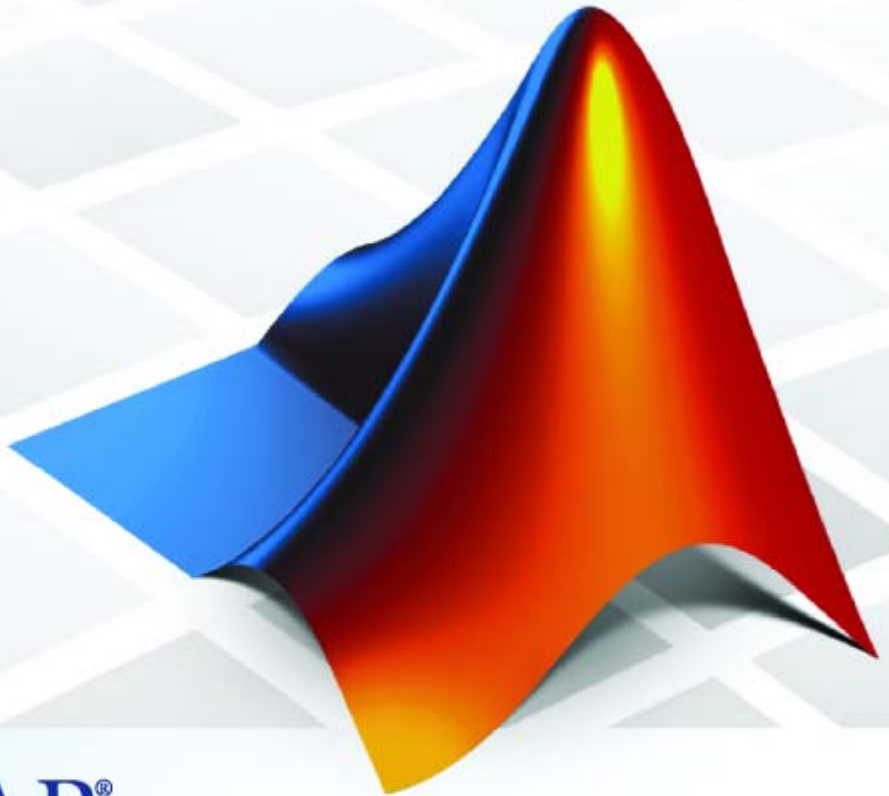


Getting Started with **SimBiology** 2



MATLAB[®]

How to Contact The MathWorks



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



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

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



508-647-7000 (Phone)



508-647-7001 (Fax)



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

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

Getting Started with SimBiology

© COPYRIGHT 2005–2007 by The MathWorks, Inc.

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

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

Trademarks

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

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

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents 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+)
September 2006	Online only	Updated for Version 2.0.1 (Release 2006b)
September 2006	Online only	Updated for Version 2.1 (Release 2006b+)
March 2007	Online only	Rereleased for Version 2.1.1 (Release 2007a)
September 2007	Online only	Rereleased for Version 2.1.2 (Release 2007b)
October 2007	Online only	Updated for Version 2.2 (Release 2007b+)

Introduction

1

What Is SimBiology®?	1-2
Overview	1-2
Expected Users	1-3
Required Software	1-4
Optional Software	1-4
Related Software	1-5
Integrating SimBiology into Your Existing Workflow ..	1-6
Existing Workflow in Experimental Research	1-6
Workflow Incorporating Mathematical Modeling	1-7
Workflow Incorporating SimBiology for Mathematical Modeling	1-9
Getting Started in the SimBiology Desktop	1-11
Start Building a Model	1-11
Opening a SimBiology Project	1-12
Importing an SBML Model	1-13
Features and Functions	1-14
Working in the SimBiology Desktop Versus the MATLAB Command Window	1-15
Graphical User Interface	1-15
Command-Line Interface	1-19
Data Formats and Projects	1-19
Modeling	1-20
Simulation	1-21
Analysis	1-22
SimBiology and SBML	1-23
What Is SBML?	1-23
SBML Support in SimBiology	1-23
SimBiology Model Components	1-25
Organization of SimBiology Objects	1-26

Model	1-27
Compartment	1-27
Species	1-27
Reaction	1-28
Kinetic Law	1-28
Parameters	1-29
Rules	1-30
Events	1-31
Configuration Sets	1-31
Variants	1-32

Building Models Using Graphical and Command-Line Features

2

Gene Regulation Model	2-2
Model Diagram	2-2
Model Reactions	2-3
Opening the Gene Regulation Model in SimBiology	2-6
Modeling Using the SimBiology Graphical User	
Interface	2-8
About This Example	2-8
Creating a SimBiology Model	2-8
Saving Your Work as a SimBiology Project File	2-9
Adding Reactions to a Model	2-10
Verifying the Model	2-14
Simulating a Model	2-15
Modeling Using the SimBiology Diagram	2-17
Creating a SimBiology Model	2-17
Saving Your Work as a SimBiology Project File	2-19
Adding Reactions to a Model Diagram	2-19
Simulating a Model	2-24
Setting Block Appearance and Diagram Layout	2-25
Printing and Exporting the Diagram	2-36
Modeling Using the Command-Line Interface	2-39
About This Example	2-39
Creating a SimBiology Model	2-39

Adding the Reaction for Transcription	2-41
Adding the Reaction for Translation	2-42
Adding the Reactions for Gene Regulation	2-42
Adding the Reactions for mRNA and Protein	
Degradation	2-43
Simulating the Model with an ODE Solver	2-44
Simulating the Model with a Stochastic Solver	2-45

Index



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
Integrating SimBiology into Your Existing Workflow (p. 1-6)	Description and diagrammatic representation of how SimBiology fits into an existing workflow
Getting Started in the SimBiology Desktop (p. 1-11)	How to start building a model, open a project, or import an SBML model
Features and Functions (p. 1-14)	Functions grouped into categories that support systems biology tasks
SimBiology and SBML (p. 1-23)	Support for the Systems Biology Markup Language (SBML) specification and extended features in SimBiology
SimBiology Model Components (p. 1-25)	Definition of the SimBiology object structure

What Is SimBiology®?

In this section...
“Overview” on page 1-2
“Expected Users” on page 1-3
“Required Software” on page 1-4
“Optional Software” on page 1-4
“Related Software” on page 1-5

Overview

SimBiology provides 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 events 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 1-20.

- **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 1-21.

- **Analyze** — Save data from a simulation, compare simulation and experimental data, perform sensitivity analysis, species or parameter scans, parameter estimation, and search for conserved moieties. For more information, see “Analysis” on page 1-22.

Expected Users

The people who use SimBiology come from a wide range of areas including biology, systems biology, pharmacology, computer science, and engineering. SimBiology is intended for research scientists, computational biologists, and students who need to develop and study biological pathways at the molecular and systems level, develop custom analysis applications, or implement published ones, and visualize results.

Industry/Professional — SimBiology allows you to model, simulate, and analyze biochemical and system pathways for applications in drug discovery and design, and target identification.

Modeling, simulating, and analyzing a biological system can test hypotheses for a pathway, identify side effects caused by drug interactions with a target compound, and identify biochemical pathways that lead to disease.

Academia — Build rigorous, dynamic, quantitative models that allow you understand and predict system behavior at the molecular level. Build models to explore enzyme kinetics. Leverage the MATLAB foundation to easily manipulate large data sets.

Simulating the dynamic behavior of a model can confirm the validity of the models and identify behaviors and control mechanisms not apparent from studying static models. Validate models experimentally and use the model to predict *in vitro* and *in vivo* behavior.

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 this toolbox is not installed, SimBiology uses the optimization algorithms available in MATLAB.

Optimization Toolbox

Solve standard and large-scale optimization problems. If Optimization Toolbox is installed, you can use some algorithms included in Optimization Toolbox for parameter estimation in SimBiology. If 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.

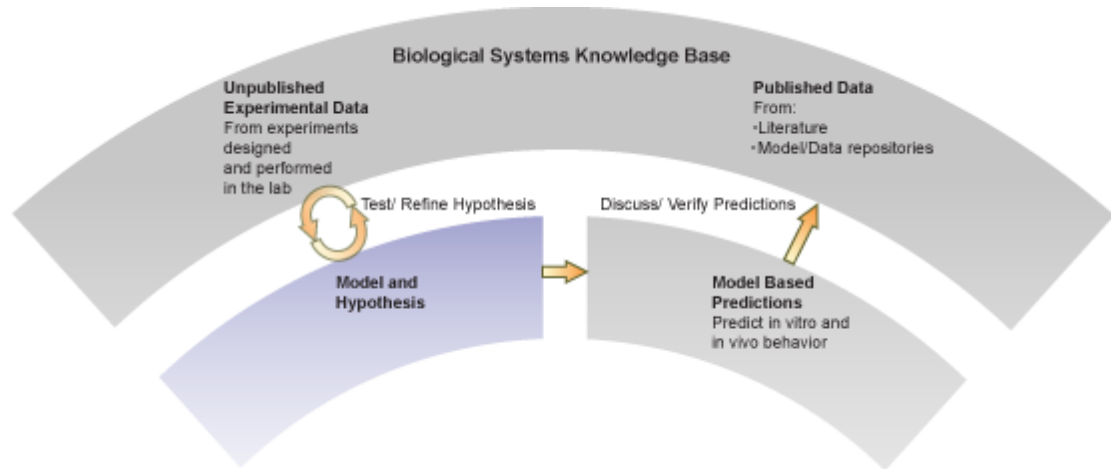
Integrating SimBiology into Your Existing Workflow

In this section...
“Existing Workflow in Experimental Research” on page 1-6
“Workflow Incorporating Mathematical Modeling” on page 1-7
“Workflow Incorporating SimBiology for Mathematical Modeling” on page 1-9

Existing Workflow in Experimental Research

Consider the following typical workflow in research. The figure shows the following general steps in experimental research.

- 1** Use preliminary results and published data to develop a model and hypothesis in your area of interest.
- 2** Validate your model by making predictions and designing experiments to test your model and hypothesis.
- 3** Make further predictions based on your model.
- 4** Publish results and predictions.
- 5** Continue to test predictions experimentally.

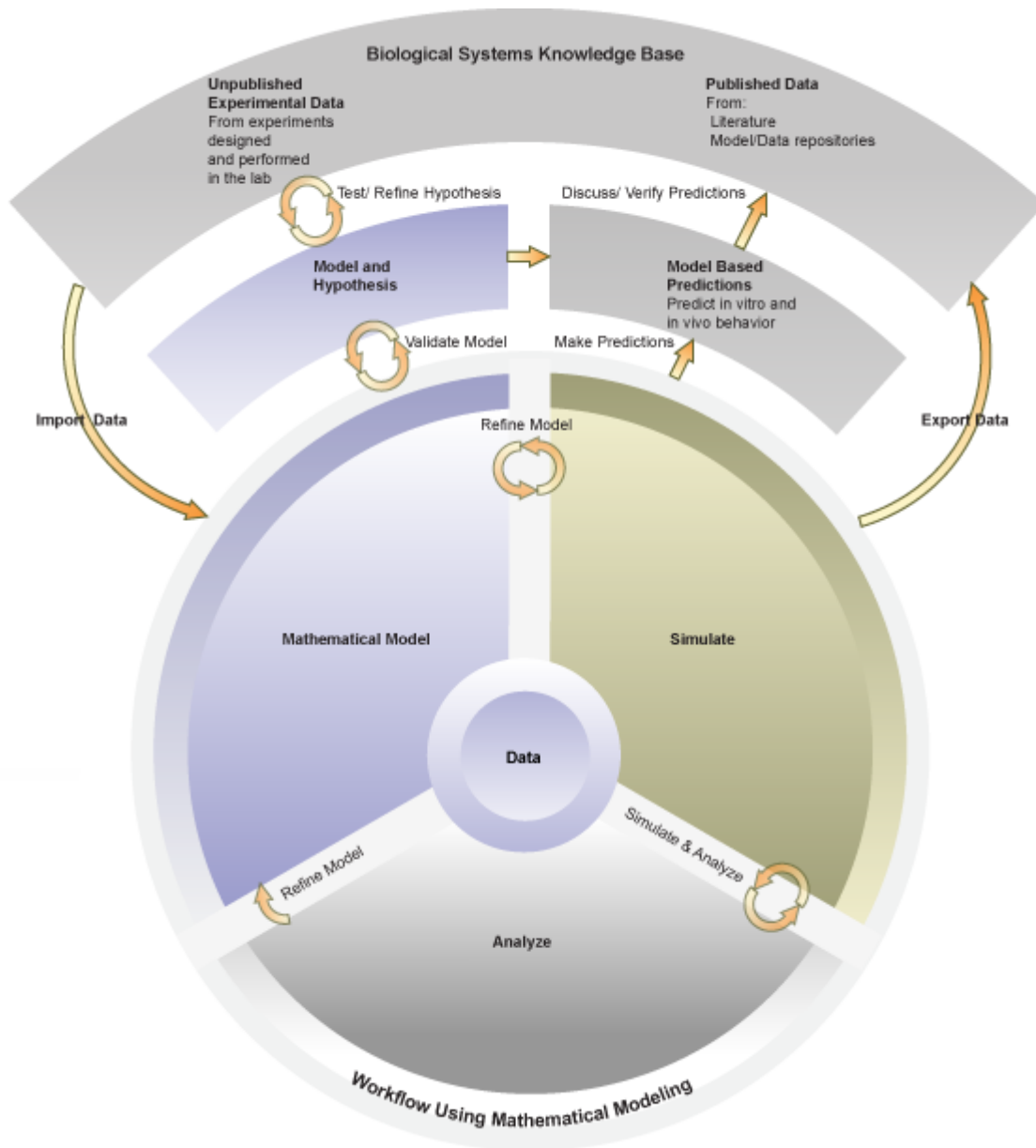


Typical Workflow in Experimental Research

Workflow Incorporating Mathematical Modeling

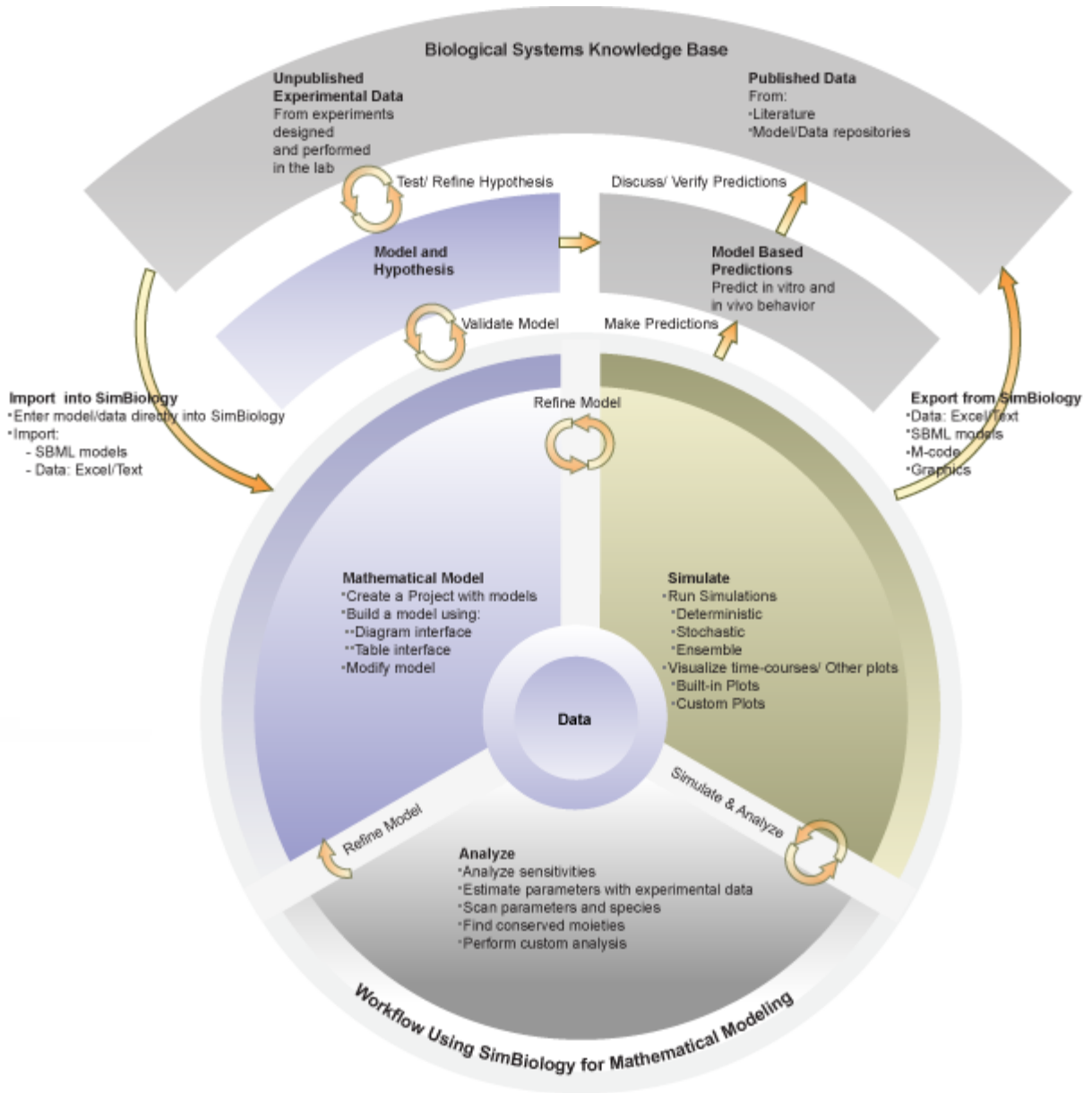
Mathematical modeling integrates into the preceding workflow by providing the means to:

- Consolidate quantitative data into the model.
- Represent large pathways and networks and lay them out in meaningful ways using a software application.
- Study emergent properties of pathways and use these predictions to guide the direction of experimental research in ways that can save time and development costs.



Workflow Incorporating SimBiology for Mathematical Modeling

SimBiology provides an integrated environment for mathematical modeling and analysis of biological and biochemical pathways. The following diagram shows you an overlay of features and tasks in SimBiology pictured in the context of the complete workflow in experimental research.



Getting Started in the SimBiology Desktop

In this section...
“Start Building a Model” on page 1-11
“Opening a SimBiology Project” on page 1-12
“Importing an SBML Model” on page 1-13

Depending on your application area and the extent of the existing knowledge base for your area of interest and ease of data availability, you might choose various different paths into, within, and out of SimBiology. For example:

- Build the model in SimBiology; use published or unpublished kinetic data that you directly enter into SimBiology or import the kinetic data using Excel or text files, and then perform simulation and analysis.
- Open and work with a SimBiology model (project) that you have obtained.
- Import an SBML format model into SimBiology and perform simulation and analysis.

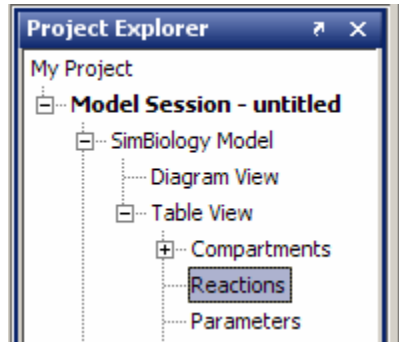
Start Building a Model

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

```
sbiodesktop
```

Alternatively, open MATLAB and click the following link, SimBiology Desktop. The SimBiology desktop window opens with an empty project.

- 2 From the **File** menu, select **New Model Session**.
- 3 Enter a pathway in the form of reactions. To enter a reaction:
 - a In the **Project Explorer**, expand **Table View** and double-click **Reactions** to open the **Reactions** pane.



- b Follow the steps described in 'Adding a New Reaction', in SimBiology Desktop Help.



If SimBiology Desktop Help is not already open, select **Help > SimBiology Desktop Help** to open the help.

Opening a SimBiology Project

To open a SimBiology project that you received,

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

```
sbiodesktop
```

- 2 Select **File > Open Project**. The Open SimBiology Project dialog box opens.
- 3 Browse to and select the `.sbproj` format file containing the project.

Importing an SBML Model

To import a Systems Biology Markup Language (SBML) model into SimBiology,

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

```
sbiodesktop
```

- 2 Select **File > Import Model > From SBML File**. The Load Systems Biology Markup Language File dialog box opens.
- 3 Browse to and select the `.xml` file containing the SBML model.
- 4 Click **Open** to open the model.

Features and Functions

In this section...
“Working in the SimBiology Desktop Versus the MATLAB Command Window” on page 1-15
“Graphical User Interface” on page 1-15
“Command-Line Interface” on page 1-19
“Data Formats and Projects” on page 1-19
“Modeling” on page 1-20
“Simulation” on page 1-21
“Analysis” on page 1-22

Working in the SimBiology Desktop Versus the MATLAB Command Window

SimBiology includes an extensive desktop GUI that allows you to accomplish modeling, simulation, and analysis tasks. You can also access most of these functions and features programmatically through the MATLAB command window allowing you to write scripts for the same tasks.

In addition, the desktop shows you the equivalent M-code that is being used to execute the actions performed on the desktop. This feature allows you to quickly get started using the tool through the desktop and switch to the command-line if required.

There are two tutorials in the Getting Started guide that walk you through building models through the desktop:

- “Modeling Using the SimBiology Graphical User Interface” on page 2-8
- “Modeling Using the SimBiology Diagram” on page 2-17

There is one tutorial in the Getting Started guide that walks you through building models at the command-line:

- “Modeling Using the Command-Line Interface” on page 2-39

There are video demos that show you how to work in the desktop and M-file demos that show you how to programmatically build a model, simulate, and analyze. To access SimBiology demos type the following at the MATLAB command-line,

```
demo matlab simbiology
```

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.
- **Diagram** — Enter model data using block representations for species, and reactions. Use the Plot block to visualize simulation data during a simulation.
 - **Block Library Browser** — View, and use blocks from the built-in **Basic** library, add custom blocks to user-defined block libraries for use in the diagram.
 - **Alignment Tool** — Align blocks in a diagram.
- **Table View** — 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.
- **Simulation** — Simulate a model with deterministic or stochastic solvers, save simulation data, export data to the MATLAB workspace.
- **Analysis Tasks** — Calculate sensitivities, scan species or parameters, find conserved moieties, perform ensemble simulations, or create custom analysis tasks.
- **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.
- **Report Templates** — Create reports for your project, include, diagrams, figures, and data in the reports, and save as HTML files.

- **Output** — Shows you the results of Find, error or warnings generated by Verify (model verification and validation), and errors or warnings generated by Simulation.
- **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.
- **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.

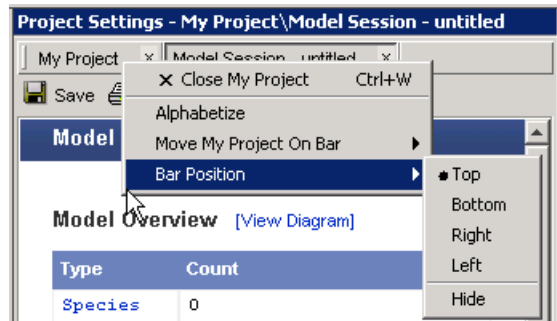
Customizing Desktop Views and Preferences

You can arrange the desktop layout according to your preference.

- **Changing preferences** – Select **File > Preferences**, to open the SimBiology Preferences dialog box. You can change preferences for the following:
 - **Desktop Display** — which toolbars to display and which properties to display in model component tables)
 - **Project Explorer** — Specify whether to double-click or single-click to open each pane from the **Project Explorer**
 - **Diagram** — Specify **Indicator** options, and whether to always warn when there are disallowed block actions.

- **Search** — Specify whether double-clicking a search result should open the selected component in the table view or the diagram.
- **Report Generator** — Specify default format for report output, stylesheet to use when generating the report, default image formats, size, and location.
- **History** — Specify number of recently used models, projects, help queries, searches, and libraries to display.
- **Confirmation Dialogs** — Specify whether dialog boxes should open for confirmation of various deletions, parameter reordering, defining undefined variables.
- **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.

- **Functions** — Use functions to create projects, manage the unit and kinetic law libraries, simulate, and analyze 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, 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 1-23.

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 pathway models

- **Model** — A SimBiology *model* is a collection of interrelated reactions, rules, and events that transform, transport, and bind species. The model components are compartments, species, reactions, parameters, rules, and events. After creating a model, add a compartment, then you can add reactions to the model, define a kinetic law for each reaction, and add the parameters for the reaction.
- **Model properties** — The properties define the characteristics of a component. For example, a reaction includes properties for reactions such as reaction rates, and links to associated species, parameters, and kinetic laws. You can interactively change these properties in the graphical user interface.

At the command-line, all model components are represented as objects. Use the MATLAB `get` and `set` commands to list object properties and change their values at the command line.

- **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 stochastic and deterministic 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.

Analysis

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

In this section...
“What Is SBML?” on page 1-23
“SBML Support in SimBiology” on page 1-23

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. You can import from and export to SBML-compatible .xml files.

Unsupported features include:

- 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 `hasOnlySubstanceUnits` field does not have a corresponding property in the SimBiology species object. For more information on dimensions for SimBiology species, see `DefaultSpeciesDimension`.

The following features in SimBiology are not supported by SBML:

- **Projects** — SimBiology creates and stores all your models in project files with the `.sproj` extension. You can store multiple models in one project file.
- **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 Henri-Michaelis-Menten equation,

$$V_m * S / (K_m + S)$$

is a kinetic law stored in the kinetic law library.

Specifically, for a one-substrate enzyme-catalyzed reaction, if you assign $V_m = v$, $K_m = K_a$, 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, **Tag**, and **Active** cannot be exported.

SimBiology Model Components

In this section...

“Organization of SimBiology Objects” on page 1-26

“Model” on page 1-27

“Compartment” on page 1-27

“Species” on page 1-27

“Reaction” on page 1-28

“Kinetic Law” on page 1-28

“Parameters” on page 1-29

“Rules” on page 1-30

“Events” on page 1-31

“Configuration Sets” on page 1-31

“Variants” on page 1-32

Organization of SimBiology Objects

If you plan to work mainly from the SimBiology desktop GUI you can skip this section and go to “Model” on page 1-27. The GUI allows you to interactively work with SimBiology model components without having prior knowledge of object-oriented programming, or the object implementation thereof in SimBiology.

If however, you plan on using SimBiology at the command-line, it is useful understand the hierarchical organization of SimBiology objects.

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

Models and each model component in SimBiology are represented as objects. Further, there are objects to store simulation settings (configset object) and data from simulation and analysis tasks (data object)

Models must contain at least one species object and one compartment object. The hierarchy of objects in models is as follows:

```
Model
  Compartment
    Species
  Parameters
  Reaction
    Kinetic Law
      Parameters
  Rules
  Events
  Variant
  Configset
```

```
SimData
```

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, rules, and events that transform, transport, and bind species. The model components are compartments, species, reactions, parameters, rules, and events. After creating a model, add a compartment, then you can add reactions to the model, define a kinetic law for each reaction, and add the parameters for the reaction.

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.

Compartment

A *compartment* allows you to associate pools of species to physically isolated regions.

Since compartments define physically isolated regions, species are assigned to compartments. Reactions, rules, events, and parameters are assigned to the model.

All models must have at least one compartment and all species in a model must be assigned to a compartment. You can create a compartment within another compartment, but compartment names must be unique within a model. For information on specifying names of compartments see `Name` in the SimBiology Reference.

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.

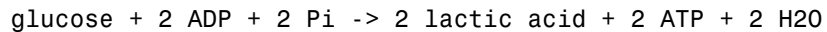
For information on specifying names of species see `Name` in the SimBiology Reference.

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.



Enter a reaction in the form



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.

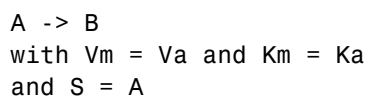
The `ReactionRate` is the result of defining the names of parameters and species variables in a rate law. To get `ReactionRate`, the species variables and parameter variables that participate in the reaction rate should be clearly assigned in the **Project Settings-Reactions** pane **Kinetic Law** tab.

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

$$V_m * S / (K_m + S)$$

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.

Reaction Rate Equation

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

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

Parameters

A *parameter* is a quantity that can change or can be constant. Parameters are generally used to define rate constants. But, you could also use parameters as constants to modify model component values in rules and events.

For information on specifying names of parameters see Name in the SimBiology Reference.

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, event, 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 events that reference the parameter.

Rules

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

- **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 `species_total = species1 + species2` where `species_total` is the independent variable, write the rule as

```
species1 + species2 - species_total
```

- **initialAssignment** — `initialAssignment` rules are evaluated once at the beginning of a simulation. `initialAssignment` rules are expressed as `Variable = Expression`. For example you could write an `initialAssignment` rule to set the amount of `species1` to be proportional to `species2`.

```
species1 = k/species2
```

- **repeatedAssignment** — `repeatedAssignment` rules are evaluated at every time-step during a simulation. `repeatedAssignment` rules are expressed as `Variable = Expression`. For example, you could use the rule to specify the amount of `species1` to always be proportional to `species2`.

```
species1 = k/species2
```

- **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 `Variable = 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.

For more information see “Rules” in the SimBiology User’s Guide.

Events

Events are used to describe sudden changes in model behavior. An event lets you specify discrete transitions in model component values that occur when a user-specified condition become true. You can specify that the event occurs at a particular time, or specify a time-independent condition. For example, you can use events to activate or deactivate certain species (activator or inhibitor species), change parameter values, change reaction rates in response to addition or removal of species. For more information about events, evaluation of events, and examples see, “Events”.

Configuration Sets

Configuration sets contain the options that the solver uses during simulation of a model. To add, view or edit configuration sets associated with a model, in the **Project Explorer** expand **Model Variable Settings** and double-click **Configuration Settings**. The configuration set contains the following options for you to choose:

- Type of solver
- Data to record (**Data Logging**)
- Stop time for the simulation
- Solver error tolerances, and for ode solvers – the maximum time step the solver should take

- Whether to perform sensitivity analysis during simulation
- Whether to perform dimensional analysis and unit conversion during simulation
- Species and parameter input and output factors for sensitivity analysis

Variants

Variants let you store the names and values of model components and allow you to use the values stored in the variant as the alternate value to be applied during a simulation. To add, view or edit variants associated with a model, in the **Project Explorer** expand **Model Variable Settings** and double-click **Variants**. You can store values for species InitialAmount, parameter Value, and compartment Capacity, in variants.

Using one or more variants associated with a model allows you to evaluate model behavior during simulation, with different values for the various model components without having to search and replace these values, or having to create additional models with these values. If you determine that the values in a variant accurately define your model, you can permanently replace the values in your model with the values stored in the variant object, by clicking **Commit**.

Building Models Using Graphical and Command-Line Features

This chapter will get you started with creating and simulating dynamic models with SimBiology.

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. 2-8)

Model created interactively with the SimBiology desktop

Modeling Using the SimBiology Diagram (p. 2-17)

Model created interactively with the SimBiology diagram interface

Modeling Using the Command-Line Interface (p. 2-39)

Model created interactively with the MATLAB command-line interface

Gene Regulation Model

In this section...

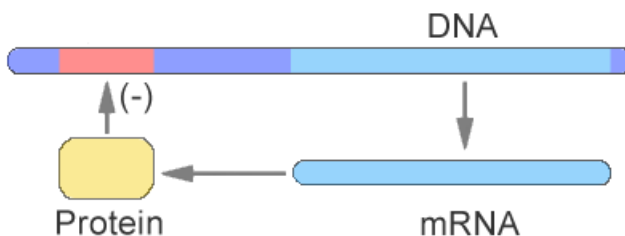
“Model Diagram” on page 2-2

“Model Reactions” on page 2-3

“Opening the Gene Regulation Model in SimBiology” on page 2-6

Model Diagram

You can visualize a systems biology model with various levels of detail. One view sketches only the major species and processes. 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. 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.



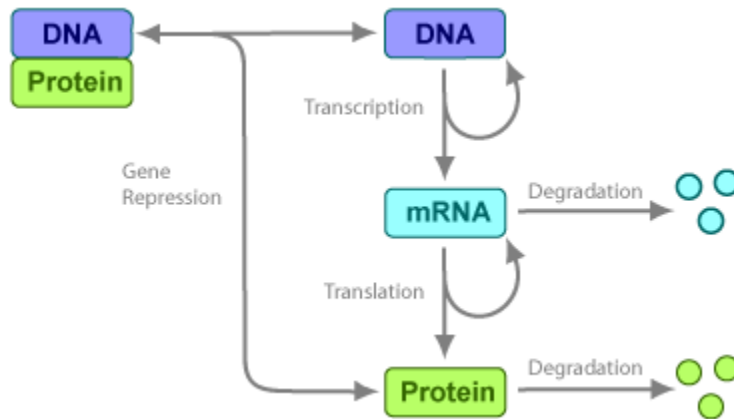
Another way of looking at a systems biology model is to list the reactions in a model with the processes they represent.

```

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 = k3*DNA*protein molecule/second
Species:     DNA = 50 molecule
             protein = 0 molecule
Parameters:  k3 = 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 $1/\text{mole-second}$. However, software evaluations of these common ways are sometimes ambiguous.

Reverse Reaction (Unbinding)

```

Reaction: DNA_protein -> DNA + protein
Reaction rate: k3r*DNA_protein molecule/second
Species:     DNA_protein = 0 molecule
Parameters:  k3r = 1 1/second

```

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

```

Reaction: DNA + protein <-> DNA_protein
Reaction rate: v = k3*DNA*protein - k3r*DNA_protein molecule/second
Species:     DNA = 50 molecule
             protein = 0 molecule
Parameters:  k3 = 0.2 1/(molecule*second)
             k3r = 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 = k4*mRNA molecule/second
Species: mRNA = 0 molecule
Parameters: k4 = 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 = k5*protein molecule/second
Species: protein = 0.0 molecule
Parameters: k5 = 1.0 1/second
```

Opening the Gene Regulation Model in SimBiology

A project containing the model built in gene regulation model example is available in SimBiology. To skip the sections on model building and to learn about simulations in SimBiology, you can access the completed model in the SimBiology desktop by doing the following steps:

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

```
sbiodesktop
```

The SimBiology desktop window opens.

- 2 From the **File** menu, select **Open Project**.
- 3 In the Open SimBiology Project dialog box navigate to

`matlabroot/toolbox/simbio/simbiodeemos`

- 4** Select `gene_reg.sbproj`, and click **Open**.

The project opens with two models `cell` and `cell2`. These two models are identical except for the graphical representation of the blocks in the **Diagram**.

- 5** To explore the models, double-click an item in the **Project Explorer**, to open individual panes.
- 6** See “Simulating a Model” on page 2-15 for procedures on simulating a model in the SimBiology desktop.

Modeling Using the SimBiology Graphical User Interface

In this section...

“About This Example” on page 2-8

“Creating a SimBiology Model” on page 2-8

“Saving Your Work as a SimBiology Project File” on page 2-9

“Adding Reactions to a Model” on page 2-10

“Verifying the Model” on page 2-14

“Simulating a Model” on page 2-15

About This Example

This section shows how to mathematically model a simple gene-regulation pathway using the SimBiology desktop. The example uses the model described in “Gene Regulation Model” on page 2-2 and shows you how to build this model using a spreadsheet style interface in the SimBiology desktop.

If you prefer to interact with diagrammatic representations of pathways, see “Modeling Using the SimBiology Diagram” on page 2-17.

Alternatively, you can use M-code and the MATLAB command-line interface (see “Modeling Using the Command-Line Interface” on page 2-39). 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

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.

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

```
sbiodesktop
```

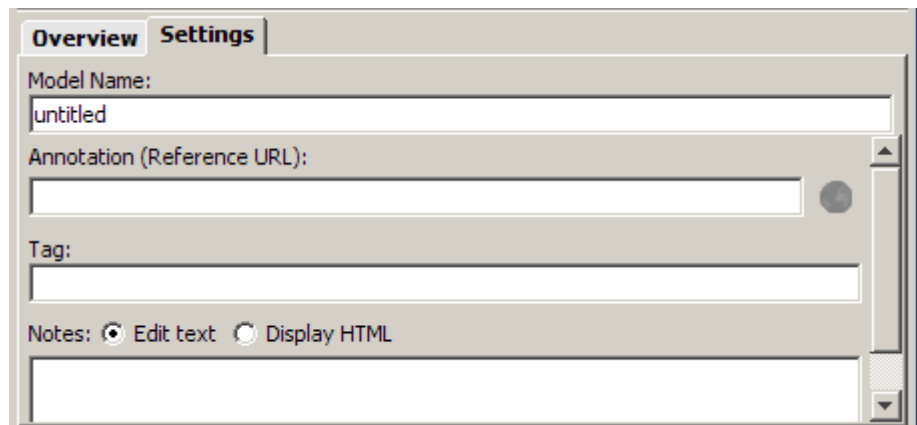
The SimBiology desktop window opens.

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

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

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

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



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

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 1-19.

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**, expand **Table View** and 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

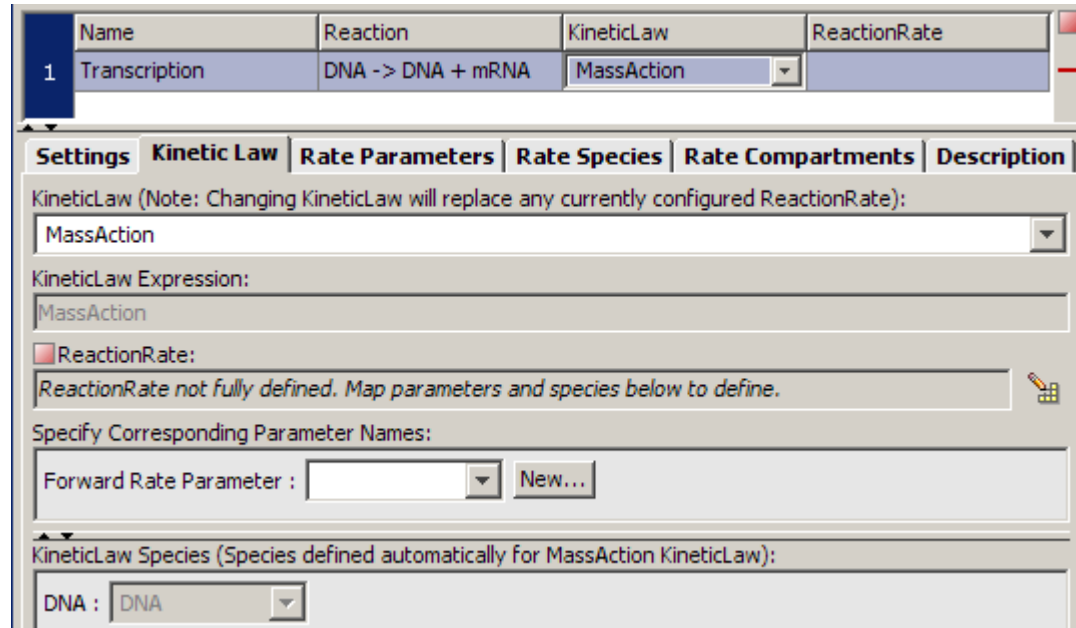
- 3 Click **Add**.

Notice that a colored indicator (red) appears to the right of the table. Move the pointer over the indicator for the reaction, to get more information about the reason for the indicator. In this case, the indicator shows that the reaction rate is as yet undefined. Beginning in step 5 of this example you will learn how to define the reaction rate.

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

Transcription

- 5 From the **KineticLaw** list, select **MassAction**.
- 6 Click the **Kinetic Law** tab. Your screen should now look similar to this one:



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 DNA -> DNA + 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

Notice that the colored indicator is now green, showing that the reaction rate is now defined.

- 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
1	k1	DNA -> DNA + mRNA	0.2	1/second

- 10 In the **Project Explorer**, double-click **Compartments**, to open the **Compartments** pane.

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

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

- In the **Name** cell, double-click and type a name for the compartment and press **Enter**.

contents

The compartment table updates with the new entry. Leave other compartment properties such as **Owner**, and **Capacity** in the default settings, for the purposes of this example.

- 11** In the **Project Explorer**, expand **Compartments** and double-click **contents Species**. The species you entered in the reaction are automatically added to the species table.

The default value for a species is 0.0.

- 12** 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.
- 13** 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/(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

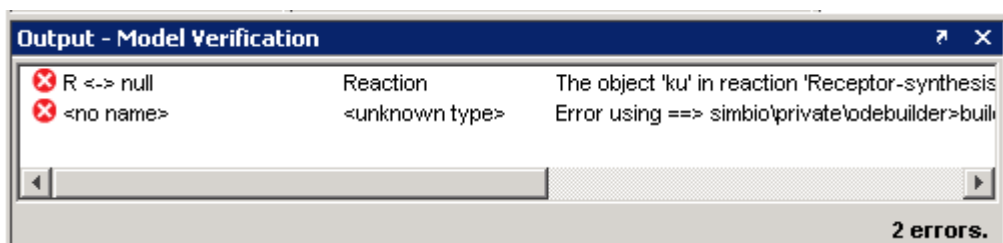
For more detailed information about the reactions in this model, see “Gene Regulation Model” on page 2-2.

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 is as yet undefined.



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

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

Trouble-Shooting Using Color Indicators

Model component tables have colored indicators that let you see if the specified properties are valid. Move the mouse over the indicator to see more

details about the warning or error. The table below shows the indicators and their significance:

Color Indicator	Description
Green	Components have valid properties
Yellow	There are warnings in the associated rows
Red	There are errors in the associated rows

- Click **Verify** in the **Toolbar** to see if any errors are listed in the **Output** pane. You must fix any reported errors before you can simulate a model. Warnings may not need to be resolved before simulating, but might result in unexpected answers.
- Click the colored square, if present, in the **Settings** tab to see a dialog box with tips on how to fix the warning or error.


Simulating a Model

After you enter the reactions for your model, you can simulate its dynamic behavior and plot the results.

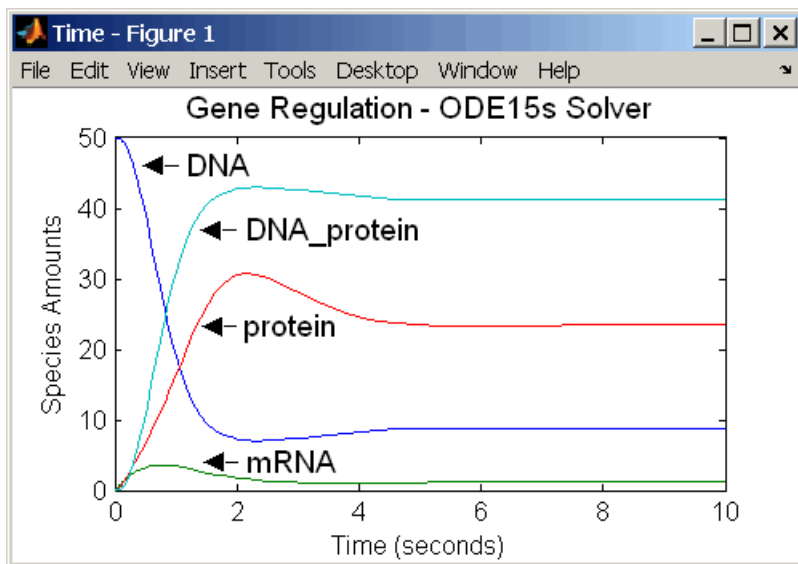
1 Add a **Simulation** task to the model.

- From the **Analysis** menu, select **Add Analysis Task to cell > Simulate model**

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

2 In the **Simulation** pane, click  (**Run**).

The simulation uses the default configuration settings, runs to completion and then plots the results in a figure window. You can annotate your plot in the figure window.



If you have more than one model session in your project, then you can specify which model should be **Active**. The **Active** setting lets you select actions on the model through menus, and also allows you to simulate the model from the **Diagram** pane. 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.
- 2 Select **Activate (model name)**.

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

Modeling Using the SimBiology Diagram

In this section...

“Creating a SimBiology Model” on page 2-17

“Saving Your Work as a SimBiology Project File” on page 2-19

“Adding Reactions to a Model Diagram” on page 2-19

“Simulating a Model” on page 2-24

“Setting Block Appearance and Diagram Layout” on page 2-25

“Printing and Exporting the Diagram” on page 2-36

Creating a SimBiology Model

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.

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. 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 compartment, species, and reaction. The model reactions are shown in “Gene Regulation Model” on page 2-2.

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**.

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

```
sbiodesktop
```

The SimBiology desktop window opens.

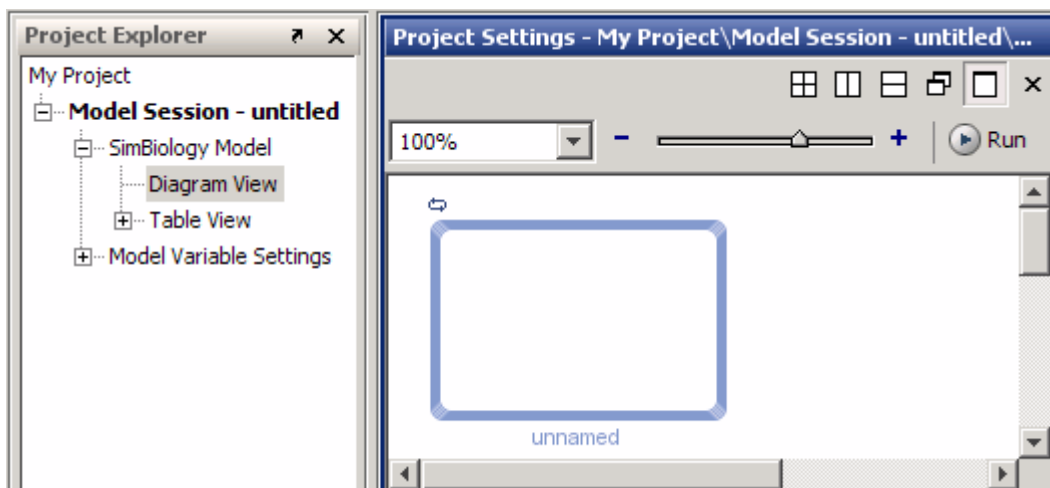
- 2 Select **File > New Project**.

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

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

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

- 4 In the **Project Explorer**, double-click **Diagram View**. SimBiology opens the **Diagram** pane.



Notice the Compartment block (*unnamed*) in the model. In the desktop, all models contain a compartment by default.

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

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

- 6 Double-click the **Compartment** block to open the Compartment Properties dialog box.

All models created in the desktop contain a compartment by default. The process of adding a reaction automatically adds the reaction species to the compartment in the model. If there are multiple compartments in the

model, you must specify the reactants and products using qualified names (*compartmentName.speciesName*). For example `nucleus.DNA` denotes the species DNA, in the compartment nucleus.

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

- In the **Name** cell, double-click and type a name for the compartment and press **Enter**.

```
contents
```

Leave other compartment properties such as **Owner**, and **Capacity** in the default settings, for the purposes of this example.

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 1-19.

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**.

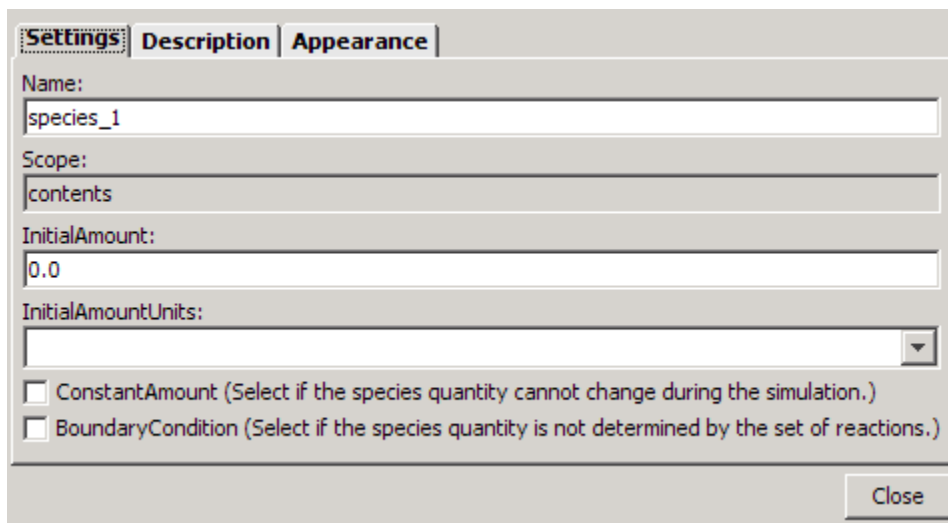
```
gene_regulation.sbproj
```

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 block library.

- 1 From the **Block Library Browser**, click and drag a Species block into the compartment in the diagram.

- 2 Double-click the Species block to open the Species Properties dialog box.



The screenshot shows the 'Species Properties' dialog box with the 'Settings' tab selected. The fields are as follows:

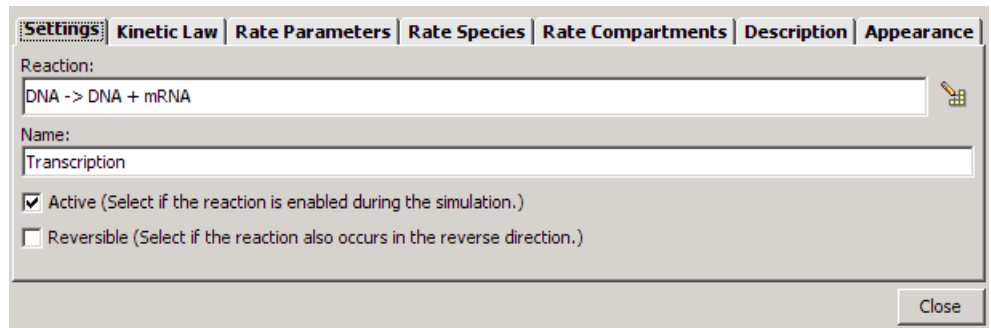
- Name:** species_1
- Scope:** contents
- InitialAmount:** 0.0
- InitialAmountUnits:** (empty dropdown menu)
- ConstantAmount (Select if the species quantity cannot change during the simulation.)
- BoundaryCondition (Select if the species quantity is not determined by the set of reactions.)

A 'Close' button is located at the bottom right of the dialog.

- 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 Click **Close**.
- 6 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.
- 7 From the **Block Library Browser**, click and drag a Reaction block into the diagram.
- 8 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. The diagram draws a dashed line, indicating that DNA is a modifier for the reaction

- 9 Draw a line from the Reaction block to the mRNA Species block.
- 10 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.
- 11 Click the **Name** box and give the reaction a name. For example, enter Transcription.



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

Notice that a colored indicator (red) appears next to **ReactionRate**. Click the indicator for the reaction, to get more information about the reason for the indicator. In this case, the indicator shows that the reaction rate is as yet undefined. In the following steps you will learn how to define the reaction rate.

- 13 From the **KineticLaw** list, select MassAction.
- 14 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 \rightarrow DNA + mRNA reaction.

c Click **Create**.

Notice that SimBiology updates **ReactionRate** for the reaction to show $k_1 \cdot \text{DNA}$, and the colored indicator is now green, showing that the reaction rate is now defined.

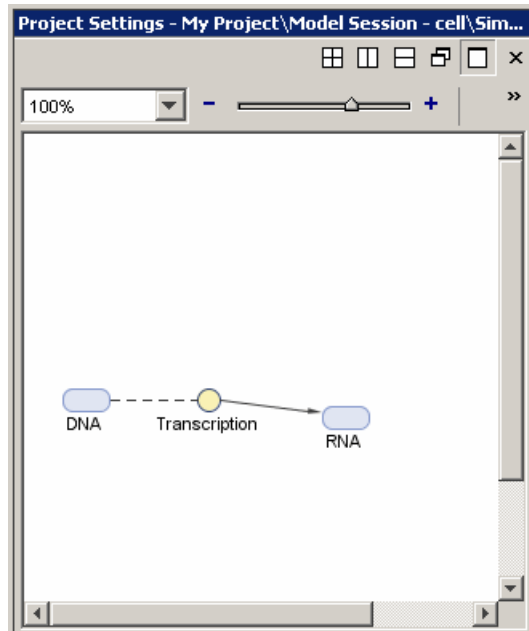
15 Click the **Rate Parameters** tab. Note that SimBiology lists k_1 with a default value of 1.0 for the parameter.

16 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 resemble the following:

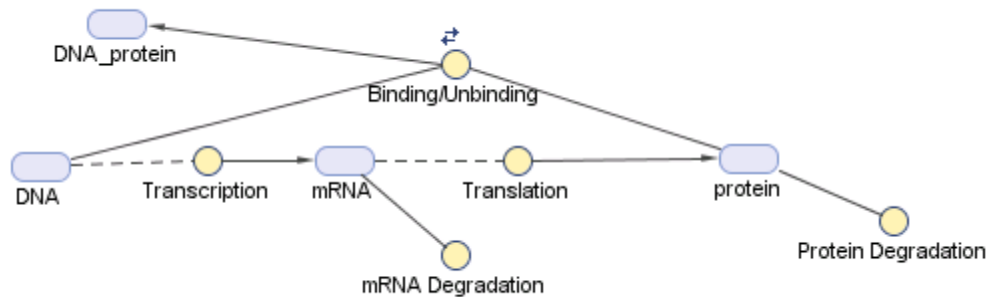
	Name	Scope	Value	ValueUnits
1	k_1	DNA \rightarrow DNA + mRNA	0.2	1/second

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



- 18** 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 (modifiers). 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 2-26 for more information.

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,

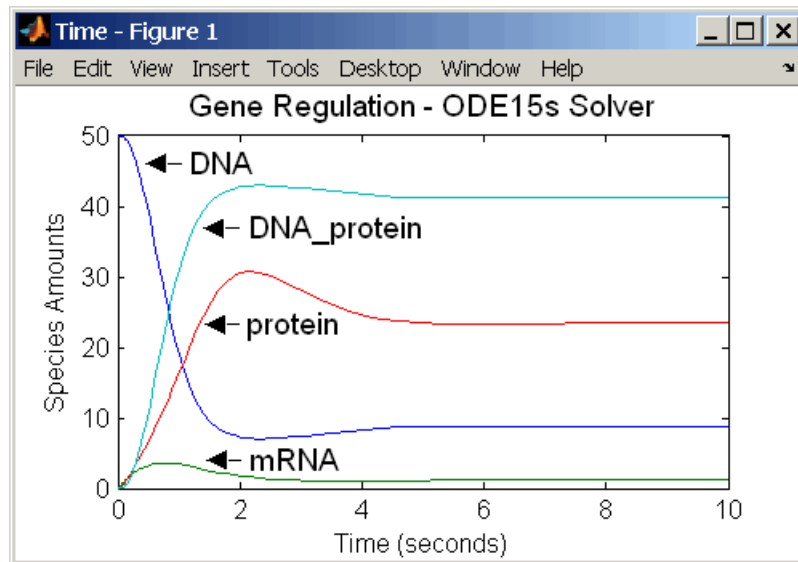
- 1 In the **Project Explorer**, right-click the model.
- 2 Select **Activate (model name)**.

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

To simulate 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 In the **Diagram** pane, click  (**Run**).

The SimBiology desktop adds a **Simulation** task to the project. The simulation runs to completion and then plots the results in a figure window.



- In the **Project Explorer**, double-click **Simulation** to open the **Simulation** pane.
- In the **Project Explorer**, under **Simulation**, double-click **Data** to open the **Data** pane containing the data associated with the simulation.

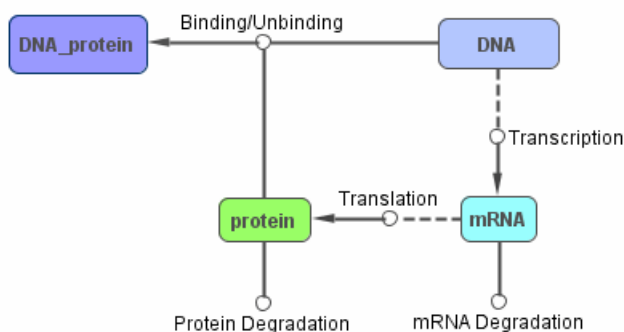
Setting Block Appearance and Diagram Layout

This section shows you how to change the appearance and layout of a SimBiology diagram.

- “Changing Block Appearance” on page 2-26
- “Applying Block Appearance Styles” on page 2-28
- “Changing the Diagram Layout Manually” on page 2-29
- “Changing the Diagram Layout Automatically” on page 2-32
- “How SimBiology Does Diagram Layout” on page 2-35

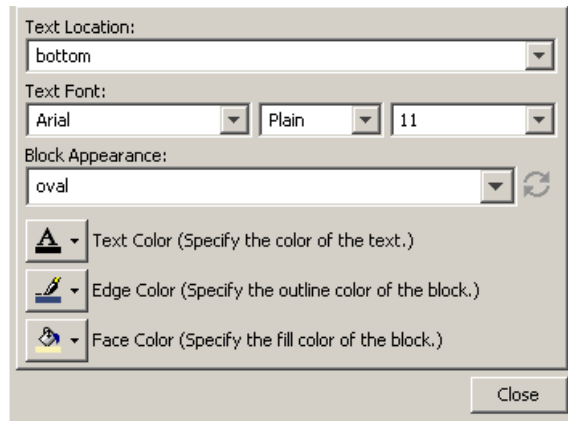
Changing Block Appearance

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.



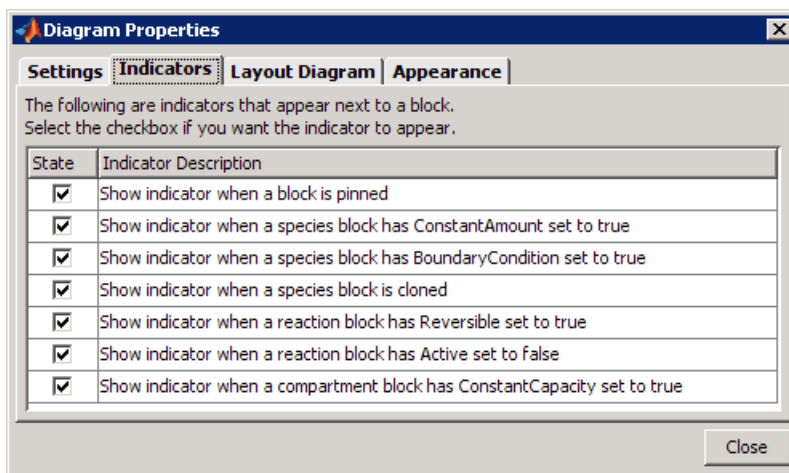
- 3 From the **Text Location** list, select where the text should be positioned in the diagram. The default is bottom. For example:
 - For the DNA Species block, select center to position the text within the center of the Species block.
 - For the Binding/Unbinding Reaction block, select top.
- 4 From the **Text Font**, **Block Appearance**, **Text Color**, **Edge Color**, and **Face Color** lists, select the text font, block appearance, text color, edge color, and block fill color.

Tip Select **Face Color > No Face Color** to obtain the result shown for reaction blocks in the diagram figure above.

Tip Select **Block Appearance > Image** to insert an image for the block. SimBiology opens a dialog box that lets you browse to the image location. If you change the image, click  **Refresh** to reload the updated image.

- 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 previous example diagram 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 in the diagram.

- 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 the Diagram Layout Manually” on page 2-29, and for information on automatic layout in SimBiology, see “Changing the Diagram Layout Automatically” on page 2-32.

Applying Block Appearance Styles

To replicate the appearance style of a block to other blocks by copying and applying the style on selected blocks:

- 1 Select a Species block, for example, the DNA Species block.
- 2 Right-click the block and select **Copy Block Style**.
- 3 Select another block, for example, the DNA protein Species block.

You can also select multiple blocks to apply the style to all selected blocks.

- 4 Right-click the block and select **Apply Block Style**. SimBiology changes the appearance of the block to match the selected style.

Changing the Diagram Layout Manually

You can manually lay out the diagram using the pin, split, clone, and hide features in the diagram. Use the **Diagram Table View** pane or a block's right-click context menu to change the properties discussed below. To access **Diagram Table View**, from the **Diagram** menu, select **Tools > Diagram Table View**.

Tip While you are working on the 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 lay out the model diagram:

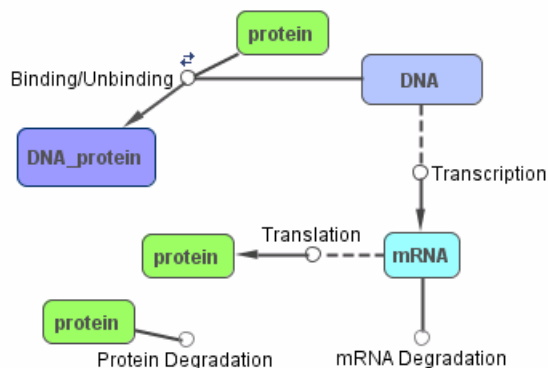
- 1 Right-click a Species block, and select **Split Block**. For example, split the protein Species block.

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 box for any of these split blocks, the change will propagate 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.

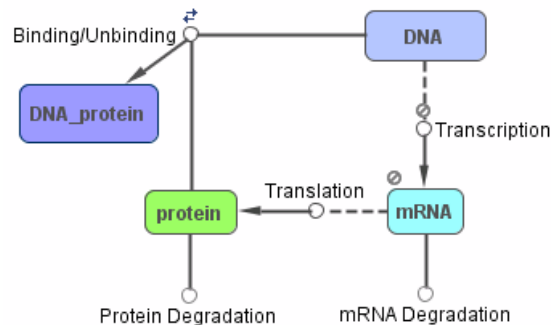


3 From the Diagram menu, select **Tools > Diagram Table View** to open the **Diagram Table View** pane.

	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**+left-click to select the row for the mRNA Species block. **Ctrl**+click lets you select discontinuous rows.
- 6 Right-click and select **Pin Block**. SimBiology pins the blocks. You cannot move pinned blocks manually or with automatic layout. Note the indicator for pinned blocks in the following diagram.



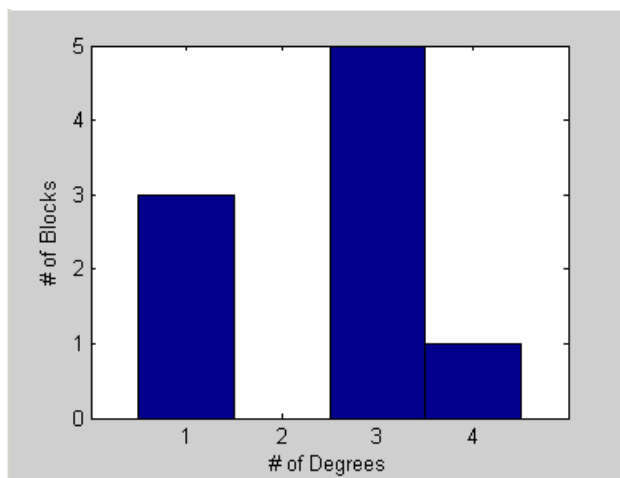
- 7 Select the row for the Protein Degradation reaction and press **Shift**+left-click to select the row for the mRNA Degradation reaction.
- 8 Right-click and select **Hide Selected Blocks**. SimBiology hides the two reactions in the diagram. The reaction rows are visible in the **Diagram Table View**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

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

- 9 Select a block, right-click and select **Rotate Clockwise** to rotate the selected block.
- 10 Select **Diagram > Show All Hidden Blocks and Lines** to see all blocks.

- 11 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.

Changing the Diagram Layout Automatically

From the **Diagram** menu select **Layout** to use the automatic layout feature. Select blocks and use the automatic layout feature to lay out only the selected blocks in the diagram.

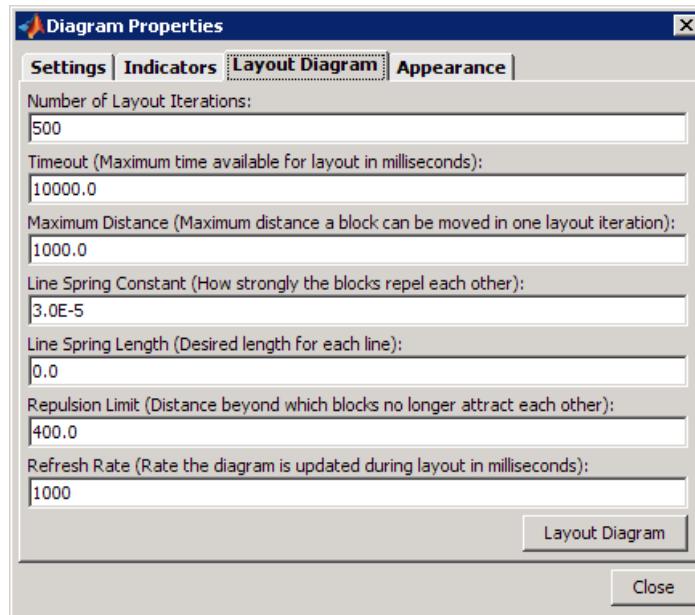
Automatic layout works only on blocks that are not pinned. You can right-click and select **Unpin Block** if you want to lay out all blocks. Leave blocks pinned if you do not want the position of the block to change during layout.

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 change the automatic layout settings in SimBiology.

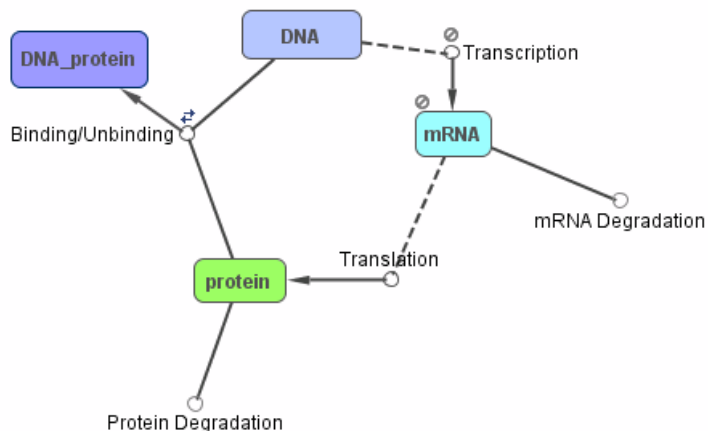
To change the settings for automatic layout:

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

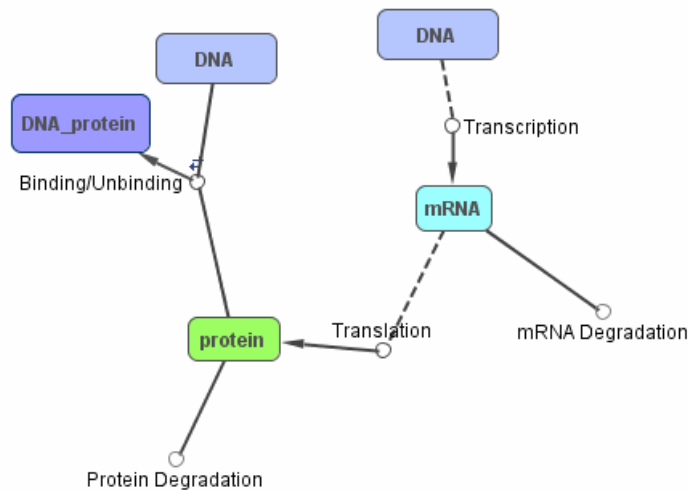


- 3 In the **Number of Layout Iterations** box, enter the number of iterations to include in the layout. The default is 500. More iterations give you a better layout, but also take more time. For example, enter 700.
- 4 In the **Maximum Distance** box, specify the distance that a block moves in one iteration. 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 0 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
- 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.

Printing and Exporting the Diagram

You can annotate and print your pathway’s block diagram in SimBiology, or you can export the diagram to .svg, .jpeg, or .pdf file formats. For example, you can annotate the diagram with the name of the author, the date, notes, and name of the organization. You can choose to place the content as a header or footer on the diagram page.

Printing a Diagram

Right-click in the SimBiology diagram and select **Print** to open the Print dialog box and print the diagram. By default there are no annotations. However, if you previously annotated the diagram using **Diagram Print Setup**, the printed document contains the annotations. You can access **Diagram Print Setup** from the diagram’s right-click (context) menu.

To annotate and print a diagram:

- 1 In the SimBiology **Diagram** pane, right-click the background of the diagram and select **Diagram Print Setup**.

The Diagram Print Setup dialog box opens.

- 2** In the **Diagram Options** tab do the following:
 - a** In the **Scale** box, select the size to adjust the diagram to (100% is default size). Alternatively, click the **Fit diagram to** button and specify the desired width and height.
 - b** Clear the **Print frame around diagram** check box if you do not want a frame in the printed document. SimBiology adds a frame by default.
 - 3** Click the **Page Annotation** tab.
 - 4** Click **Edit**. The Customize Page Annotation dialog box opens. This dialog box lets you enter and store customized information for print annotation.
 - a** In the relevant fields, enter the name of your organization, and your name.
 - b** From the **Date Format** and **Page Format** lists, select the date and page formats respectively.
 - c** Click **OK**
 - 5** Place the cursor in a **Header** or **Footer** box and click **Author**, **Company**, **Model**, or **Date**.
- SimBiology inserts the information entered in the Customize Page Annotation dialog box. Alternatively, you can type information into each field.
- 6** In the **Diagram Notes** box, enter notes to be included.
 - 7** From the notes location list select whether to include notes in the printout, and the location. You can also choose to include the notes only on the first page.
 - 8** Click **OK** to leave Diagram Print Setup.

Alternatively, click **Print** to open the Print dialog box and print the diagram.

Exporting a Diagram Image

You can export the image of a pathway diagram in a portable file format for use by other applications such as PowerPoint®. SimBiology exports the graphical representation of the model.

- 1 In the SimBiology **Diagram** pane, right-click the background of the diagram and select **Export Current Diagram As**.

The Export Current Diagram As dialog box opens.

- 2 From the **File type** list, specify the type of file to export.
- 3 Click ... (Browse), to choose the file location.
- 4 In the **File name** box, specify the file name.
- 5 Keep the default selection for the **Preserve aspect ratio** check box to preserve the width and height ratio of the diagram.
- 6 (Optional) Specify the desired width and height of the image in inches, in the **Width** and **Height** boxes respectively.
- 7 Click **OK** to export the file.

Modeling Using the Command-Line Interface

In this section...

“About This Example” on page 2-39

“Creating a SimBiology Model” on page 2-39

“Adding the Reaction for Transcription” on page 2-41

“Adding the Reaction for Translation” on page 2-42

“Adding the Reactions for Gene Regulation” on page 2-42

“Adding the Reactions for mRNA and Protein Degradation” on page 2-43

“Simulating the Model with an ODE Solver” on page 2-44

“Simulating the Model with a Stochastic Solver” on page 2-45

About This Example

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 2-8 and “Modeling Using the SimBiology Diagram” on page 2-17). 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

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.

- 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.

```
SimBiology Model - cell
```

```
Model Components:
  Compartments:    0
  Events:          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: ''
  Events: [0x1 double]
  Name: 'cell'
  Notes: ''
  Parameters: [0x1 double]
  Parent: [1x1 SimBiology.Root]
  Species: [0x1 handle]
  Compartments: [0x1 double]
  Reactions: [0x1 double]
  Rules: [0x1 double]
  Tag: ''
  Type: 'sbiomodel'
  UserData: []
```

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

3 Add a compartment named contents to the model.

```
compObj = addcompartment(Mobj, 'contents');
```

All models must contain a compartment. You can either add a compartment and specify its name (as shown here), or you can let the process of adding a

reaction automatically create a default compartment (*unnamed*) and add the reaction species to that compartment. You can rename the compartment later.

If there are multiple compartments in the model, you must specify the reactants and products using qualified names (*compartmentName.speciesName*). For example `nucleus.DNA` denotes the species DNA, in the compartment `nucleus`.

This example contains only one compartment.

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 `DNA -> DNA + mRNA` with reaction rate equation `v=k*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'`.

Since this example has only one compartment, you need not specify the compartment to which the species belong. If there are multiple compartments, below is an example of the reaction syntax:

```
Robj1 = addreaction(Mobj, 'nucleus.DNA -> nucleus.DNA + cytoplasm.mRNA');
```

Where `nucleus` and `cytoplasm` are the names of the compartments.

- 2 Define the reaction rate constant `k1=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 DNA = 50 and mRNA = 0. Select the species using the function `sbioselect`, which returns an object.

```
s = sbioselect('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.

```
SimBiology Species Array
```

Index:	Compartment:	Name:	InitialAmount:	InitialAmountU
1	contents	DNA	50	
2	contents	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 $k_2 = 20$ 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:            all

  CompileOptions:
    UnitConversion:         false
    DimensionalAnalysis:    true

  SensitivityAnalysisOptions:
    InputFactors:           0
    Outputs:                0
```

- 2 Set the final simulation time.

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

- 3 Run the simulation.

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

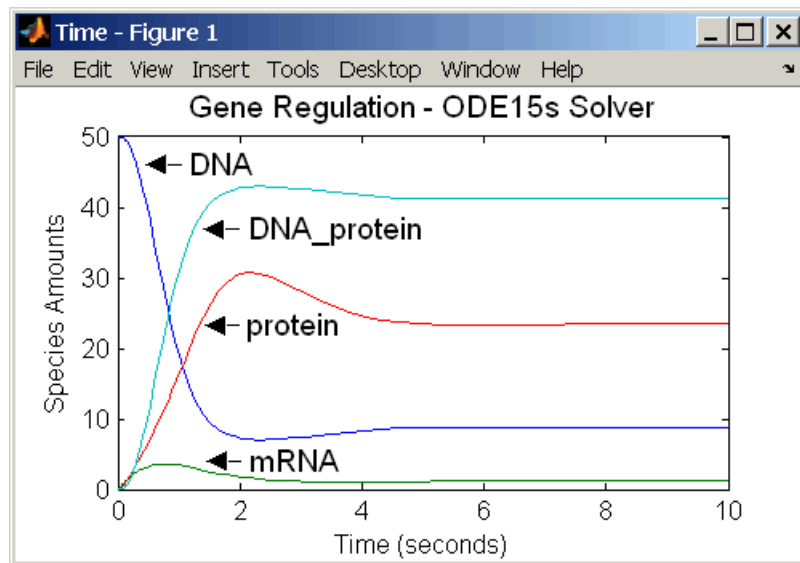
- 4 Plot the results.

```

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 plots 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:           8.000000

  SolverOptions:
    LogDecimation:    1

  RuntimeOptions:
    StatesToLog:      all

  CompileOptions:
    UnitConversion:   false
    DimensionalAnalysis: true

  SensitivityAnalysisOptions:
    InputFactors:     0
    Outputs:          0
```

3 Change the values for the StopTime and LogDecimation properties.

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

4 Run the simulation.

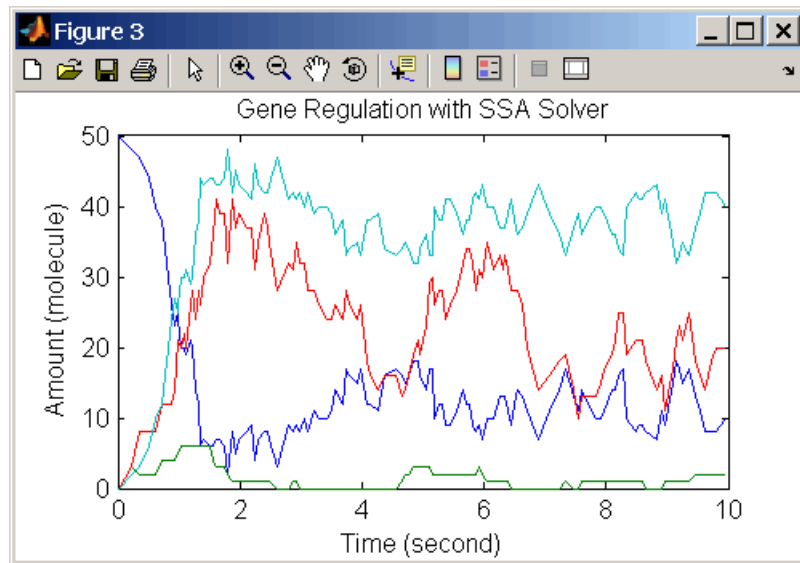
```
[t_ssa, x_ssa] = sbiosimulate(Mobj, cs);
```

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



- 6 Another method to visualize stochastic simulations is through ensemble runs.

```
simDataObj = sbioensamplerun(Mobj, 10, cs);
sbiosubplot(simDataObj);
```


G

gene regulation
 GUI desktop 2-8
 GUI Diagram 2-17
 model diagram 2-2
 reaction list 2-2
 reactions 2-3
 tutorial model 2-2

K

Kinetic Law
 object definition 1-28

M

model
 adding reactions 2-10 2-19 2-41
 creating 2-8 2-17
 creating command-line 2-39
 gene regulation 2-42
 project file 2-9 2-19
 protein degradation 2-43
 simulating 2-15 2-24
 translation 2-42

P

Parameters
 definition 1-29
 object definition 1-29

R

Rate constants. *See* Parameters
Reaction
 object definition 1-28
Reaction rate
 definition 1-28

reactions
 degradation 2-43
 gene regulation 2-42
 transcription 2-41
 translation 2-42

Rules
 object definition 1-30

S

SimBiology
 computation with MATLAB 1-2
 defined 1-2
 expected user 1-3
 features and functions 1-14
 model component definitions 1-25
 optional software 1-4
 related software 1-5
 required software 1-4
 SBML support in 1-23
 visualizing data 1-2
simulation
 ODE solver 2-44
 stochastic solver 2-45
software
 optional 1-4
 required 1-4
Species
 object definition 1-27

T

tutorial
 basic 2-1
 command-line 2-39
tutorial models
 gene regulation 2-2