SimBiology

For Use with MATLAB®

- Computation
- Visualization
- Programming

Getting Started

Version 1

The MathWorks

How to Contact The MathWorks:

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

@	`support@mathworks.com`
	`suggest@mathworks.com`
	`bugs@mathworks.com`
	`doc@mathworks.com`
	`service@mathworks.com`
	`info@mathworks.com`
		Technical Support
		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	Mail

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

Getting Started with SimBiology

Revision History

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

Contents

1 Introduction

2 Tutorial Models

1

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?

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

- **Model** — Design and build models by entering reactions, species, parameters, kinetic laws, rules, and submodels with a graphical user interface or using the MATLAB® Command Window. Import SBML models created with SimBiology or other modeling software that is compliant with the SBML standard. Visualize the reaction pathways in a block-diagram explorer.

  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 1-6.
- **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 ODE solvers. The stochastic solvers include a stochastic simulation algorithm with implicit and explicit tau variations. For more information, see "Simulation" on page 1-7.
- **Analyze** — Save data from a simulation and export that data to the MATLAB workspace. Compare simulation and experimental data. For more information, see "Further Analysis with MATLAB" on page 1-8.

Required Software

SimBiology requires the following products from the MathWorks to be installed on your computer:

MATLAB	Provides a command-line interface and integrated software environment. SimBiology Version 1.0 (Release 14SP3+) requires MATLAB Version 7.1 on the Release 14SP3 CD.

Features and Functions

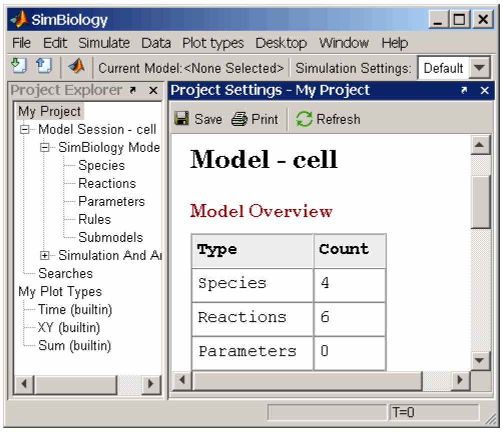
SimBiology includes many features to help you model mathematically biological processes, simulate those processes, and analyze the results.

Graphical User Interface (p. 1-4)	Enter reactions and kinetic data using a familiar spreadsheet format.
Command-Line Interface (p. 1-5)	Create and simulate model objects interactively or by using M-code scripts.
Data Formats and Projects (p. 1-6)	Group related models and simulation data together into project files.
Modeling (p. 1-6)	Model metabolic, genetic, and signaling pathways.
Simulation (p. 1-7)	Select deterministic or stochastic solvers, simulate models, save data, and plot results.
Further Analysis with MATLAB (p. 1-8)	Use MATLAB functions to analyze model properties, compare simulation data with experimental data, and plot results.

Graphical User Interface

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

The window for this interface is divided into the Project Explorer and model object panes. Optionally, you can view Kinetic Law Manager, Unit Manager, Table Manager, Block Diagram Explorer, and embedded help panes.



Model - cell

Model Overview

Type	Count
Species	4
Reactions	6
Parameters	0

- **Project Explorer** — View projects and the organization of models, and navigate to components and subcomponents using a hierarchical tree view. The project branch is divided into separate model sessions. A model session includes model and simulation branches. The model branch is further divided into components while the simulation branch is divided into simulation settings and simulation data.

- **Model Objects** — Enter model data using a familiar spreadsheet format, forms, comma-separated files, or previously saved SBML models. Model biological processes as reactions with reactants, products, and reaction rates. Reactions include processes that transform species, and processes such as transport, diffusion, and binding.
- **Embedded Help** — The embedded help pane is context sensitive to the tasks you are performing while building 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.

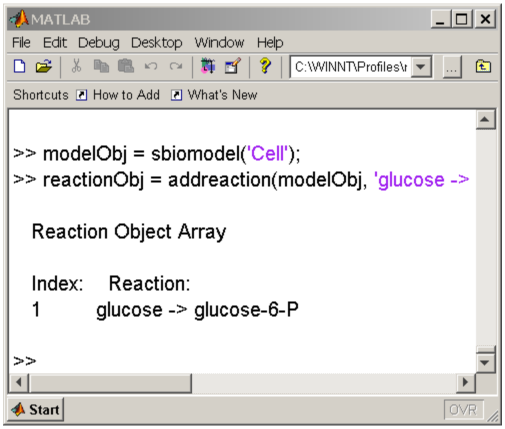
  The Help browser supplements the embedded help information with detailed reference material for command-line functions; object methods and properties; tutorials to help you get started; and more details about modeling, simulation, and analysis.

- **Kinetic Law Manager** — View built-in kinetic laws and define your own kinetic laws.
- **Unit Manager** — View built-in units or define your own units composed from basic physical quantities.
- **Table Manager** — Customize the desktop spreadsheets using the Table Manager to display or hide the parameters you are interested in viewing or editing.
- **Block Diagram Explorer** — View the components of a model and visualize reaction pathways with a block diagram created from the reactions and species you entered.
- **Search** — Search for object properties with specific values.
- **M-code capture** — Convert interactive GUI actions with the SimBiology desktop to M-code scripts, and load them into MATLAB.

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.



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

  Reaction Object Array

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

>>
```

- **Functions** — Use functions to create projects, manage the unit and kinetic law libraries, and simulate models.
- **Methods** — Methods create objects and the relationships between objects.

- **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

Projects group related models and simulation data into project files.

- **Projects** — Organize projects with related models, simulation data, and analysis results. Explore and search for model components.

  Projects store the kinetic law and unit libraries, model component information, configuration set (simulation settings), and time-series data (simulation results).

- **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 in an SBML file created with SimBiology or other modeling software.

  When you save a project, the model definition is saved with the project as SimBiology objects. 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, and creates a model session on the SimBiology desktop.

- **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.

Modeling

SimBiology is an environment for designing and building biochemical models.

- **Model objects** — Biological models are represented as a related group of objects. After creating a model object, you can add reaction objects to the model, and then add kinetic law objects and parameter objects to the reaction object. Other objects define simulation settings, units, and rules.

- **Model properties** — The properties define the characteristics of an object. For example, a reaction object includes properties for reactions, reaction rates, and links to associated species, parameters, and kinetic laws. Use the MATLAB `get` and `set` commands to list object properties and change their values.
- **Submodel objects** — SimBiology extends the current version of SBML by hierarchically organizing a model with submodels. A submodel can be a group of reactions.

  By partitioning your model into functional units, you can easily understand the model and the interactions between the submodels. You also can reuse the functional units in other modeling projects.
- **Notes** — SimBiology extends the usefulness of the SBML notes field by allowing HTML formatted text. This allows you to link to journal articles in PDF and other formats, Web pages, databases, graphic files, experimental data, and verbose project notes that do not fit into a text box.

Simulation

SimBiology includes both deterministic and stochastic solvers.

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

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

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

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

- **Deterministic simulation** — You can choose from the list of stiff and nonstiff deterministic simulation solvers that are included with MATLAB. These solvers are reliable and robust.
- **Dimensional analysis and unit conversion** — Dimensional analysis compares the physical quantities for parameters and species in mathematical expressions (reaction rates, algebraic rules, assignment rules, 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.
- **Simulations settings** — Configuration set objects define the properties needed for simulation. These properties include the simulation time, absolute and relative tolerances, and controls for dimensional analysis and unit conversion.

Further Analysis with MATLAB

SimBiology is closely integrated with MATLAB.

- **Getting simulation data into MATLAB** — During a simulation, SimBiology saves (logs) the values for parameters and the amounts of species to the MATLAB workspace.
- **Plotting simulation data and analyzing results** — Using the functions in MATLAB, you can select and scale species for plotting.

SimBiology Objects and SBML

One way to define a biological system is to divide the system into multiple components or objects. After making the divisions, you can define the characteristics of the components and the relationships between the components.

What Is SBML?

Systems Biology Markup Language (SBML) is an information standard for sharing, evaluating, and developing systems biology models cooperatively . Models include signaling, metabolic, and gene regulation pathways with other biological processes.

Its purpose is to provided 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/
```

SimBiology supports a subset of the SBML specification and does not currently support the following SBML features:

- Compartments — SimBiology models are hierarchically structured with one model having one compartment, while an SMBL model can have multiple compartments in one model.
- Volume — The volume property for compartments is not supported.
- Events — Events are not supported.

On the other hand, SimBiology models have features that the current SBML specification does not support.

- Projects
- Hierarchical models
- Kinetic law information such as the kinetic law name.

What Are SimBiology Objects?

SimBiology translates SBML model components into a series of objects. The objects follow the current SBML standard (Level 2 Version 1) closely. However, some objects have additional properties not in the current standard. These additional properties are needed for displaying models, saving simulation properties with simulation data, and analysis.

This is the hierarchical organization of SimBiology objects:

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

Parameters for kinetic laws are listed in the model parameter list, but they can be scoped to a kinetic law or the model.

2

Tutorial Models

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

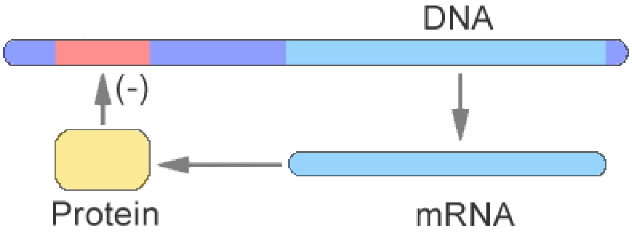
Gene Regulation Model

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

Model Diagram (p. 2-2)	Visualize a systems biology model for gene regulation with different levels of detail.
Model Reactions (p. 2-3)	Describe a systems biology model for gene regulation with a series of reaction and rate equations.

Model Diagram

You can visualize a systems biology model with variouslevels of detail. One view sketches only the major species and processes. The gene-regulation example below 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.

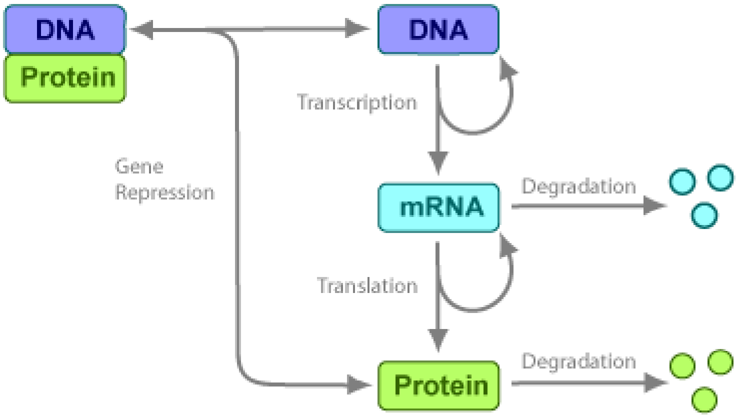


Another way of looking at a systems biology model is to list the reactions in a model with the processes they represent.

```
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.

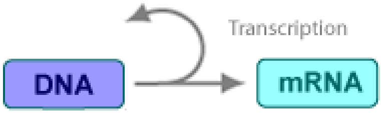


Model Reactions

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.

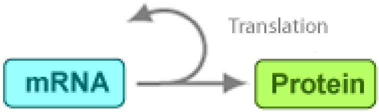


This model simplifies the transcription and the synthesis of mRNA by using a single reaction.

```
     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.

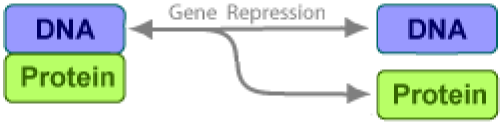


Synthesis of the proteins is modeled as a single reaction.

```
     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 K3. 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 evaluation 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
```

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

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.

Protein
Degradation

```
     Reaction: protein -> null
Reaction rate: v = k5*protein molecule/second
      Species: protein = 0.0 molecule
   Paremeters:      k5 = 1.0 1/second
```

Gene Regulation Using Graphical Interface

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

Alternatively, you can use M-code and the MATLAB command-line interface (see “Gene Regulation Using Command-Line Interface” on page 2-12). A comparison of the two methods will help you understand the model objects that MATLAB creates behind the `sbiodesktop`, and how to access and edit a model through the MATLAB Command Window.

Creating a Systems Biology Model (p. 2-7)	Use the SimBiology desktop to create a structure for entering reaction and kinetic equations.
Adding Reactions to a Model (p. 2-8)	Add reactions with species, reaction rate equations, and parameter values to your model .
Simulating a Model (p. 2-11)	Simulate your model and plot the change in amount of species over time.

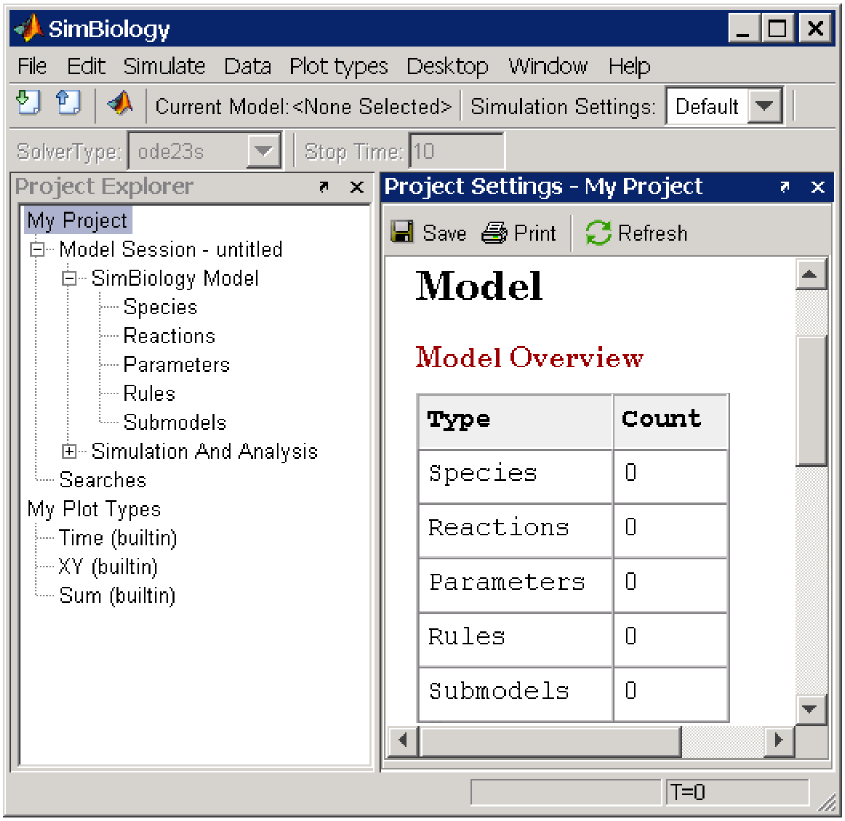
Creating a Systems Biology Model

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

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

   ```
   sbiodesktop
   ```

   The SimBiology window opens with a new but empty model.



Model

Model Overview

Type	Count
Species	0
Reactions	0
Parameters	0
Rules	0
Submodels	0

2 In the left pane, click **SimBiology Model** and enter a name for your model. For example, in the **Model Name** box, enter `cell`.

Adding Reactions to a Model

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

1 In the left pane, click **Reactions**.

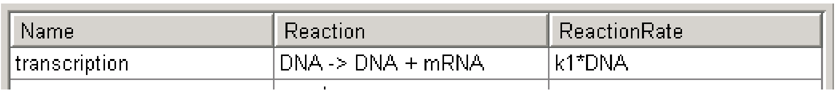
2 In the **Enter Reaction** box, enter the reactants and products, separated by a hyphen and right angle bracket. For example, enter the following simple reaction for transcription.

```
DNA -> DNA + mRNA
```

3. Click **Add**.

4. In the **ReactionRate** box, enter a reaction rate equation. For example, enter `k1*DNA`. In the **Name** box, enter a name for the reaction.

   Your reaction table should look similar to the one below.



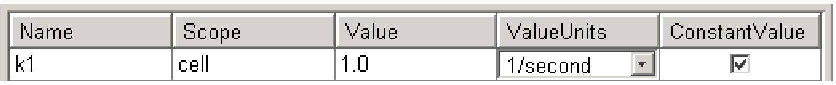
Name	Reaction	ReactionRate
transcription	DNA -> DNA + mRNA	k1*DNA

5. In the left pane, click **Parameters**. In the **Enter Name** box, enter the name for a reaction rate constant. In this case, enter `k1`, and then click **Add** or press **Enter** key.

   The default value for a parameter is `1.0`.

6. 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 look similar to the one below.



Name	Scope	Value	ValueUnits	ConstantValue
k1	cell	1.0	1/second ▾	☑

7. In the left pane, click **Species**. The species you entered in the reaction are automatically added to the species table.

   The default value for a species is `0.0`.

8. 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.

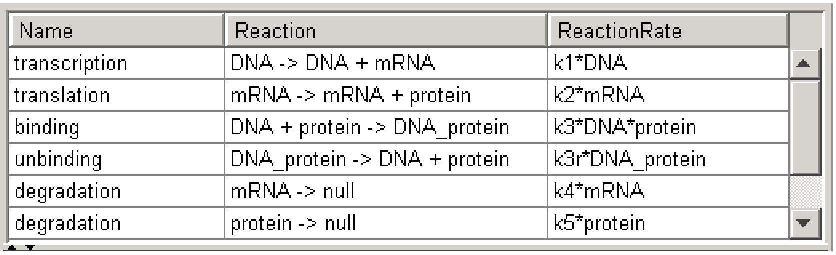
9. Enter the remaining reactions. For a list of reactions and parameters, see “Model Reactions” on page 2-3.

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

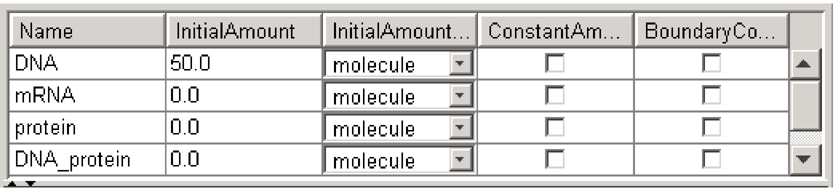
```
1/(molecule*second)
```

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

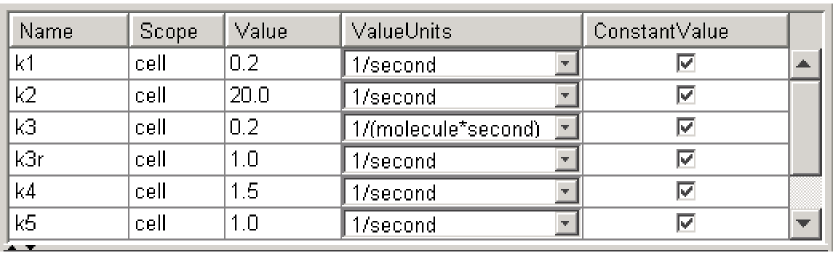
Your reaction, species, and parameter panes should look similar to the ones below.



Name	Reaction	ReactionRate
transcription	DNA -> DNA + mRNA	k1*DNA
translation	mRNA -> mRNA + protein	k2*mRNA
binding	DNA + protein -> DNA_protein	k3*DNA*protein
unbinding	DNA_protein -> DNA + protein	k3r*DNA_protein
degradation	mRNA -> null	k4*mRNA
degradation	protein -> null	k5*protein



Name	InitialAmount	InitialAmount...	ConstantAm...	BoundaryCo...
DNA	50.0	molecule	□	□
mRNA	0.0	molecule	□	□
protein	0.0	molecule	□	□
DNA_protein	0.0	molecule	□	□



Name	Scope	Value	ValueUnits	ConstantValue
k1	cell	0.2	1/second	☑
k2	cell	20.0	1/second	☑
k3	cell	0.2	1/(molecule*second)	☑
k3r	cell	1.0	1/second	☑
k4	cell	1.5	1/second	☑
k5	cell	1.0	1/second	☑

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

Simulating a Model

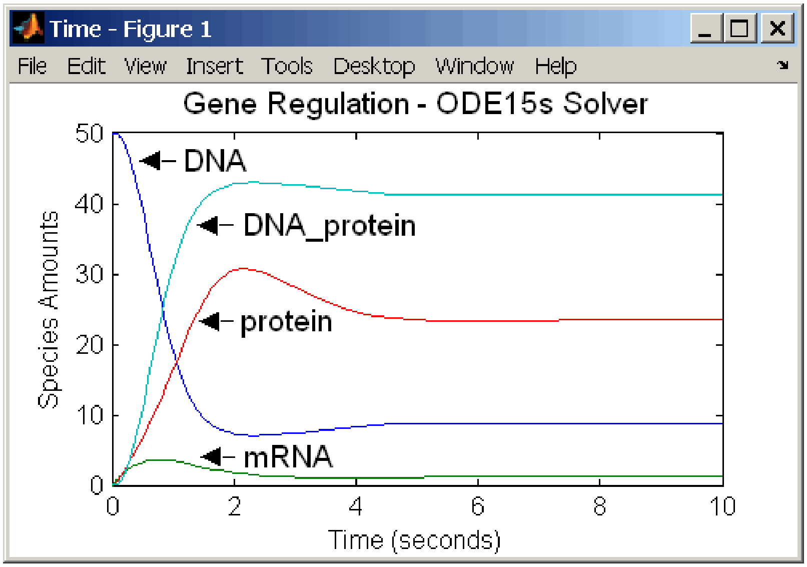
After you enter the reactions for your model, you can simulate its dynamic behavior and plot the results.

1 From the **Simulation** toolbar, and from the **Solver Type** list, select a stochastic or ordinary differential equation (ODE) solver. For example, select `ode15s`.

2 In the **Stop time** box, enter the length of time for the simulation. For example, enter `10` seconds.

3 From the **Simulate** menu, click **Start**, or from the toolbar, click the simulate button ▶.

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



Gene Regulation Using Command-Line Interface

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

Alternatively, you could use a graphical user interface (see "Gene Regulation Using Graphical Interface" on page 2-7). A comparison of the two methods will help you understand the model objects that MATLAB creates behind the `sbiodesktop`, and how to access and edit a model through the MATLAB Command Window.

Creating a Systems Biology Model (p. 2-12)	Use the MATLAB Command Window to create a structure for entering reaction and kinetic equations.
Adding the Reaction for Transcription (p. 2-14)	A DNA molecule produces multiple copies of mRNA.
Adding the Reaction for Translation (p. 2-15)	Multiple copies of a protein are produced from each mRNA molecule.
Adding the Reactions for Gene Regulation (p. 2-15)	The protein binds with DNA and stops transcription.
Adding the Reactions for mRNA and Protein Degradation (p. 2-16)	Degradation reactions balance the synthesis reactions and maintain a steady state.
Simulating the Model with an ODE Solver (p. 2-16)	Select a deterministic numerical solver.
Simulating Model with Stochastic solver (p. 2-18)	Select a stochastic solver that considers the probability of the occurrence of a reaction.

Creating a Systems Biology Model

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

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

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

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

```
Systems Biology Model - cell

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

2 Display all the model object properties by typing

```
 get(Mobj)
```

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

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

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

Adding the Reaction for Transcription

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

1 Enter the reaction `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'`.

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`.

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

The value for `mRNA` defaults to `0`.

4 Check the initial amounts for the species.

```
Mobj.Species
```

MATLAB displays a summary list.

```
Species Object Array

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

Adding the Reaction for Translation

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

1 Enter the reaction `mRNA -> mRNA + protein` with reaction rate equation `v = k*mRNA`.

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

2 Define the reaction rate constant `K2 = 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 = K3*DNA*protein` and a reverse rate `v = K3r*DNA: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

    RuntimeOptions:
      StatesToLog:          all

    CompileOptions:
      UnitConversion:       true
      DimensionalAnalysis:  true
```

2 Set the final simulation time.

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

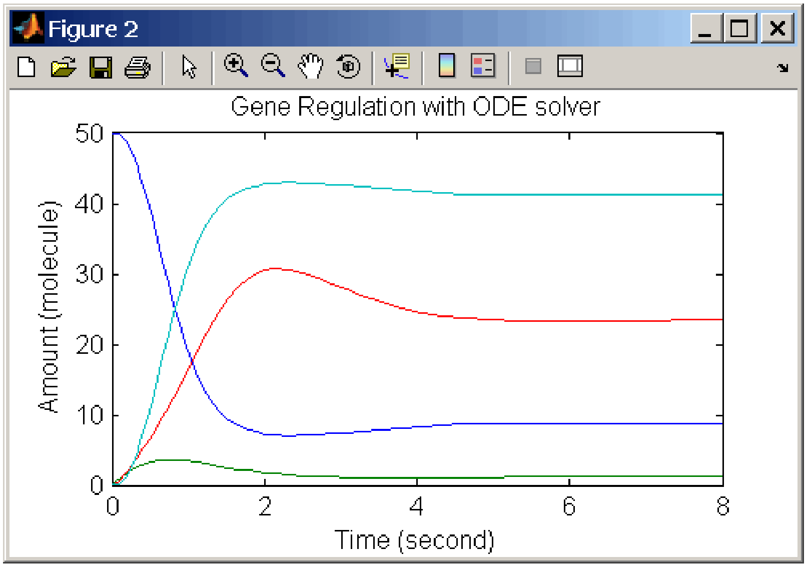
3 Run the simulation.

```
[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 with ODE Solver');
xlabel('Time (second)');
ylabel('Amount (molecule)');
axis([0, 8, 0, 50]);
```

MATLAB draws a figure with the simulation results.



Simulating Model with Stochastic solver

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

1 Get the active configuration set for a model.

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

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

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

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

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

```
    SolverOptions:
      LogDecimation:        10

    RuntimeOptions:
      StatesToLog:          all

    CompileOptions:
      UnitConversion:       true
      DimensionalAnalysis:  true
```

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

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

4 Run the simulation.

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

5 Plot the results.

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

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

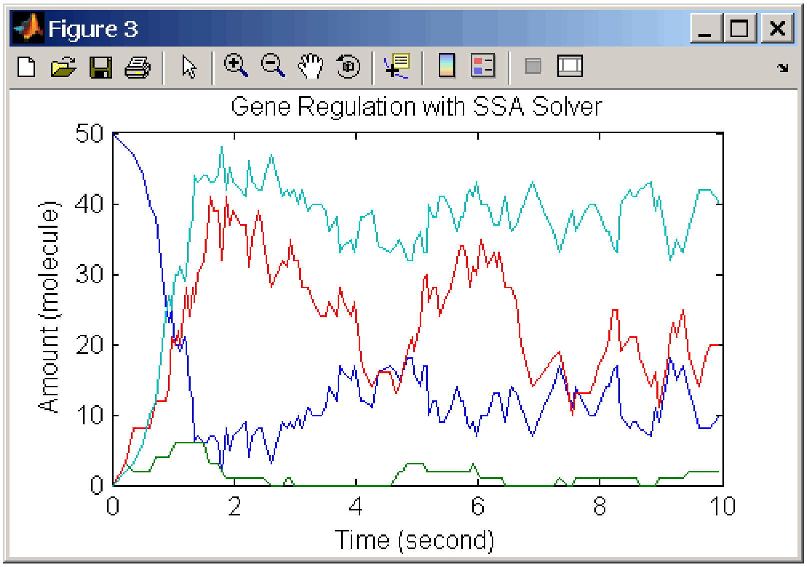


Figure 3
Gene Regulation with SSA Solver
Amount (molecule)
Time (second)
0
10
20
30
40
50
0
2
4
6
8
10

Index