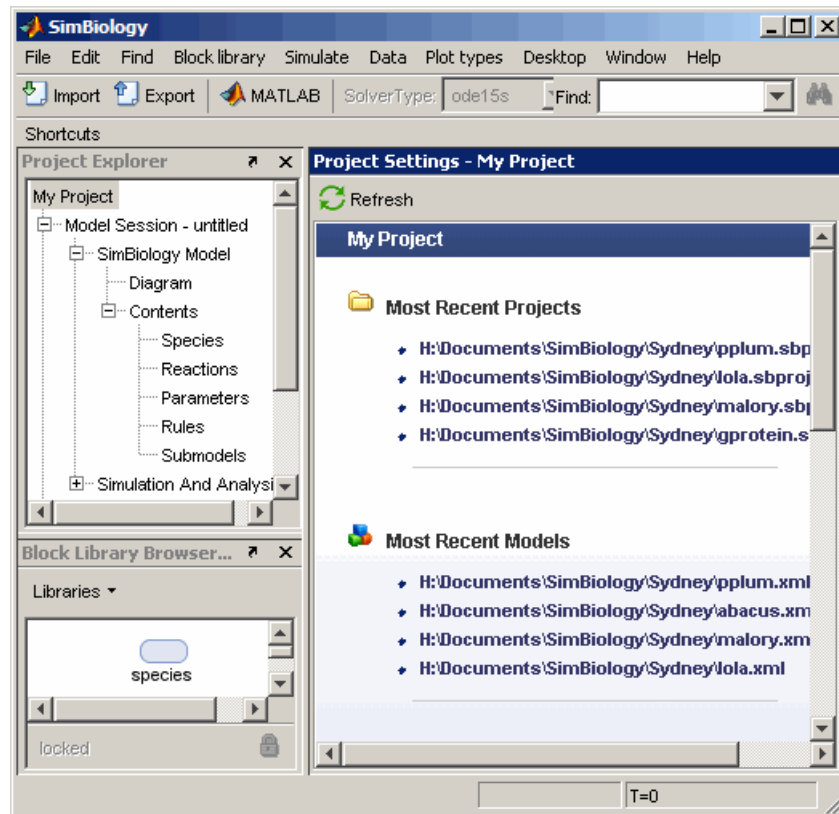
A SimBiology model is a collection of objects that are structured hierarchically. At least one model object is needed to contain all the other objects, including reaction and submodel objects. The SimBiology Diagram lets you work with the familiar graphical representation of biological pathways and processes.

This section shows you how to build your model using block representations for each species, reaction, and submodel. The model reactions are shown in “Gene Regulation Model” on page 2-2.

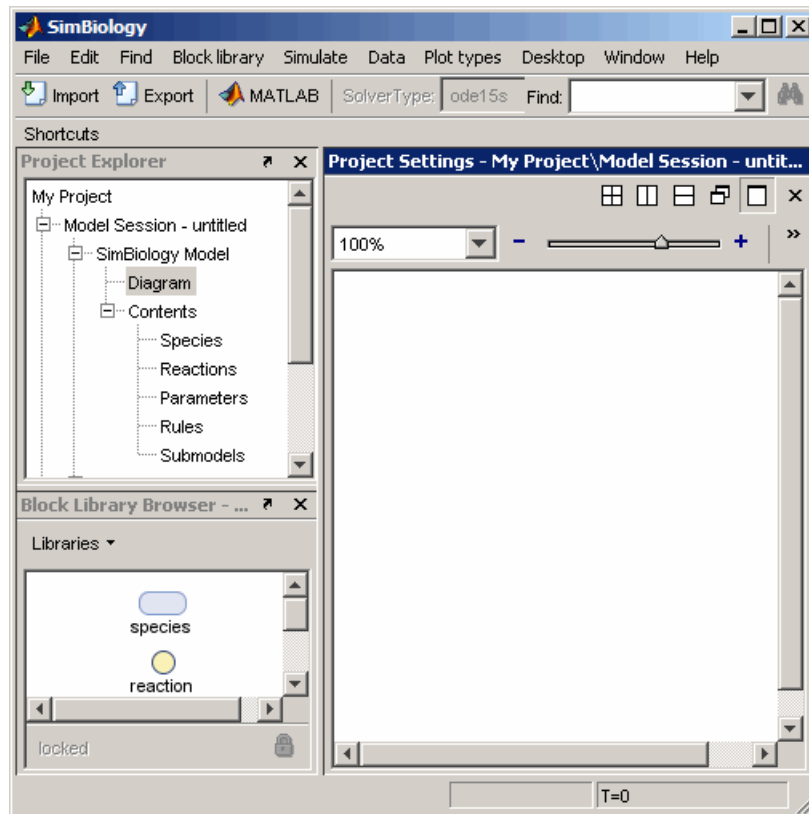
- 1 Open the SimBiology desktop window. In the MATLAB Command Window, type

```
sbiodesktop
```

The SimBiology window opens, with a new but empty, model.



2 In the **Project Explorer**, double-click **Diagram**. SimBiology opens an empty **Diagram** pane.



- 3** In the **Diagram** pane, right-click and select **Properties**. In the Diagram Properties dialog, enter a name for your model. For example, in the **Model Name** box, enter **cell**, and press **Enter**.

SimBiology updates **Model Session** in the **Project Explorer** with the new name.

All the procedures in this section refer to the SimBiology desktop in the default layout. To set the desktop to the default layout, select **Desktop > Desktop Layout > Default Layout**.

Saving Your Work as a SimBiology Project File

Project is the file format that SimBiology uses to save one or more model sessions; the file extension is `.sbproj`. Projects lets you save custom settings, notes, and data associated with your models. For more information on projects, see “Data Formats and Projects” on page 1-9.

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**.

`gene_regulation.sbproj`

Adding Reactions to a Model Diagram

SimBiology models are defined by connected reactions representing transformation, binding, and transport. You can graphically represent the reactions in the diagram. This section shows you how to add, connect, and configure blocks in a diagram. For this example, use the Basic library that is shipped with SimBiology.

- 1 From the **Block Library Browser**, click and drag a Species block into the diagram.
- 2 Double-click the Species block to open the Species Properties dialog box.

The screenshot shows a dialog box with three tabs: 'Settings', 'Description', and 'Appearance'. The 'Settings' tab is selected. It contains the following fields and options:

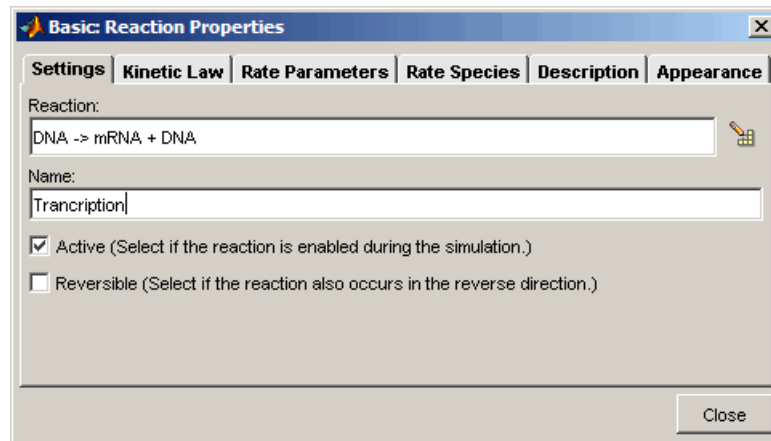
- Name:** A text box containing 'species_1'.
- InitialAmount:** A text box containing '0.0'.
- InitialAmountUnits:** A dropdown menu.
- ConstantAmount** (Select if the species quantity cannot change during the simulation.)
- BoundaryCondition** (Select if the species quantity is not determined by the set of reactions.)
- Close** button at the bottom right.

- 3** In the **Name** box, enter the species name. For example, enter DNA.
- 4** In the **InitialAmount** cell, enter a new value. From the **InitialAmountUnits** list, select the units for this species. The initial value for the species DNA is 50 molecule.
- 5** From the **Block Library Browser**, click and drag another Species block into the diagram. Name this species mRNA. The **InitialAmount** for mRNA is the default (0.0). From the **InitialAmountUnits** list, select molecule as units for this species.
- 6** From the **Block Library Browser**, click and drag a Reaction block into the diagram.
- 7** Hold down the **Ctrl** key (Windows and Linux platforms) or the **Option** key (Macintosh) while clicking the DNA Species block. Drag the line to the reaction and release the mouse. SimBiology assigns DNA as a reactant.

Since DNA is also a product of the reaction, connect a line from the Reaction block to DNA. SimBiology draws a dashed line, indicating that DNA is a modifier for the reaction.

- 8** Draw a line from the Reaction block to the mRNA Species block.
- 9** Double-click the Reaction block (alternatively, right-click and select **Properties**) to open the Reaction Properties dialog box. Notice that the **Reaction** box shows the reaction (DNA -> mRNA + DNA) to which the block applies.

- 10** Click the **Name** box and give the reaction a name. For example, enter Transcription.



- 11** Click the **Kinetic Law** tab.

- 12** In the **Specify Corresponding Parameter Names** box, click **New**. The New Parameter dialog box opens

- a** Specify name of the parameter in the **Name** box; for **Forward Rate Parameter**, enter k1.
- b** From the **Scope** list, select Kinetic Law.

When you choose this option, SimBiology makes this parameter available only to the reaction for which it was created. In this case, it is the DNA -> DNA + mRNA reaction.

- c** Click **Create**.

Notice that SimBiology updates **ReactionRate** for the reaction to show $k1 \cdot \text{DNA}$.

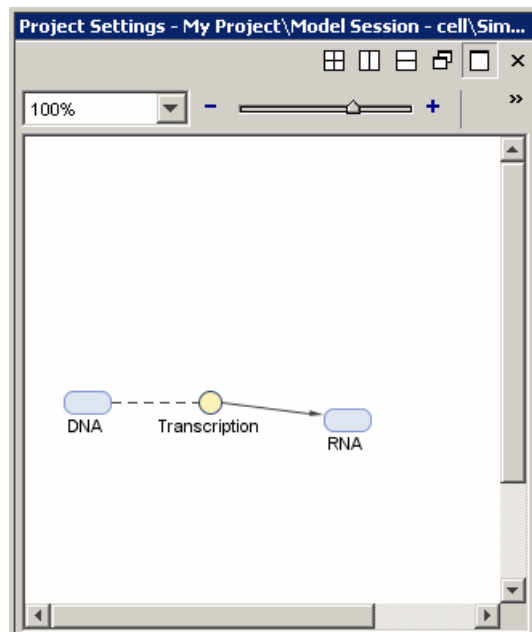
- 13** Click the **Rate Parameters** tab. Note that SimBiology lists k1 with a default value of 1.0 for the parameter.

- 14** Double-click the **Value** cell and enter a new value. From the **ValueUnits** list, select the units for this parameter. The rate constant k_1 for transcription is 0.2 1/second .

Your parameters table should look resemble the following.

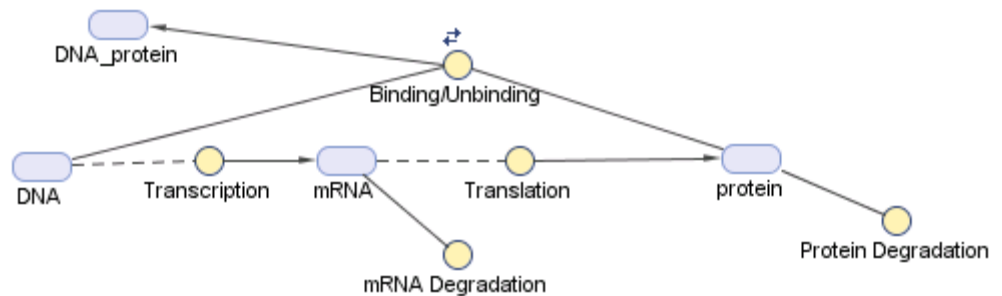
Name	Scope	Value	ValueUnits	ConstantValue
k1	cell	1.0	1/second	<input checked="" type="checkbox"/>

- 15** Click **Close**. Your diagram should now look similar to the one below.



- 16** Add the other species and reactions blocks, and create parameters for each reaction as shown in the steps above. For a list of reactions and parameters, see “Model Reactions” on page 2-3. Verify that the reaction rate is determined for each reaction.

After creating the model, the diagram should now look similar to the one below.



Notice the use of dashed lines for species that are both reactants and products in a reaction. Note the indicator for a reversible reaction associated with the Binding/Unbinding reaction.

You can change the layout and appearance of the blocks at anytime. Double-click a block and click the **Block Appearance** tab to change the color, and text font, size and location. See “Changing Block Appearance” on page 2-24.

Simulating a Model


After you enter the reactions for your model, you can simulate its dynamic behavior and plot the results. If you have more than one model session in your project, then you must first make the model you want to simulate active. SimBiology shows the name of the active model in bold in the **Project Explorer**.

To make a model active,

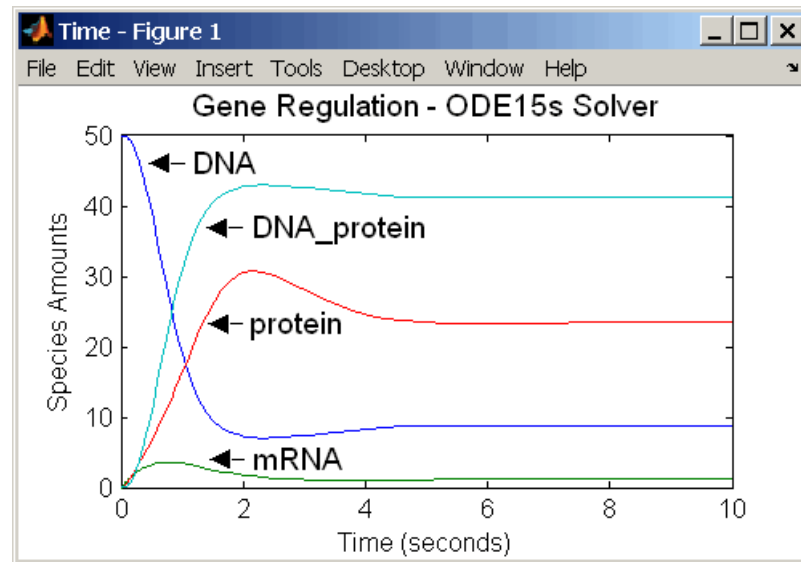
- 1 In the **Project Explorer**, right-click the model and select **Activate (model name)**.

The model name appears bold in the **Project Explorer** indicating that this model is the active model in the project.

To simulate the model,

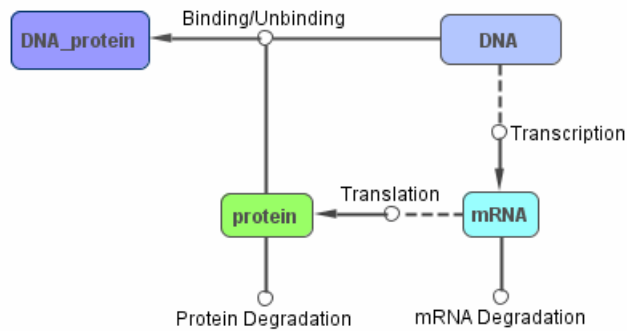
- 1** From the simulation toolbar, in the **Solver Type** list, select a stochastic or ordinary differential equation (ODE) solver. For example, select ode15s.
- 2** In the **Stop time** box, enter the length of time for the simulation. For example, enter 10 seconds.
- 3** From the **Simulate** menu, click **Start**, or from the simulation toolbar, click  (**Run**).

The simulation runs to completion and then draws the results in a figure window.



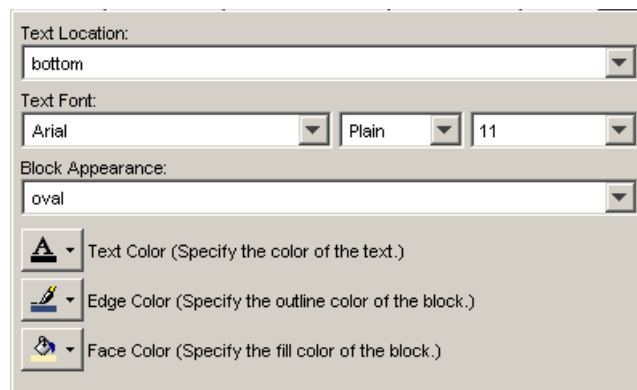
Changing Block Appearance

You can change the appearance of blocks, text, lines, and indicators in the diagram. The following figure is one example:



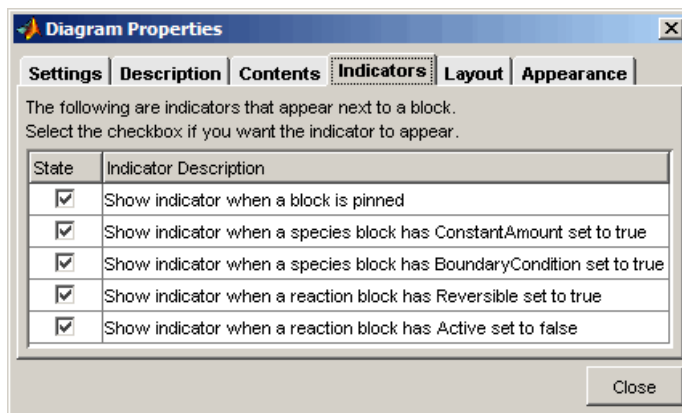
This example uses the DNA Species block and the Binding/Unbinding Reaction block, to show you how to change the color of the blocks, the location of the text, the appearance of the reversible indicator, and line appearance as shown in the previous figure.

- 1 Double-click the Species block or the Reaction block to open the Properties dialog box.
- 2 Click the **Appearance** tab.



- 3 From the **Text Location** list, select where the text should be positioned in the diagram. The default is bottom. For example:

- For the DNA Species block, select Center to position the text within and in the center of the Species block.
 - For the Binding/Unbinding Reaction block select Top.
- 4** From the **Text Font**, **Block Appearance**, **Text Color**, **Edge Color**, and **Face Color** lists, select the text font, block appearance, text color, edge color, block fill color.
- Tip: choose **Face Color** > No Face Color to obtain the result shown for reaction blocks in the diagram figure above.
- 5** Double-click a line to open the Line Properties dialog box . The default line width is 1.0 pixels. The line width in the example diagram above is 2.0.
- 6** In the diagram, right-click and select **Properties**. The Diagram Properties dialog box opens.
- 7** Click the **Indicators** tab.



- 8** Clear or select an indicator check box, to turn off or turn on the display of the indicator in the diagram. For example, clear the **Reversible** property indicator, and click **Close**.

Notice that the reversible indicator is now off for the Binding/Unbinding Reaction block.

- 9 Click **Layout** to automatically change the layout of the diagram. You can manually lay out the diagram using the pin, split, clone, and hide features in the diagram.

For more information on the pin, split, clone, and hide features in the diagram see “Changing Diagram Layout Manually” on page 2-27, and for information on automatic layout in SimBiology, see “Automatic Layout of Blocks” on page 2-30

Changing Diagram Layout Manually

You can manually lay out the diagram using the pin, split, clone, and hide features in the diagram. Use **Diagram Table View** or a block’s right-click context menu to change the properties discussed below. You can access **Diagram Table View** by clicking **Show Table** in the **Diagram** pane.

Tip While you are working on diagram layout, remember to select **File> Save Project** to save the desired layout of your diagram. SimBiology saves the layout in the project.

To manually layout the model diagram,

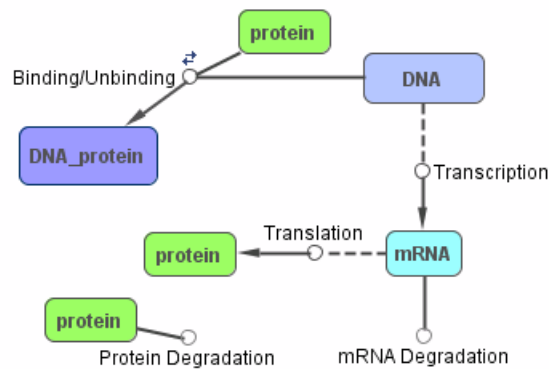
- 1 Right-click a Species block, and select **Split Block**. For example, split the protein Species block.

SimBiology provides a representation of the split block for each instance of the species in the model. Notice that although there could be multiple split blocks there is only one species.

If you change any settings in the Species Properties dialog for any of these split blocks, the change will be propagated to each of the split blocks. For example, if you change the appearance of the block, the change affects all split blocks.

Note You can only split species blocks.

- 2 Manually move the split blocks to the desired position.



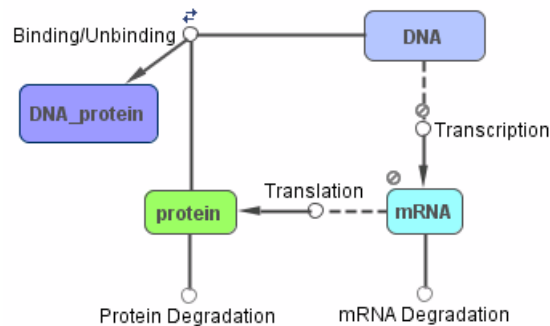
3 Click **Show Table** to open the **Diagram Table View** pane.

	Name	Split	Hide	Pin	Degree
1	DNA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
2	protein	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
3	DNA_protein	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
4	mRNA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4
5	Protein Degradation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
6	mRNA Degradation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
7	Translation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
8	Transcription	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
9	Binding/Unbinding	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3

Diagram Table View is a tool that lets you hide or show blocks, split or combine blocks, and pin or unpin blocks. You can select a single block or perform the action on multiple selected blocks. Diagram Table View also shows you the degree of connectivity in your diagram and also lets you view a histogram of the degree of connectivity.


- 4 In the row for the protein Species block, clear the **Split** check box. SimBiology combines the split blocks. Alternatively, you can right-click the Species block and select **Combine Split Blocks**.
- 5 Select the row for the Transcription Reaction block and press **Ctrl+click** to select the row for the mRNA Species block. **Ctrl+click**, lets you select discontinuous rows.

- 6** Right-click and select **Pin Selected Blocks**. SimBiology pins the blocks. You cannot move pinned blocks manually or with automatic layout. Note the indicator for pinned blocks in the diagram.

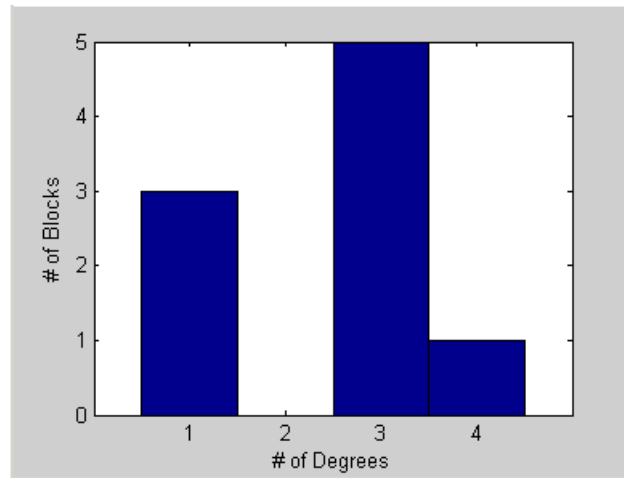


- 7** Select the row for the Protein Degradation reaction and press **Shift+click** to select the row for the mRNA Degradation reaction.
- 8** Right-click and select **Hide Selected Blocks**. SimBiology hides the two reactions in the diagram. The reactions rows are visible in the **Diagram Table View** pane.

Note Hiding a block only hides it graphically. The block is still represented in the mathematical model.

- 9** Click **Show All Blocks** to see all blocks including hidden blocks.
- 10** Click  in the Diagram Table View toolbar.

SimBiology opens a figure window with a histogram of the degrees of connectivity plotted as the number of connections versus the number of blocks.



You can also see the degree of connectivity in the **Degree** column.

Automatic Layout of Blocks

You can use the automatic layout tool in the **Diagram** pane.

- Click **Layout** to use the automatic layout feature.
- Select blocks and click **Layout** to automatically lay out the selected blocks in the diagram.

Automatic layout only works on blocks that are not pinned. You can right-click and select **Unpin Block** if you want to layout all blocks. Leave blocks pinned if you do not want the position of the block to change during layout.

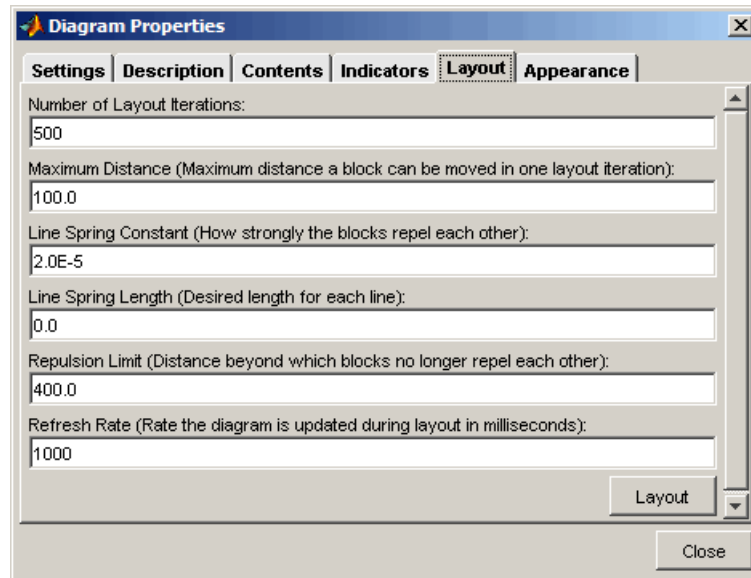
This example starts with the blocks for mRNA and Transcription pinned.

The default settings for automatic layout should work well for many models. You can optionally choose to change the automatic layout settings in SimBiology.

To change the settings for automatic layout,

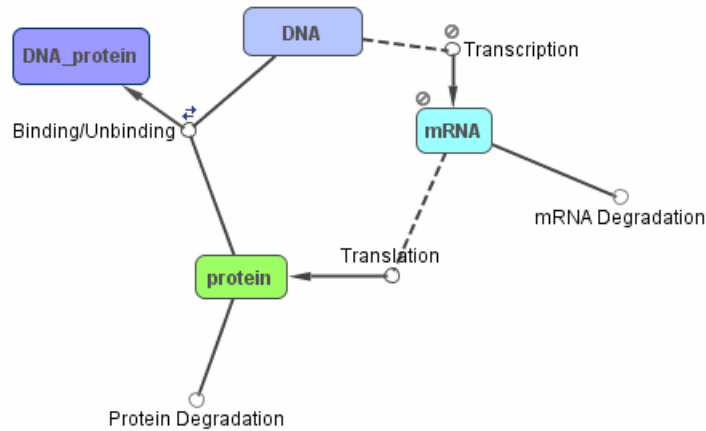
- 1 Right-click the diagram background and select **Properties**. The Diagram Properties dialog box opens.

2 Click the **Layout** tab.



- 3** In the **Number of Layout Iterations** box, enter the number of iterations to include in the layout. The default is 500. More iterations give you a better layout, but also take more time. For example, enter 700.
- 4** In the **Maximum Distance** box, specify the distance that a block moves in one iteration. The default is 100.0. If you enter a smaller value, layout is slower. Large values can also cause oscillation in the layout. For example, enter 50.
- 5** In the **Line Spring Constant** box, specify the strength with which the lines between blocks pull blocks together. The default is 2.0E-5. Increasing the spring constant creates more compact diagrams by increasing the spring attraction. For example enter 2.0E-6 and click **Layout** to apply the settings.

The layout should be similar to the following diagram:



6 In the **Line Spring Length** box, specify the length for each line. The default is 0.0, indicating that force due to the spring is zero at this length. Compress this length to push the blocks further apart.

For example enter 10.

7 In the **Repulsion Limit** box, specify the maximum repelling distance that limits the distance at which blocks repel each other. The default is 400.0. Setting this to a large distance causes all blocks to repel each other. This spreads the diagram out. A smaller value allows only nearby blocks to repel, creating a more uniformly filled diagram.

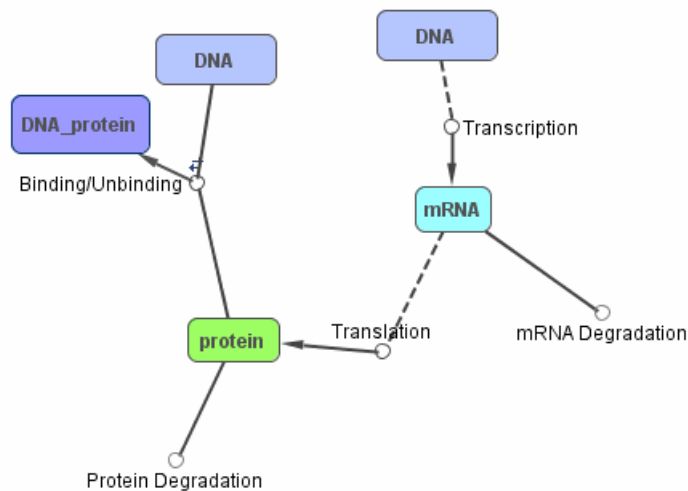
For example, enter 200.

8 In the **Refresh Rate** box, enter the rate (in milliseconds) at which the diagram is updated during layout. The default is 1000. For example, enter 500.

9 In the **Diagram** pane, right-click the DNA Species block and select **Split Block**.

10 In the **Diagram Table View** pane, clear the **Pin** check boxes for the Transcription Reaction block and the mRNA Species block.

11 Click **Layout** to apply the settings. Your diagram should resemble the following diagram:



12 Click **Close** to close the Diagram Properties dialog box.

How SimBiology Does Diagram Layout

The automatic layout feature in SimBiology arranges blocks using a simple “force directed” algorithm.

Every block is affected by three forces: edges of the diagram repel blocks, blocks repel each other, and lines between blocks act as springs that draw blocks toward each other.

The algorithm tries to minimize the sum of these forces. In each iteration, SimBiology calculates the total force on each block. SimBiology then moves each block in the direction of and proportional to the force acting on it. Two adjustable parameters control the dynamics of the layout:

- The number of iterations that are taken determines how long the layout takes and thereby how well it can do.
- The maximum distance constrains the distance that any block will be moved in one iteration. A small value makes layout progress go slowly. A large value may cause oscillation in the layout.

Three adjustable parameters influence the layout of the diagram:

- The spring constant controls how strongly the lines between blocks pull blocks together. Increasing the spring constant creates more compact diagrams as the spring attraction dominates the repulsion between blocks.
- The spring length controls the “natural length” of the lines between blocks. The force due to the spring is zero at this length. When a spring is compressed to less than this length, it pushes the blocks apart.
- The maximum repelling distance limits the distance at which blocks repel each other. Setting this value to a large distance causes all blocks to repel each other. This spreads the diagram out. A smaller value allows only nearby blocks to repel, creating a more uniformly filled diagram.

Modeling Using Command-Line Interface

This section shows how to model a simple gene-regulation pathway using M-code and the command-line interface. For more information about this model, see “Gene Regulation Model” on page 2-2.

You can also use a graphical user interface (see “Modeling Using the SimBiology Graphical User Interface” on page 2-7 and “Modeling Using the SimBiology Diagram” on page 2-16). A comparison of the three methods will help you understand the model objects that MATLAB creates behind the sbiodesktop, how to access and edit a model through the MATLAB Command Window, and enable you to determine the workflow of your preference.

Creating a SimBiology Model (p. 2-35)	Use the MATLAB Command Window to create a structure for entering reaction and kinetic equations.
Adding the Reaction for Transcription (p. 2-37)	A DNA molecule produces multiple copies of mRNA.
Adding the Reaction for Translation (p. 2-38)	Multiple copies of a protein are produced from each mRNA molecule.
Adding the Reactions for Gene Regulation (p. 2-38)	The protein binds with DNA and stops transcription.
Adding the Reactions for mRNA and Protein Degradation (p. 2-39)	Degradation reactions balance the synthesis reactions and maintain a steady state.
Simulating the Model with an ODE Solver (p. 2-39)	Select a deterministic numerical solver.
Simulating the Model with a Stochastic Solver (p. 2-41)	Select a stochastic solver that considers the probability of the occurrence of a reaction.

Creating a SimBiology Model

A SimBiology model is a collection of objects that are structured hierarchically. At least one model object is needed to contain all the other objects, including reaction and submodel objects.

- 1 Create a model object. For example, to create a model with the name `cell`, in the MATLAB Command Window, type

```
Mobj = sbiomodel('cell')
```

MATLAB creates a model object (`Mobj`) in the workspace and displays a summary of the objects in the model.

```
Systems Biology Model - cell
```

```
Model Components:
Models:           0
Parameters:       0
Reactions:        0
Rules:            0
Species:          0
```

- 2 Display all the model object properties by typing

```
get(Mobj)
```

MATLAB displays a list of properties for the model object and their current values.

```
Annotation: ''
Models: [0x1 double]
Name: 'cell'
Notes: ''
Parameters: [0x1 double]
Parent: [1x1 SimBiology.Root]
Species: [0x1 double]
Reactions: [0x1 double]
Rules: [0x1 double]
Tag: ''
Type: 'sbiomodel'
UserData: []
```

Some of the model properties are themselves objects or arrays of objects.

Adding the Reaction for Transcription

A simple model of transcription shows only the DNA and mRNA. For more details, see “Transcription” on page 2-3.

- 1 Enter the reaction $\text{DNA} \rightarrow \text{DNA} + \text{mRNA}$ with reaction rate equation $v=k*\text{DNA}$.

```
Robj1 = addreaction(Mobj, 'DNA -> DNA + mRNA');
Kobj1 = addkineticlaw(Robj1, 'MassAction');
```

The property `KineticLawName` is set to `'MassAction'`. Because the reaction is enough information to determine the rate equation, the property `Expression` is set to `'MassAction'` and the abstract kinetic law expression is assumed to be `'k*DNA'`.

- 2 Define the reaction rate constant $k_1=0.2$. For a first-order reaction, the property `ParameterVariableNames` has a single rate constant that needs defining. First create a parameter object named `k1`, and then use the parameter object name to set the rate constant in `ParameterVariableNames` to that object.

```
Pobj1 = addparameter(Kobj1, 'k1', 0.2);
set(Kobj1, 'ParameterVariableNames', 'k1');
```

- 3 Enter the initial amount for the species $\text{DNA}=50$ and $\text{mRNA}=0$.

```
s = sbioselect(Mobj, 'Type', 'species', 'Name', 'DNA');
s.InitialAmount = 50;
```

The value for mRNA defaults to 0.

- 4 Check the initial amounts for the species.

```
Mobj.Species
```

MATLAB displays a summary list.

```
Species Object Array
```

Index:	Name:	InitialAmount:	InitialAmountUnits:
1	DNA	50	
2	mRNA	0	

Adding the Reaction for Translation

A simple model of translation shows only the mRNA and protein product. For more details, see “Translation” on page 2-4.

- 1 Enter the reaction mRNA → mRNA + protein with reaction rate equation $v = k \cdot \text{mRNA}$.

```
Robj2 = addreaction(Mobj, 'mRNA -> mRNA + protein');  
Kobj2 = addkineticlaw(Robj2, 'MassAction');
```

- 2 Define the reaction rate constant $K_2 = 20 \text{ 1/second}$.

```
Pobj2 = addparameter(Kobj2, 'k2', 20.0);  
set(Kobj2, 'ParameterVariableNames', 'K2');
```

Adding the Reactions for Gene Regulation

Transcription of DNA to mRNA is regulated by the binding of the protein product from translation to the DNA. As more protein is produced, the DNA is bound with the protein more often and less time is available for transcription with the unbound DNA. For more details, see “Gene Repression” on page 2-4.

- 1 Enter the forward reaction for DNA + protein → DNA_{protein} with a forward reaction rate equation $v = K_3 \cdot \text{DNA} \cdot \text{protein}$ and a reverse rate $v = K_{3r} \cdot \text{DNA}_{\text{protein}}$.

```
Robj3 = addreaction(Mobj, 'DNA + protein -> DNA_protein');  
Kobj3 = addkineticlaw(Robj3, 'MassAction');  
Pobj3 = addparameter(Kobj3, 'k3', 0.2);  
set(Kobj3, 'ParameterVariableNames', 'k3');
```

- 2 Enter the reverse reaction.

```
Robj3r = addreaction(Mobj, 'DNA_protein -> DNA + protein');  
Kobj3r = addkineticlaw(Robj3r, 'MassAction');  
Pobj3r = addparameter(Kobj3r, 'k3r', 1.0);  
set(Kobj3r, 'ParameterVariableNames', 'k3r');
```

Adding the Reactions for mRNA and Protein Degradation

Protein and mRNA degradation are important reactions for regulating gene expression. The steady-state level of the compounds is maintained by a balance between synthesis and degradation reactions. Proteins are hydrolyzed to amino acids with the help of proteases, while nucleic acids are degraded to nucleotides.

- 1 Enter the reaction for mRNA degradation to nucleotides.

```
Robj5 = addreaction(Mobj, 'mRNA -> null');
Kobj5 = addkineticlaw(Robj5, 'MassAction');
Pobj5 = addparameter(Kobj5, 'k5', 1.5);
set(Kobj5, 'ParameterVariableNames', 'k5');
```

- 2 Enter the reaction for protein degradation to amino acids.

```
Robj6 = addreaction(Mobj, 'protein -> null');
Kobj6 = addkineticlaw(Robj6, 'MassAction');
Pobj6 = addparameter(Kobj6, 'k6', 1.0);
set(Kobj6, 'ParameterVariableNames', 'k6');
```

Simulating the Model with an ODE Solver

After you create the model, enter reactions, and set parameters, you can simulate the dynamic behavior of a model.

- 1 Get a listing of the configuration set attached to the model.

```
cs = getconfigset(Mobj)
```

The configuration set object defines the parameters needed for a simulation.

MATLAB displays a listing of the configuration set property values.

```
Configuration Settings - default (active)
SolverType:                ode15s
StopTime:                  10.000000

SolverOptions:
AbsoluteTolerance:        1.000000e-006
```

```
RelativeTolerance:      1.000000e-003
SensitivityAnalysis:    false

RuntimeOptions:
StatesToLog:           4

CompileOptions:
UnitConversion:        false
DimensionalAnalysis:   true

SensitivityAnalysisOptions:
ParameterInputFactors: 0
SpeciesInputFactors:   0
```

2 Set the final simulation time.

```
set(cs, 'StopTime', 8.0);
```

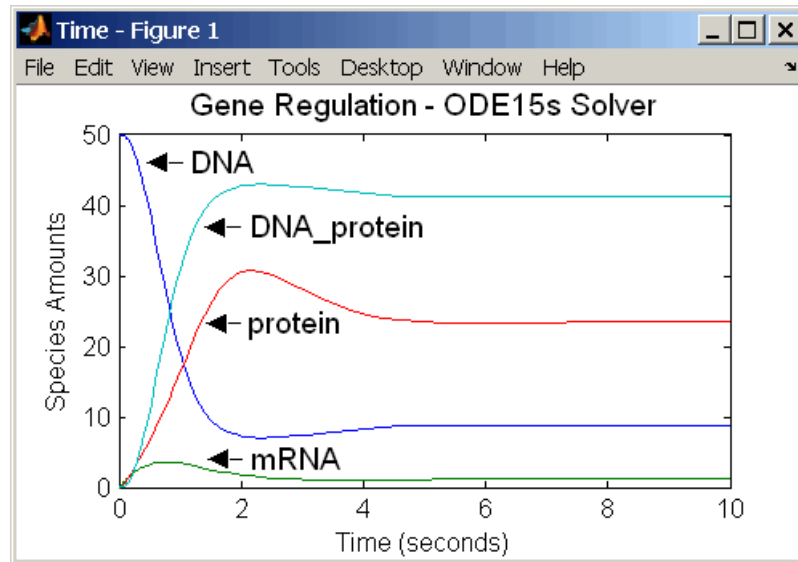
3 Run the simulation.

```
[t_ode, x_ode] = sbiosimulate(Mobj, cs);
```

4 Plot the results.

```
figure(1);
set(gcf, 'color', 'white');
plot(t_ode, x_ode(:,1:4));
title('Gene Regulation - ODE15s Solver');
xlabel('Time (seconds)');
ylabel('Species Amounts');
axis([0, 8, 0, 50]);
```

MATLAB draws a figure with the simulation results.



Simulating the Model with a Stochastic Solver

SimBiology includes three stochastic solvers. The stochastic solvers more accurately calculate the change in species amounts with a small number of molecules.

- 1 Get the active configuration set for a model.

```
cs = getconfigset(Mobj);
```

- 2 Select the stochastic solver and list the configuration set properties.

```
cs = getconfigset(Mobj);
set(cs, 'SolverType', 'ssa');
cs
```

MATLAB displays properties for the current configuration set. Notice that the SolverOptions for the stochastic solver are different from the ODE options.

```
Configuration Settings - default (active)
SolverType:          ssa
StopTime:           10.000000
```

```
SolverOptions:
  AbsoluteTolerance:      1.000000e-006
  RelativeTolerance:      1.000000e-003
  SensitivityAnalysis:    false

RuntimeOptions:
  StatesToLog:            4

CompileOptions:
  UnitConversion:         false
  DimensionalAnalysis:    true

SensitivityAnalysisOptions:
  ParameterInputFactors:  0
  SpeciesInputFactors:    0
```

3 Change the values for the `StopTime` and `LogDecimation` properties.

```
cs.SolverOptions.LogDecimation = 10
cs.StopTime = 10;
```

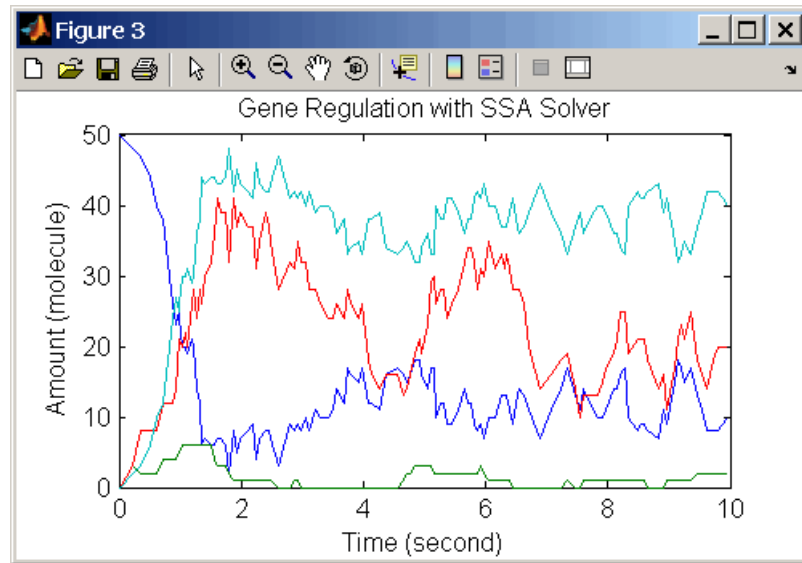
4 Run the simulation.

```
[t_ssa, x_ssa] = sbiosimulate(Mobj, cs);
```

5 Plot the results.

```
figure(2);
set(gcf, 'color', 'white');
plot(t_ssa,x_ssa(:,1:4));
title('Gene Regulation with SSA Solver');
xlabel('Time (second)');
ylabel('Amount (molecule)');
set(gcf, 'color', 'white');
axis([0, 10, 0, 50]);
```

MATLAB draws a figure with the results from the simulation. Add labels by selecting **Text Arrow** from the **Insert** menu.



G

gene regulation
 GUI desktop 2-7
 GUI Diagram 2-16
 model digram 2-2
 reaction list 2-2
 reactions 2-3
 tutorial model 2-2

K

Kinetic Law
 object definition 1-18

M

model
 adding reactions 2-9 2-19 2-37
 creating 2-7 2-16
 creating command-line 2-35
 gene regulation 2-38
 project file 2-9 2-19
 protein degradation 2-39
 simulating 2-14 2-23
 translation 2-38

P

Parameters
 definition 1-21
 object definition 1-21

R

Rate constants, *see* Parameters
Reaction
 object definition 1-18
Reaction rate

definition 1-18

reactions
 degradation 2-39
 gene regulation 2-38
 transcription 2-37
 translation 2-38

Rules
 object definition 1-23

S

SimBiology
 computation with MATLAB 1-2
 defined 1-2
 features and functions 1-5
 object definitions 1-16
 optional software 1-3
 related software 1-4
 required software 1-3
 SBML support in 1-13
 visualizing data 1-2
simulation
 ODE solver 2-39
 stochastic solver 2-41
software
 optional 1-3
 required 1-3
Species
 object definition 1-19

T

tutorial
 basic 2-1
 command-line 2-35
tutorial models
 gene regulation 2-2