

iBioSim Version 2.4

User's Manual

Chris J. Myers, Nathan Barker, Kevin Jones, Hiroyuki Kuwahara,
Curtis Madsen, Nam Nguyen, Tyler Patterson, Nicholas Roehner, Jason Stevens

January 5, 2013

Contents

| | | |
|-----------|---|-----------|
| 1 | Introduction | 3 |
| 2 | Project Management | 4 |
| 2.1 | Creating and Opening Projects | 5 |
| 2.2 | Creating Models and Graphs | 6 |
| 2.3 | Importing Models | 9 |
| 2.4 | Exporting Models and Graphs | 10 |
| 2.5 | Editing Project Objects | 11 |
| 2.6 | Creating Tool Views | 11 |
| 3 | Model Editor | 13 |
| 3.1 | Compartments | 16 |
| 3.2 | Species | 17 |
| 3.3 | Reactions | 19 |
| 3.4 | Components | 23 |
| 3.5 | Promoters | 26 |
| 3.6 | Real Variables | 28 |
| 3.7 | Boolean Variables | 29 |
| 3.8 | Petri Net Places | 30 |
| 3.9 | Petri Net Transitions | 31 |
| 3.10 | Rules | 33 |
| 3.11 | Constraints | 33 |
| 3.12 | Events | 34 |
| 3.13 | Influences | 35 |
| 3.14 | Model Editor | 38 |
| 3.15 | Constants | 39 |
| 3.16 | Functions | 40 |
| 3.17 | Units | 41 |
| 4 | SBOL Editors | 43 |
| 5 | Analysis Tool | 46 |
| 5.1 | Simulation Options | 46 |
| 5.2 | Abstraction Options | 50 |
| 5.3 | Schematic | 52 |
| 5.4 | Parameters | 54 |
| 5.5 | SBML Elements | 55 |
| 6 | Learn Tool | 56 |
| 7 | TSD Graph Editor | 59 |
| 8 | Histogram Graph Editor | 62 |
| 9 | Preferences | 65 |
| 10 | Mathematical Formulas | 67 |

| | |
|---|-----------|
| 11 Time Series Data Format | 68 |
| 12 Tutorial | 68 |
| 13 Reporting Bugs and Feature Requests | 69 |

1 Introduction

iBioSim has been developed for the modeling, analysis, and design of genetic circuits. While **iBioSim** primarily targets models of genetic circuits, models representing metabolic networks, cell-signaling pathways, and other biological and chemical systems can also be analyzed. Recently, modeling and visualization support has been added for static multi-cellular and spatial models as well. It is capable of importing and exporting models specified using the *Systems Biology Markup Language* (SBML). It can import all levels and versions while being able to export Level 3 Version 1. It has support the hierarchical modeling package, as well as, the layout package. It supports all SBML modeling constructs except the *delay* function and some types of *fast* reactions. It was the first tool to produce correct results for all examples in the SBML benchmark suite. It has also been tested successfully on the stochastic benchmark suite and the curated models in the *BioModels database*. Finally, it is one of the first tools to also support the *Synthetic Biology Open Language* (SBOL), an emerging standard for information exchange in synthetic biology. **iBioSim** includes the following components:

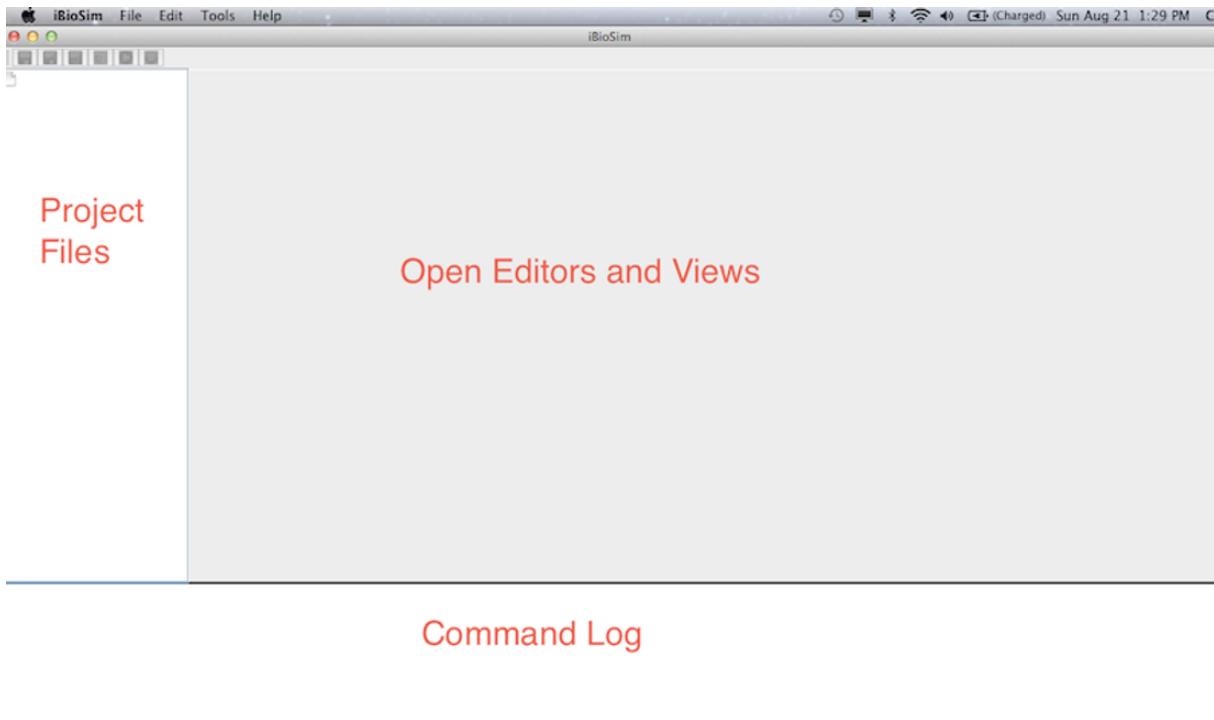
- Model Editor - a tool to create a model of a genetic circuit or other biological system.
- SBOL Browser - a tool to view SBOL files and associate DNA components to model elements.
- Analysis Tool - an abstraction-based ODE, Monte Carlo, and Markov analysis tool.
- Learn Tool - a tool to learn a model from *time series data* (TSD).
- TSD Graph Editor- a tool to visualize TSD files.
- Histogram Graph Editor - a tool to visualize probability data.

Credits

The iBioSim tool is being developed at the University of Utah by Chris Myers, Nathan Barker, Kevin Jones, Hiroyuki Kuwahara, Curtis Madsen, Nam Nguyen, Tyler Patterson, Nicholas Roehner, and Jason Stevens. Nathan Barker is now with Southern Utah University, Kevin Jones is now with Raytheon, Hiroyuki Kuwahara is now with Carnegie Mellon University, Nam Nguyen is now with the University of Texas in Austin, and Jason Stevens is now with the University of Washington.

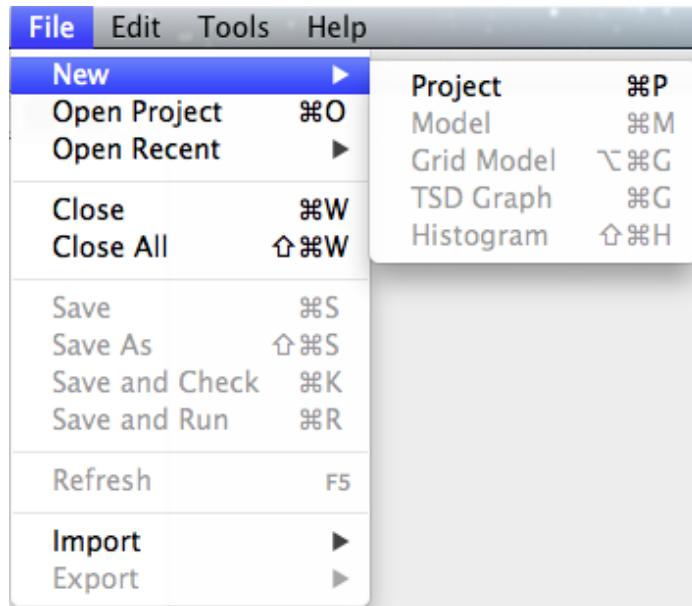
2 Project Management

Within iBioSim, all files are collected within projects. A project is a collection of models, analysis views, learn views, and graphs. As shown below, iBioSim displays all project files on the left; the open models, views, and graphs on the right; and a log of all external commands on the bottom. The menu bar is located on the top of the window in the Windows and Linux versions. It is located on the top of the screen in the MacOS version.



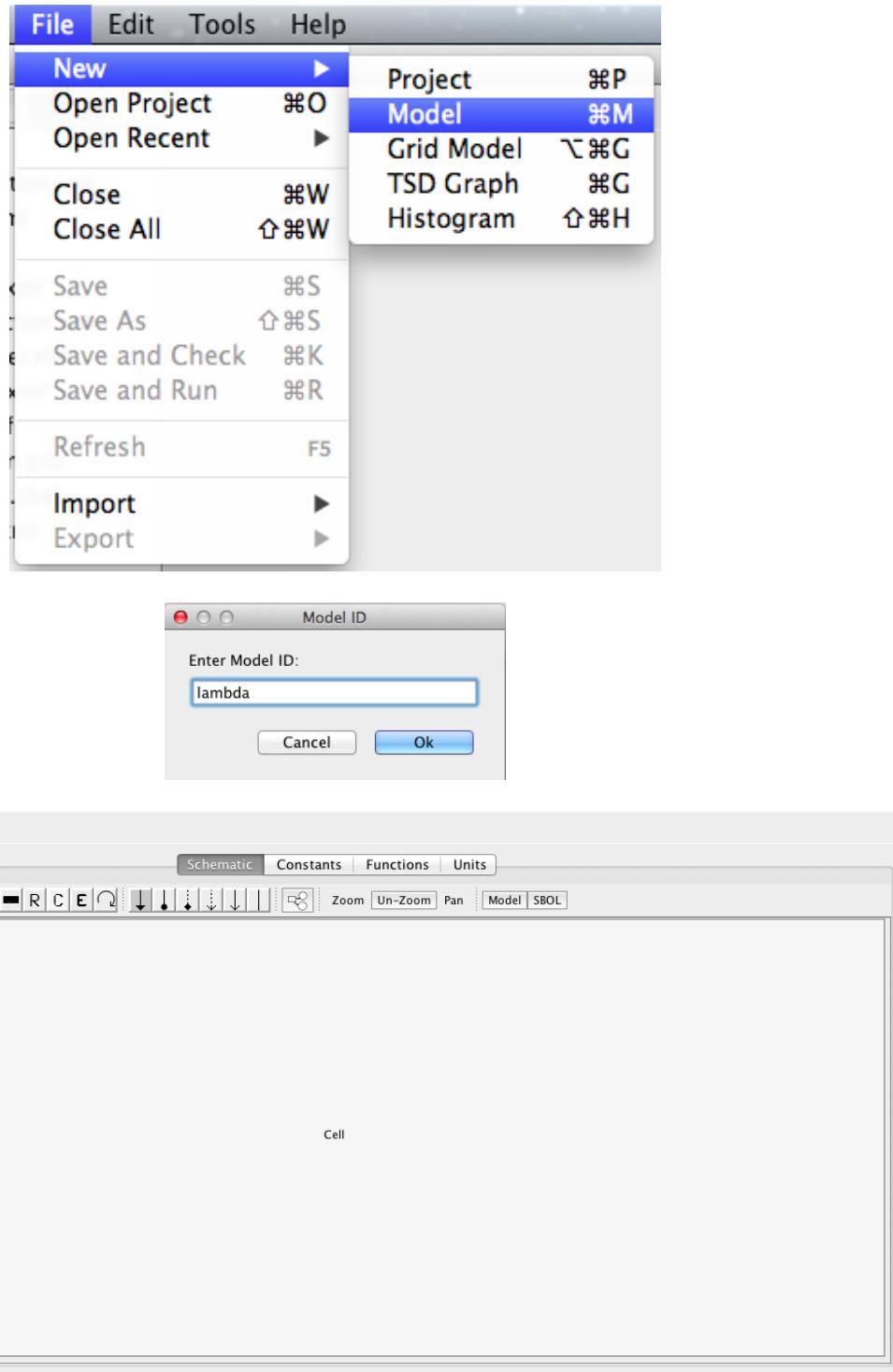
2.1 Creating and Opening Projects

To create a new project, select New → Project from the File menu as shown below. You will then be prompted to browse to a desired location and to give a name to the project directory. After you do this, click the new button and a new project directory will be created. To open a project, select Open Project from the File menu. You will then be prompted to browse to a project directory to open, and clicking open will open the project. You may also open a project by selecting one of your ten most recently opened projects by selecting the project name shown in the Open Recent menu within the File menu.

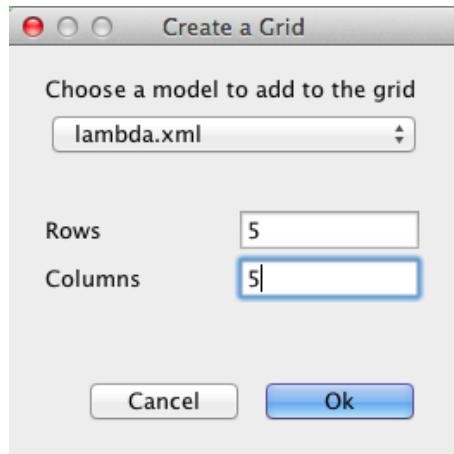


2.2 Creating Models and Graphs

After you have created or opened a project, you can create a new model or graph to add to the project. To create a new model, select New → Model from the File menu as shown below. You will then be prompted to enter a model ID. At this point, a Model editor (see Section 3) will open in a new tab.

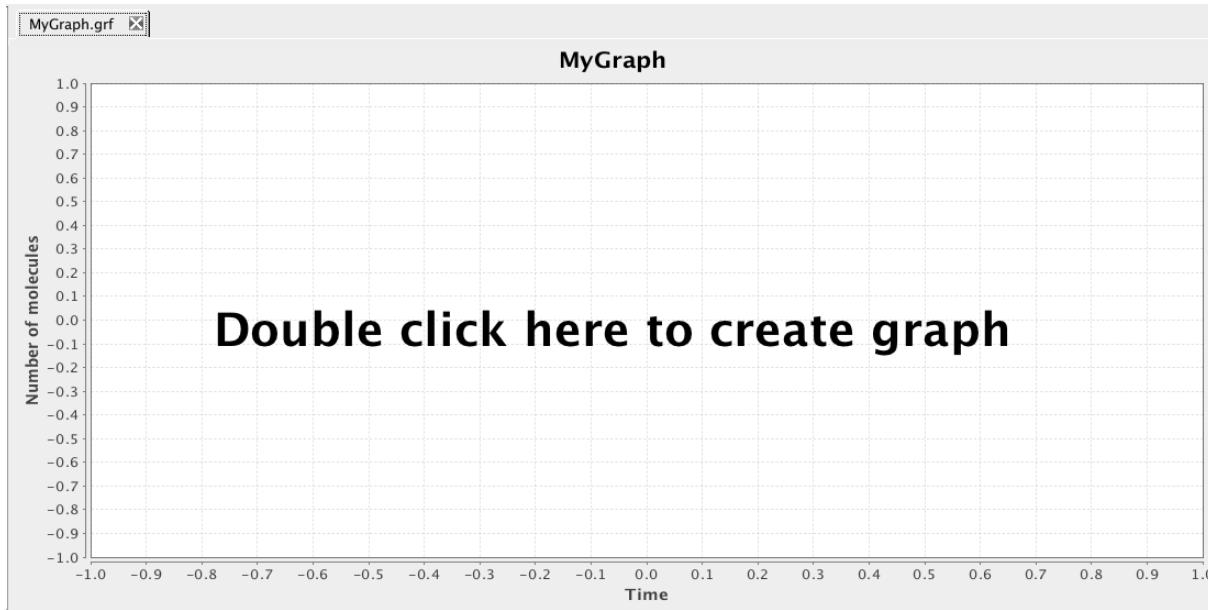


To create a new grid model, select New → Grid Model from the File menu as shown below. You will then be prompted to enter a model ID. Next, a Create Grid window opens that asks for the number of rows and columns in the grid. It also asks which model to use to populate the grid initially. You can select “none” if you wish to populate it later. At this point, a Grid Model editor (see Section 3) will open in a new tab.

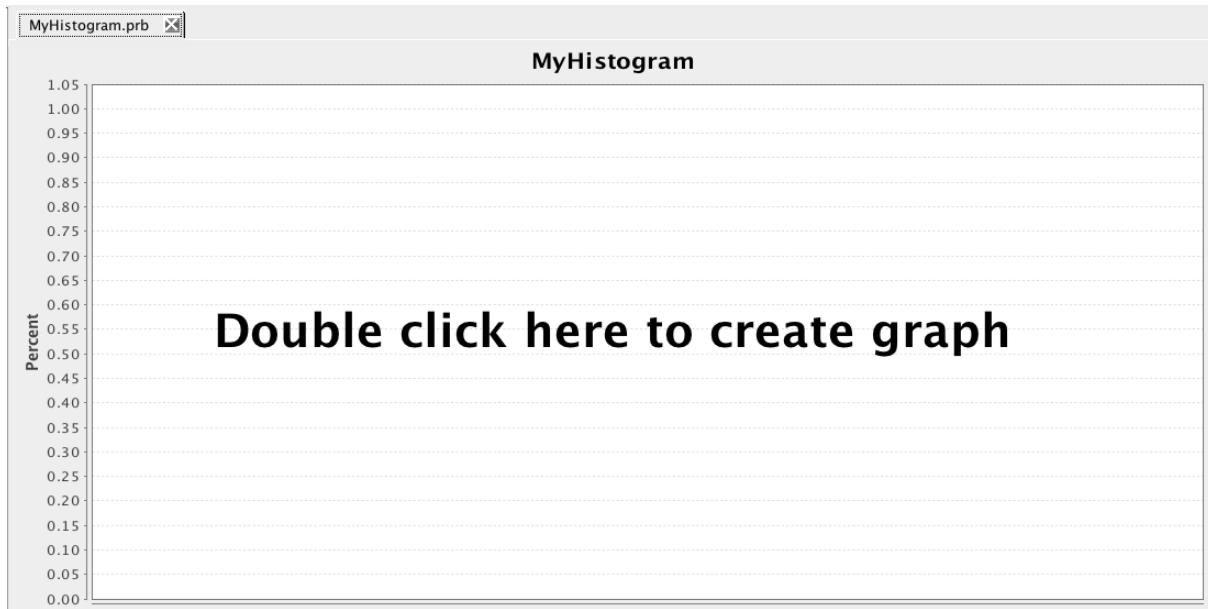


| | | | | |
|---------------|---------------|---------------|---------------|---------------|
| C1 lambda | C2 lambda | C3 lambda | C4 lambda | C5 lambda |
| C6 lambda | C7 lambda | C8 lambda | C9 lambda | C10 lambda |
| C11 lambda | C12 lambda | C13 lambda | C14 lambda | C15 lambda |
| C16 lambda | C17 lambda | C18 lambda | C19 lambda | C20 lambda |
| C21 lambda | C22 lambda | C23 lambda | C24 lambda | C25 lambda |

To create a new TSD graph, select New → TSD Graph from the File menu. You will then be prompted to give a name to the TSD graph. At this point, a TSD graph editor (see Section 7) will open in a new tab.



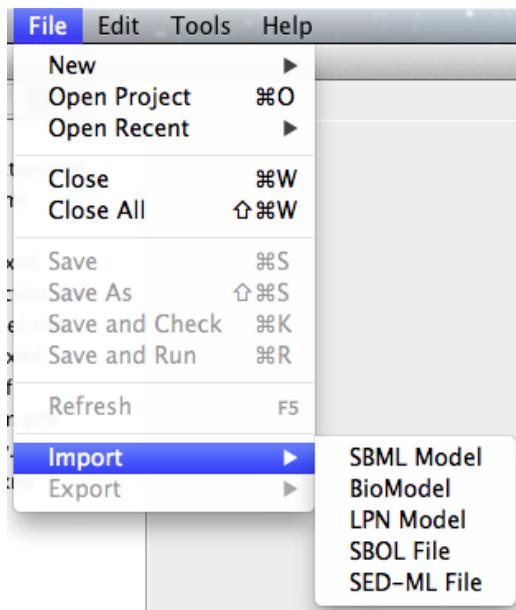
To create a new histogram, select New → Histogram from the File menu. You will then be prompted to give a name to the histogram. At this point, a histogram editor (see Section 8) will open in a new tab.



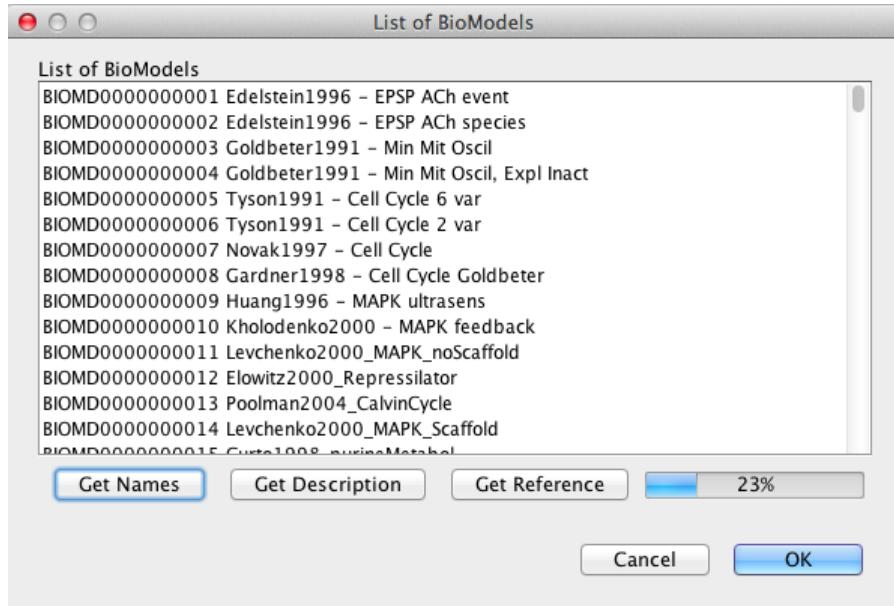
Once a model or graph is created, it can be opened again later by right-clicking on the object in the project window and selecting “View/Edit” or, alternatively, double-clicking on the object. Note that a model can be opened in either a graphical or tabular editor. An open model or graph can be closed by either clicking on the “X” in the tab or by selecting Close from the File menu. The Close All option in the File menu closes all tabs.

2.3 Importing Models

You can also import files into the current working project. These can be models in the *Systems Biology Markup Language* (SBML), models from the *BioModels database*, *Labeled Petri Net* (LPN) models, a *Synthetic Biology Open Language* (SBOL) file (see Section 4), or a *Simulation Experiment Description Markup Language* (SED-ML) file.

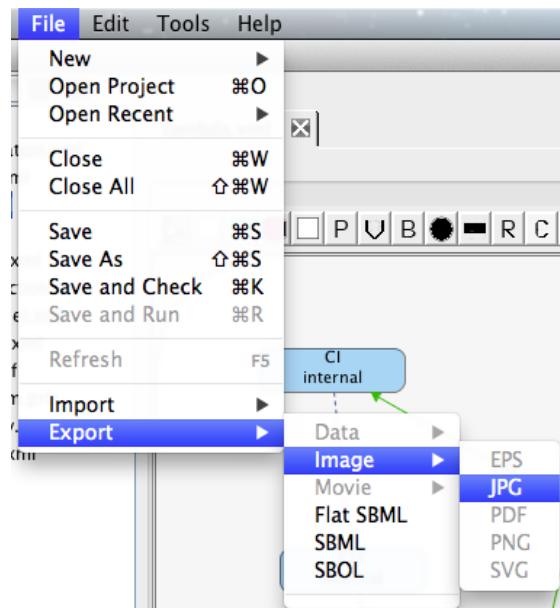


To import a SBML, select Import → SBML Model from the File menu. You will then be able to browse to find a model to import. After selecting the desired model, click the import button to bring the model into the project. Before bringing the model into the project, it will be checked to see if it is a valid SBML file. The model will also be checked for consistency, and any errors or warnings will be reported. These should be corrected before analysis of the model is performed. Importing of an LPN model, an SBOL file, or a SED-ML file are similar in that you are asked to locate the appropriate file to import. When importing a model from the BioModels database, the window below will open, initially only listing the BioModel numbers. Selecting “Get Names” will fetch the model names from the database. Beware that this can take a significant amount of time. When you click on a model to select it, you can use the “Get Description” button to fetch a description of the model which will open in a browser. Similarly, the “Get Reference” button will fetch the reference describing this model and again open it in a browser.



2.4 Exporting Models and Graphs

You can also export a model in either SBML or SBOL format. To export an SBML model, select Export → Flat SBML or Export → SBML. Then, you will be able to browse to find a location to export the SBML file. A flat SBML file is a single flattened chemical reaction model while an SBML file is a single file hierarchical model with high-level genetic constructs. When exporting model as an SBOL component, you must browse to find an SBOL library. The SBOL component will then be added to the selected library. Finally, you can export a JPG image of a model.



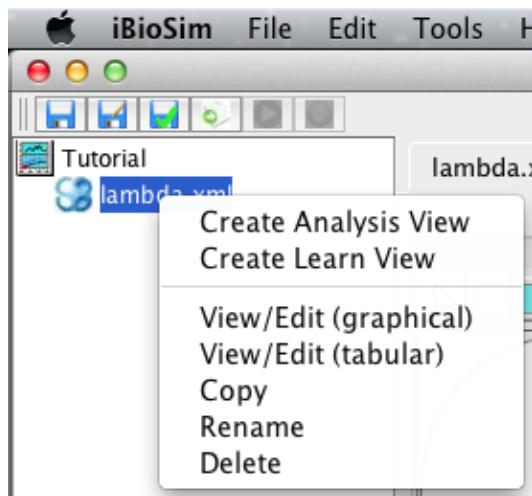
Data and graphs of time series data or histograms can be exported from the project in many formats. The supported file formats are:

- Time series data format (tsd), see Section 11.

- Comma separated value (csv).
- Column separated data (dat).
- Encapsulated postscript (eps).
- Joint Photographic Experts Group (jpg).
- Portable document format (pdf).
- Portable network graphics (png).
- Scalable vector graphics (svg).

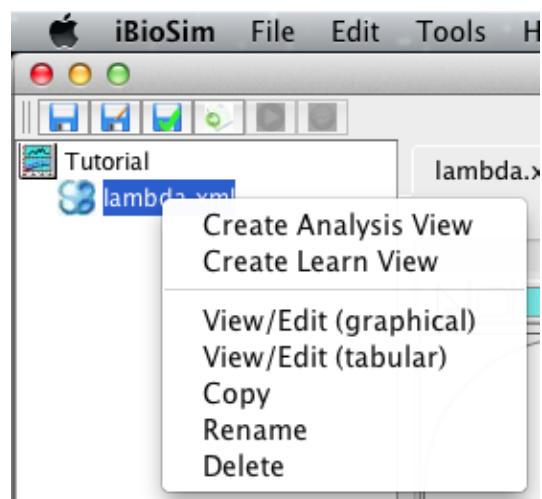
2.5 Editing Project Objects

All project objects can be modified by highlighting the object and using a right mouse click to open a menu of options, as shown below. Using this menu, every type of object can be copied, renamed, or deleted (these actions can also be done from the Edit menu). For a model file, the “View/Edit (graphical)” option opens the graphical model editor while the “View/Edit (tabular)” opens the tabular model editor (see Section 3). For a TSD graph, the “View/Edit” option opens the TSD graph editor (see Section 7). For a histogram, the “View/Edit” option opens the histogram graph editor (see Section 8). For an analysis view, the “Open Analysis View” option opens the analysis view (see Section 5). For a learn view, the “Open Learn View” option opens the learn view (see Section 6).



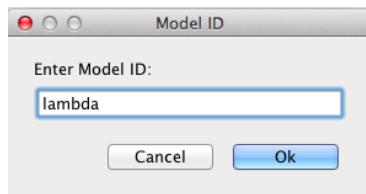
2.6 Creating Tool Views

To perform analysis or learning, right-click on a model and select “Create Analysis View” (see Section 5) to perform analysis or “Create Learn View” (see Section 6) to perform learning. You will then be prompted to give a name to your analysis or learn view. After a name is entered, a tab with the newly created view will open. Views can also be created using the Tools menu. Once a view is created, it can be opened again later by right-clicking on an analysis directory and selecting “Open Analysis/Learn View” or, alternatively, double-clicking on the view.



3 Model Editor

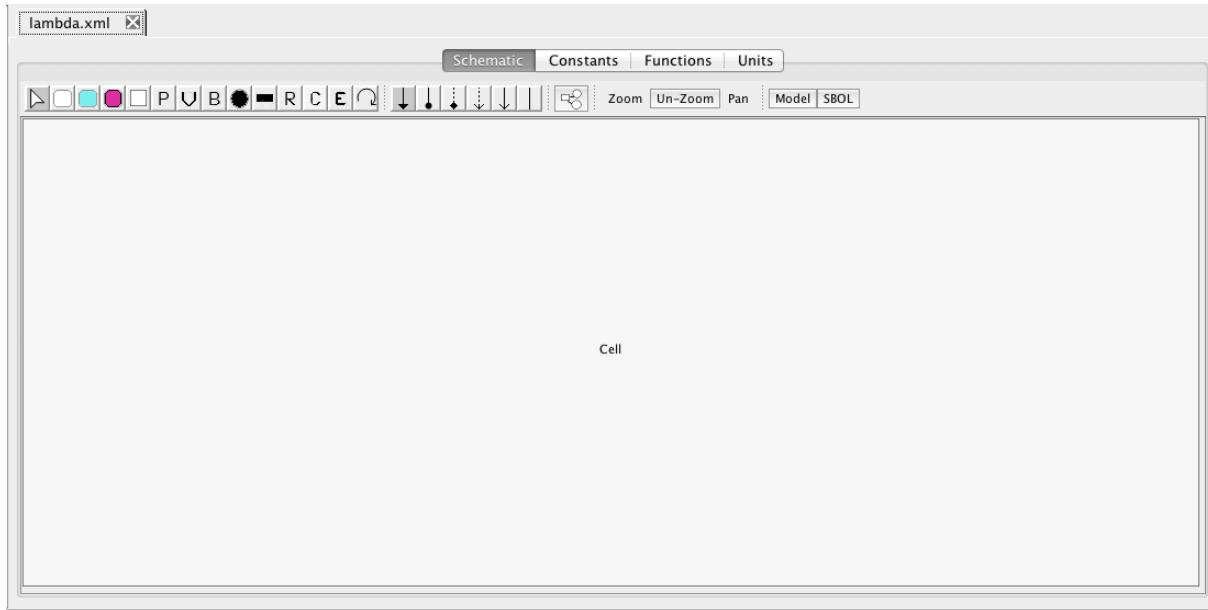
The model editor allows the user to create or modify a model of a genetic circuit or other biochemical system. iBioSim models are based on SBML Level 3 Version 1 models with added features to support visualization, hierarchy, and modeling of genetic regulation. SBML models can be imported and exported to allow interfacing with SBML-compliant tools. When exporting an SBML model, the hierarchy and genetic regulation can be flattened to create a single SBML model that uses only SBML L3V1 core constructs. The model editor includes both a schematic editor as well as several tabular editors. For large models, the schematic portion can also be edited tabularly instead. You can create both ordinary models and grid models.



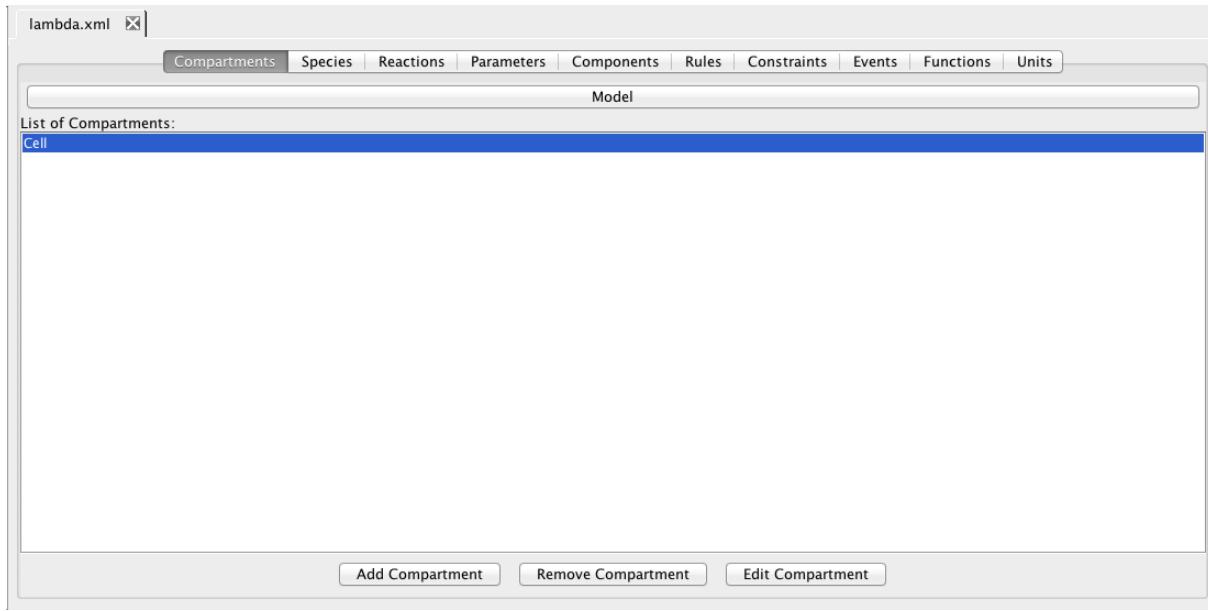
The schematic editor for an ordinary model is shown below. There are two groups of icons. By selecting the appropriate icon in the first group, the user can add *compartments* (see Section 3.1), *chemical species* (see Section 3.2), *chemical reactions* (see Section 3.3), *components* (see Section 3.4), *promoters* (see Section 3.5), *real variables* (see Section 3.6), *Boolean variables* (see Section 3.7), *Petri net places* (see Section 3.8), *Petri net transitions* (see Section 3.9), *rules* (see Section 3.10), *constraints* (see Section 3.11), and *events* (see Section 3.12) to the schematic. When the select icon is highlighted,

, double clicking on an object in the schematic opens the appropriate object editor. In this mode, relationships can also be created between objects by selecting an object and, while holding the mouse, dragging the connection to another object. The type of relationship (or *influence*) is determined by the icon that is highlighted in the second group (see Section 3.13). Finally, if the self regulation icon,

, is selected, clicking on a species creates a self influence relationship. The remaining items on the toolbar allow the user to apply an automatic layout routine



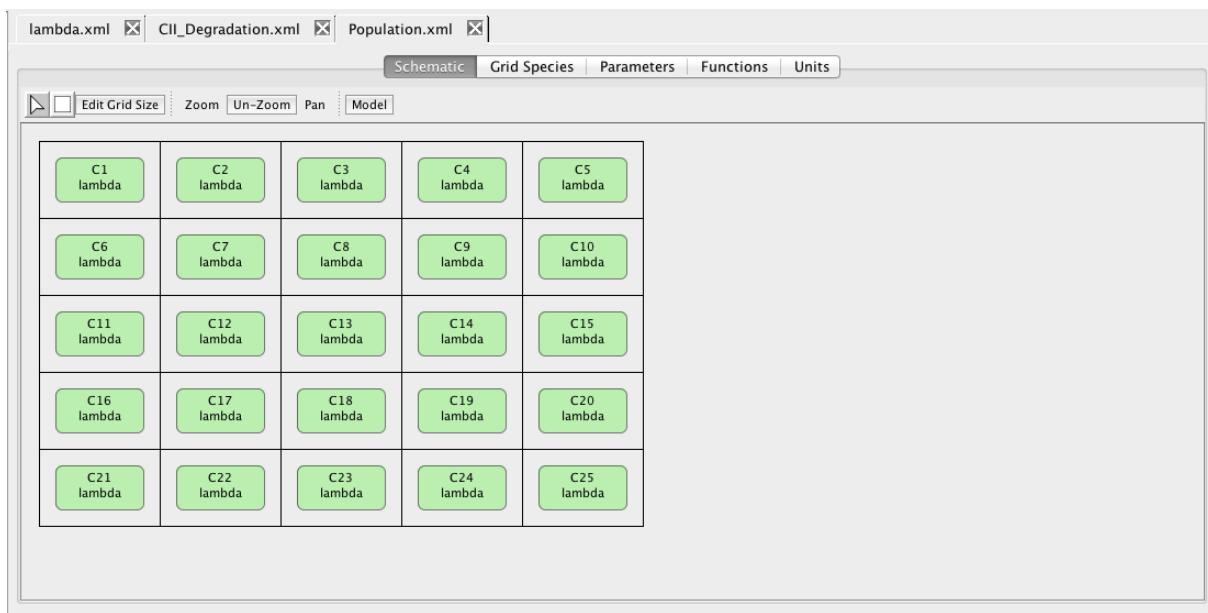
For large models, it may be preferable to use the tabular model editor shown below. The tabular editor includes all the same elements as the graphical editor except they are presented as lists of their Ids. In each list, the user can add new element, remove existing elements, or edit the properties of existing elements. Note that constants, real variables, Boolean variables, and places appear in the Parameter list, influences appear in the Reaction list, and Petri net transitions appear in the Event list.

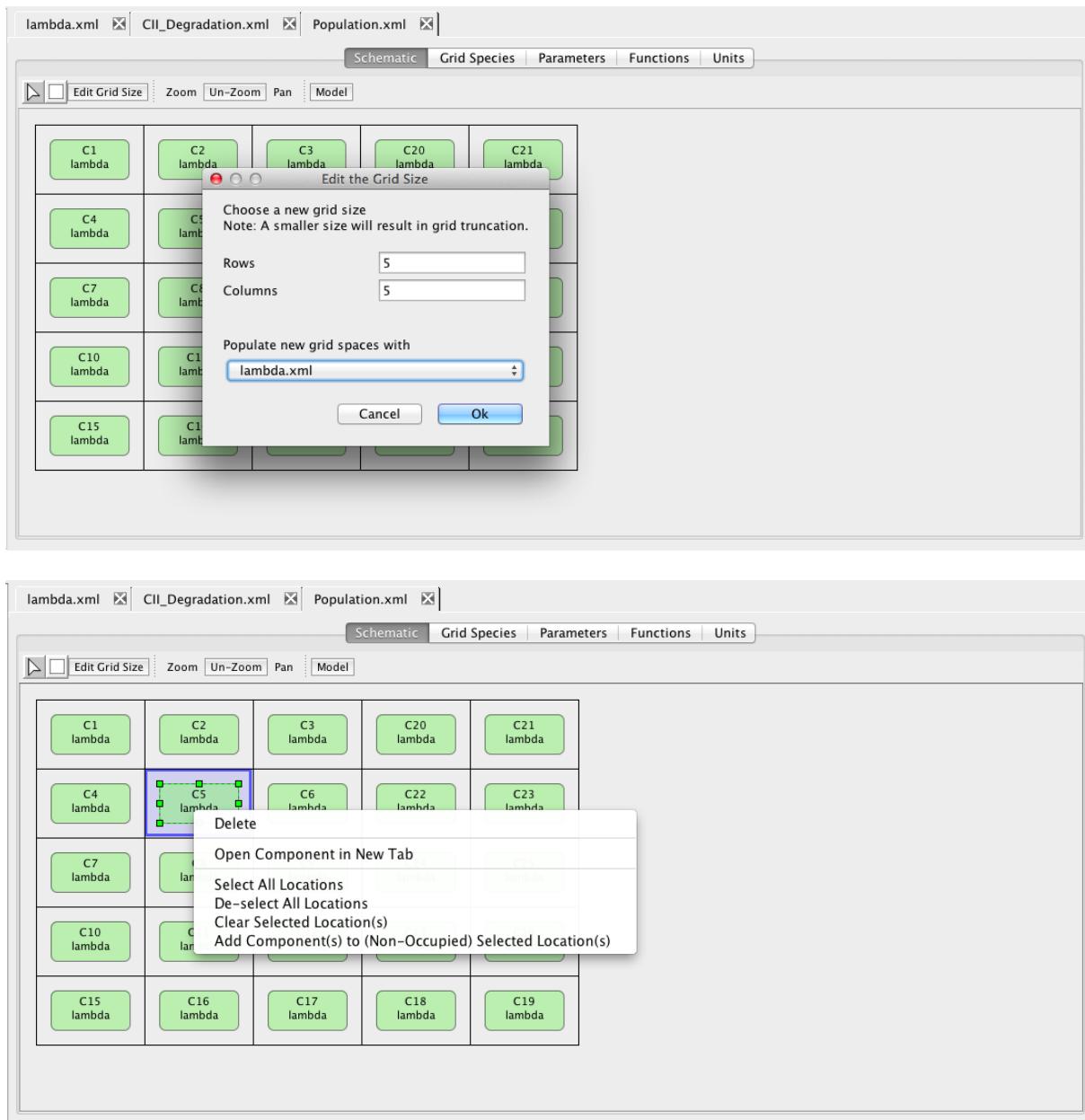


iBioSim also supports the creation of models on a grid. The idea of a grid is to provide a coarse-grain spatial representation. A single model can be instantiated within each grid location. The models can be for individual cells, allowing one to model population dynamics. The models can also be different segments of a single cell, allowing one to model diffusion within a cell. Key to grid-based modeling are diffusible species, described in Section 3.2. When the models within grid locations are compartment-enclosed components, reactions are added to represent the movement of these

diffusible species between the compartment and the area outside the compartment within the grid location (i.e., *membrane diffusion*). When performing analysis, reactions are also automatically added to represent the movement of these diffusible species between the grid locations.

The schematic for a model on a grid is a bit different than for an ordinary model, which is evident from the toolbar. Namely, a grid-based model can only include components, with at most one component per grid location. By pressing the Add Component icon  and clicking on an empty grid location, you are allowed to select a model to instantiate within that grid location. You can also select Edit Grid Size to change the size of the grid. If the grid is larger in one or more dimensions than before, you can select a model to populate the new grid locations. A smaller grid size simply removes the component instantiations that fall outside the new grid. You can also edit the grid by selecting individual grid locations by clicking on them or selecting multiple grid locations using the rectangle selection tool by left-clicking and dragging the mouse. When you then use the right mouse button, it brings up the menu shown below. This menu allows you to delete the component that was right-clicked on or to open its model in a Model Editor. You can also use this menu to select or de-select all grid locations, and you can either clear occupied, or populate non-occupied, selected grid locations. The grid model editor has one additional tab which allows the user to edit properties for the species that diffuse on the grid.





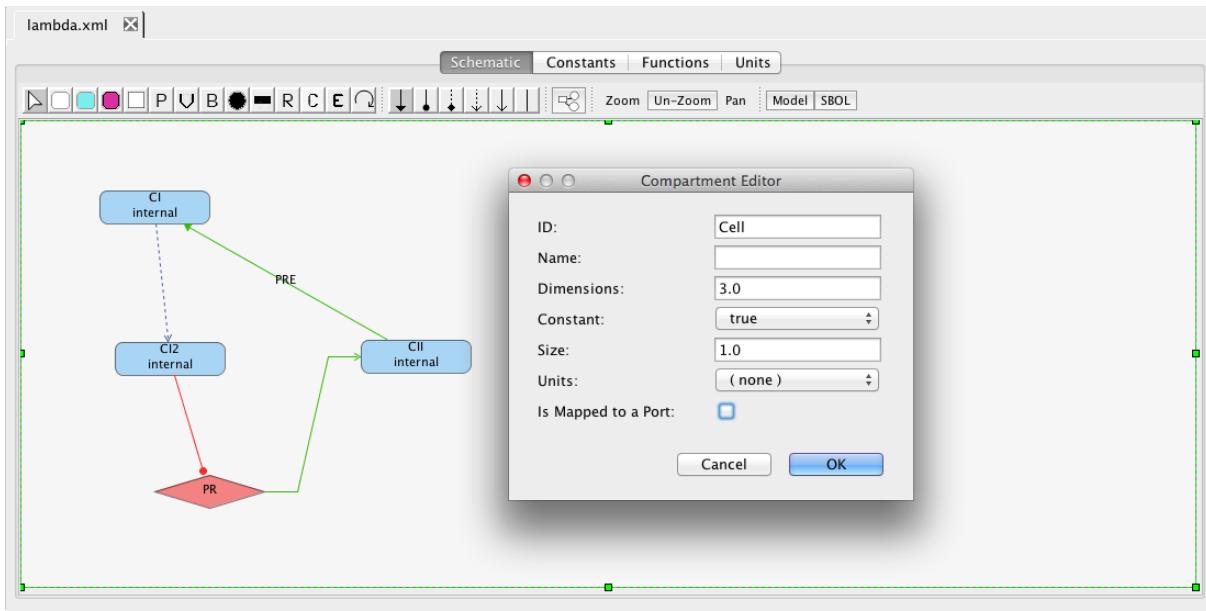
3.1 Compartments

Compartments are used to specify membrane-enclosed regions where species are found. A new model includes a compartment named **Ce11** by default. Note that when a compartment has species or reactions assigned to it that it cannot be removed without first removing or reassigning these species and reactions. To add a new compartment to the model, select the Add Compartment icon and click on the schematic canvas. This will drop a new compartment with default ID and other values. You may change these defaults by double-clicking on the compartment to open the Compartment Editor. As shown below, a compartment has the following fields:

- ID: a unique ID composed of only alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).

- Dimensions: number of spatial dimensions (default=3).
- Constant: Boolean indicating if the size is constant (default=true).
- Size: initial size of the compartment (default=1.0). This value can be a mathematical expression (see Section 10) which is called an *initial assignment*.
- Units: the units for the size (default=none).

Finally, a compartment can be mapped to a port (default=true). When a compartment is assigned to a port, it is assumed that the compartment is not membrane enclosed. This is indicated by the compartment having square corners. By assigning a compartment to a port, when this model is used as a subModel in another model, the compartment can be replaced with the one in the enclosing model. When a compartment is not assigned to a port, it is assumed that the compartment is membrane enclosed which is indicated with rounded corners.



3.2 Species

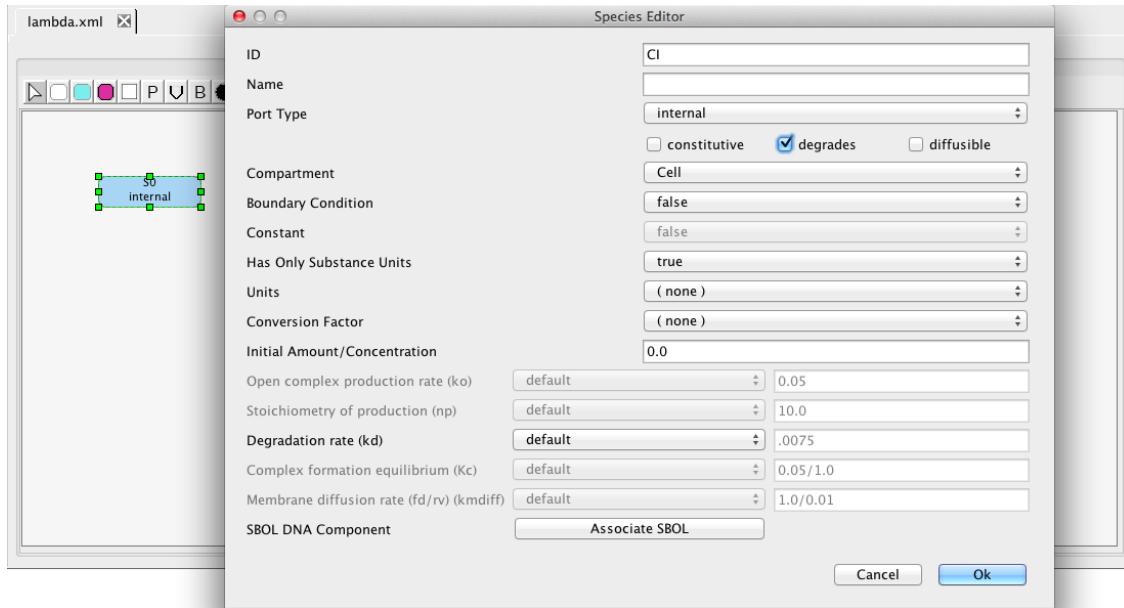
Species are the molecules, such as proteins, that are produced by genes or chemical reactions. To add a species to the model, select the Add Species icon

 and click on the schematic canvas. This will drop a new species with default ID and other values. You may change these defaults by double-clicking on the species to open the Species Editor. A species has the following elements:

- ID: a unique ID composed of only alphanumeric characters and underscores.
- Name: an arbitrary string description of the species (optional).
- Port Type: used to indicate how this species can be connected in hierarchical models. The *input* type is used to indicate a species that is produced outside this model, the *internal* type is used to indicate that a species is produced inside this model but cannot be used in other models, and the *output* type is used to indicate that a species is produced by this model and can be used in other models (default=internal).

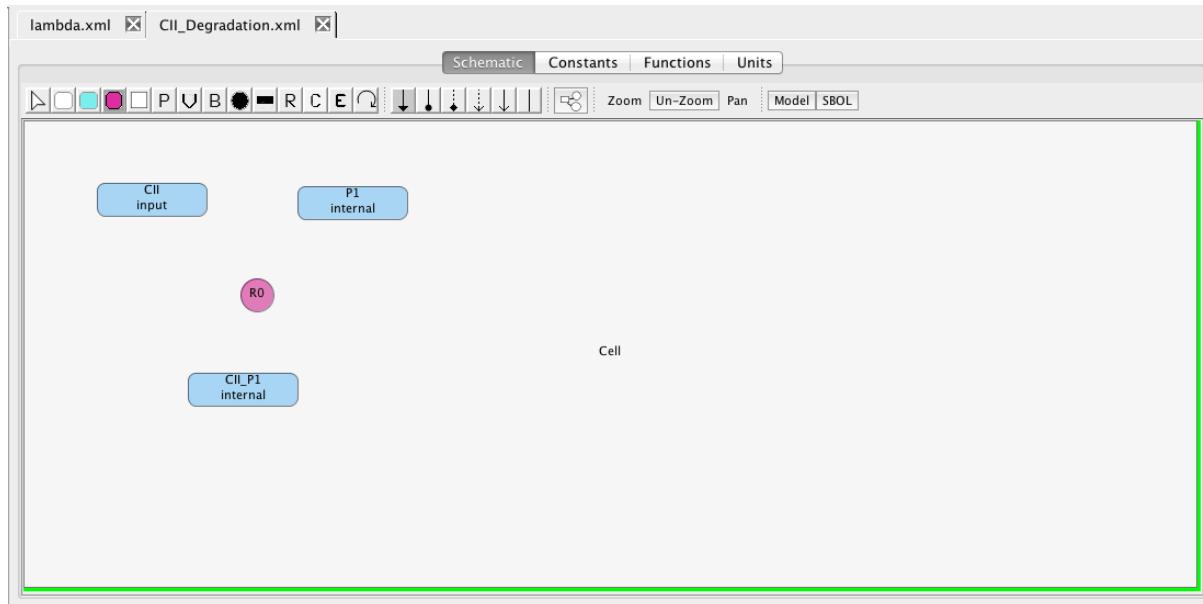
- Constitutive: this checkbox indicates that a default production reaction should be created for this species (default=false).
- Degrades: this checkbox indicates that a default degradation reaction should be created for this species (default=false).
- Diffusible: this checkbox indicates if this species can diffuse in multi-compartment and grid-based models (default=false).
- Compartment: location of the species (default is set by where the species is placed in the schematic).
- Boundary Condition: Boolean indicating if the species amount/concentration cannot be changed by reactions (default=false).
- Constant: Boolean indicating if the species amount/concentration is constant (default=false).
- Has Only Substance Units: Boolean indicating if the species is to be considered an amount in mathematical equations (default=true).
- Units: the units for the amount/concentration (default=none).
- Conversion Factor: is a constant global parameter, the value of which is used to convert this species' units into the units used for extent (i.e., the units of change due to reactions).
- Initial Amount/Concentration: initial value of the amount or concentration of the species (default=0.0). This value can be a mathematical expression (see Section 10) which is called an *initial assignment*. If the value/expression is enclosed in brackets (i.e., [e]), it is a concentration, otherwise it is an amount.
- Open complex production rate (ko): once RNAP binds to this promoter, this is the rate at which transcription is initiated. Note that this production rate is used in the production reaction created when a species is marked as constitutively produced. The default value is the global parameter ko.
- Stoichiometry of production (np): the average number of proteins that are produced by an mRNA before it degrades. This value is used in the production reaction created when a species is marked as constitutively produced. The default value is the global parameter np.
- Degradation rate (kd): the degradation rate for this species. This value is used in the degradation reaction created when a species is marked that it degrades. The default is the value of the global parameter kd.
- Complex formation equilibrium (Kc): the equilibrium constant for formation of this species when it is produced by a complex formation reaction. The default is the global parameter Kc. The equilibrium constant can be specified as a forward and reverse rate constant using the < forward rate >/< reverse rate > form.
- Membrane diffusion rate (fd/rv) (kmdiff): the forward (in to out) and reverse (out to in) rate constants (slash-separated) for diffusion of this species through its compartment's membrane. Note that when membrane diffusion reactions are generated, the rates are taken from the inner species. If no reverse rate is given, a default of 1.0 is used. The default is the global parameter kmdiff.

Finally, one can specify an SBOL DNA Component with this species by clicking on the Associate SBOL button. More details can be found in Section 4.



3.3 Reactions

Reactions are used to create or destroy molecular species in a biochemical reaction network. To add a reaction, select the Add Reaction icon and click on the schematic canvas which drops a new reaction with a default ID and parameter values.

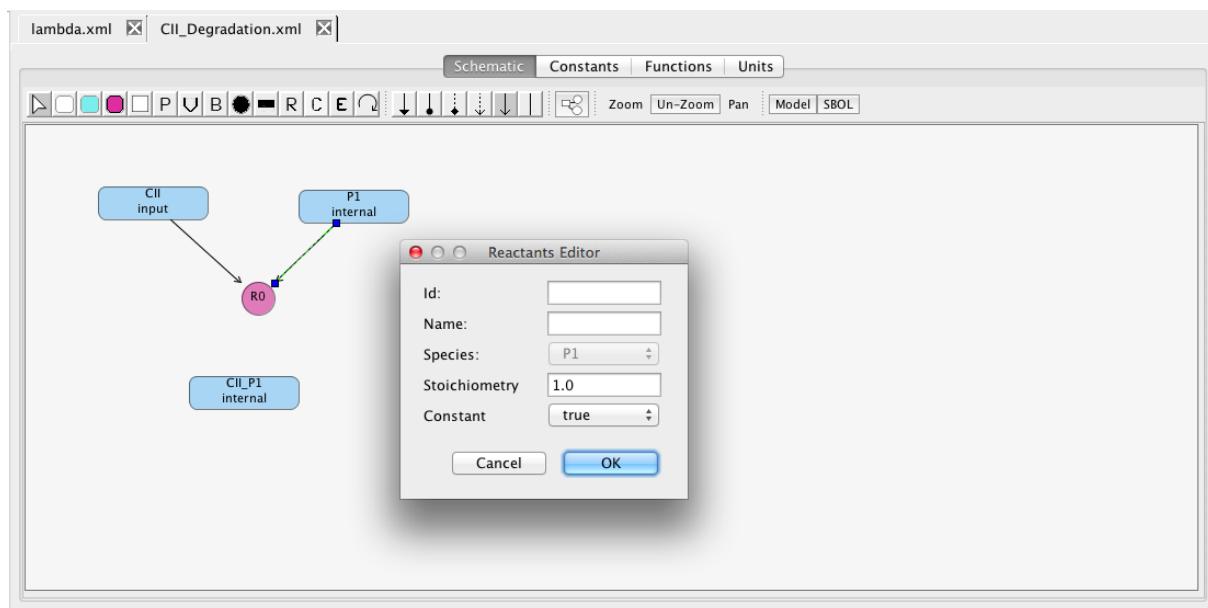


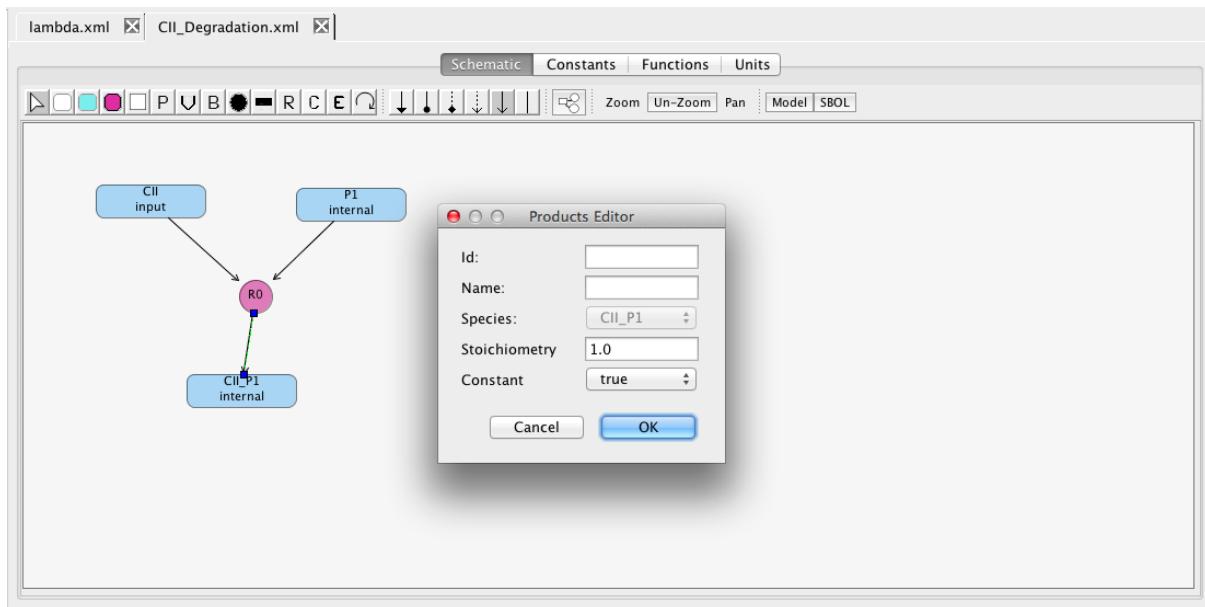
Reactions convert reactant species into product species with perhaps modifier species (e.g., enzymes or catalysts) affecting the rate of this conversion. The next step, therefore, is to indicate which species are reactants, products, and modifiers for the reaction. This is accomplished by

selecting the Reaction icon  or Modifier icon . With the Reaction icon selected, selecting a species, holding the mouse button, and dragging the arc to a reaction makes the selected species a reactant in that reaction. Similarly, selecting a reaction, holding the mouse button, and dragging the arc to a species makes the species a product in the selected reaction. Finally, with the Modifier icon selected, dragging an arc between a species and reaction (in either direction) adds the species as a modifier in the reaction. When you select a reactant edge, it opens a Reactants Editor. This editor allows you to change the following things:

- ID: this is the ID for this reactant and should not be confused with the Species ID. This ID can be used in equations such as rules and event assignments. Use this ID when you want the stoichiometry to be determined by an equation or used in an equation. The ID is optional.
- Name: an arbitrary string description of the species (optional).
- Species: the ID of the reactant species.
- Stoichiometry: the number of molecules consumed by the reaction (default=1.0). Note that this value will be overridden when the reactant ID is used on the left-hand side of an assignment.
- Constant: indicates whether the stoichiometry can change dynamically (default=true).

Similarly, the ID, Name, Species, Stoichiometry, and Constant fields can be edited for a Product edge as shown below. Modifier edges do not have any of these fields, so they cannot be edited.

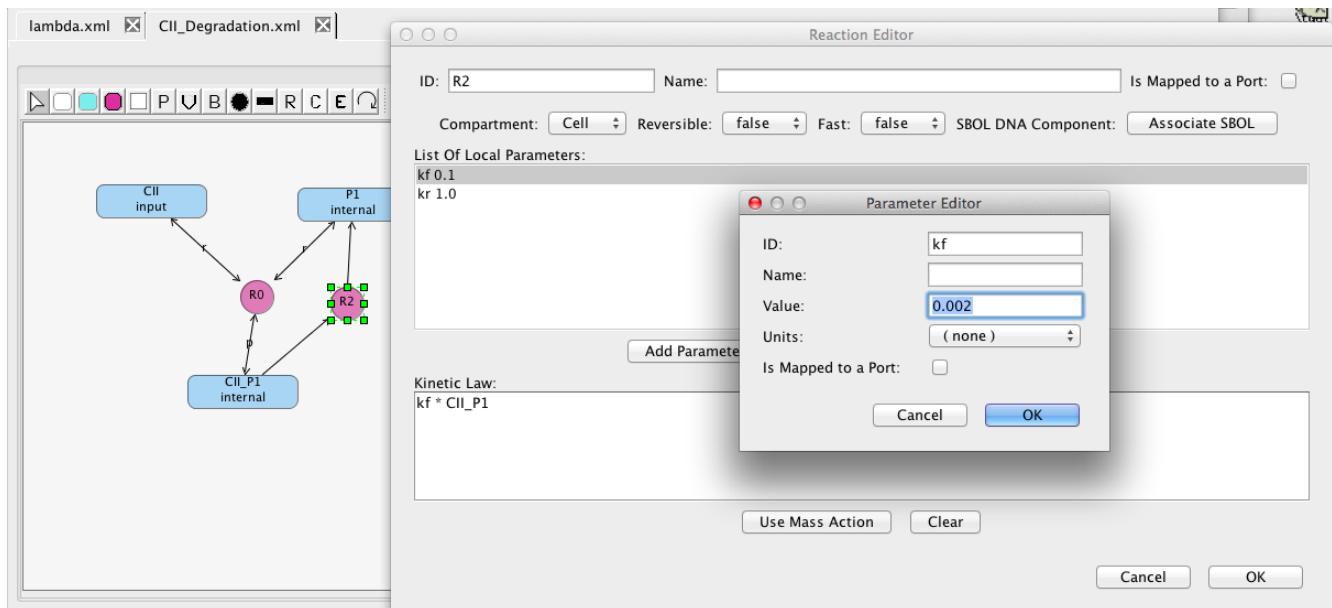
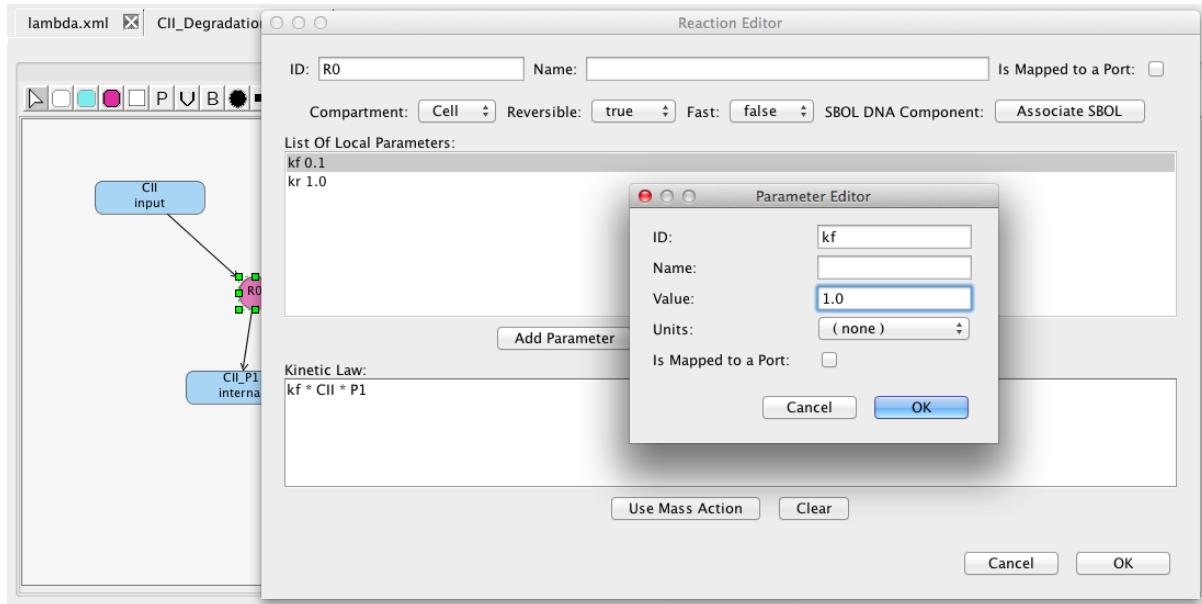




When you select a reaction, it opens the Reaction Editor which allows you to edit the following reaction fields:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Is Mapped to a Port: a checkbox indicating whether or not this reaction is mapped to a port, so it can be replaced or deleted in hierarchical models (default=false).
- Compartment: the location where this reaction takes place. This is set by the location in which the reaction is placed in the schematic.
- Reversible: a Boolean indicating if the reaction is reversible (default=false). Note that reversible reactions are indicated in the schematic with double headed arrows.
- Fast: a Boolean indicating if the reaction reaches equilibrium rapidly (default=false). Note that this only has limited support in analysis, so it should be used sparingly.
- SBOL DNA Component: the Associate SBOL button allows the user to annotate the reaction with SBOL (see Section 4).
- List of Local Parameters: symbolic values that can be used in the *kinetic law* for this reaction. Their IDs only need to be unique to this reaction. Local parameters can be added, removed, and edited using the appropriate buttons. Each parameter is composed of an ID, Name (optional), Value, Units (optional), and a checkbox indicating whether or not this parameter is mapped to a port. The list of parameters begins with a default forward reaction rate (kf) and reverse reaction rate (kr). These names and their values should likely be edited.
- Kinetic Law: a mathematical formula (see Section 10) describing the rate or probability for this reaction. The kinetic law can either be automatically generated using the Use Mass Action button or manually entered. The Use Mass Action button creates a rate law using the law of mass action assuming that the first parameter in the list is the forward reaction rate

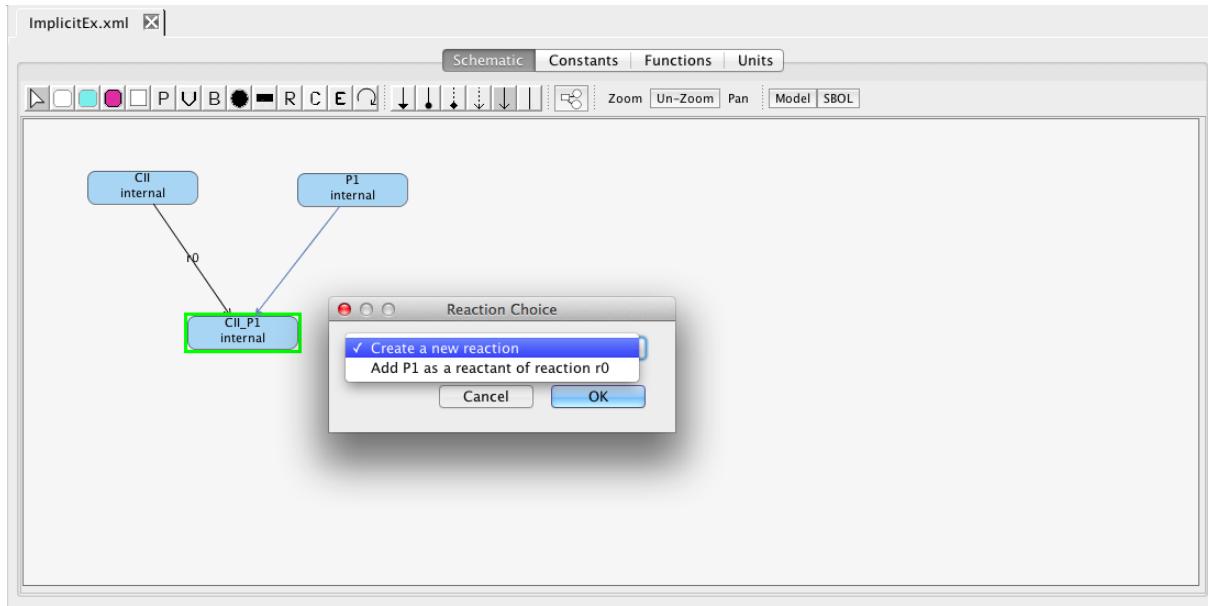
and the second parameter in the list is the reverse reaction rate. The Clear button clears the kinetic law editor. Note that the kinetic law formula can only include species IDs that are reactants, products, or modifiers to this reaction.



To simplify a schematic, one can use implicit reactions in some limited situations. To create an implicit reaction, with the Reaction icon highlighted and the mouse button selected, drag a reaction edge between two species. This action creates a reaction with a default ID and parameters, including the source species as a reactant and sink species as a product. If you connect an additional reaction edge to either species, you are asked if you wish to create a new reaction or add this new relationship to an existing reaction on the source and/or sink species. Implicit reactions can only be used when the following conditions hold:

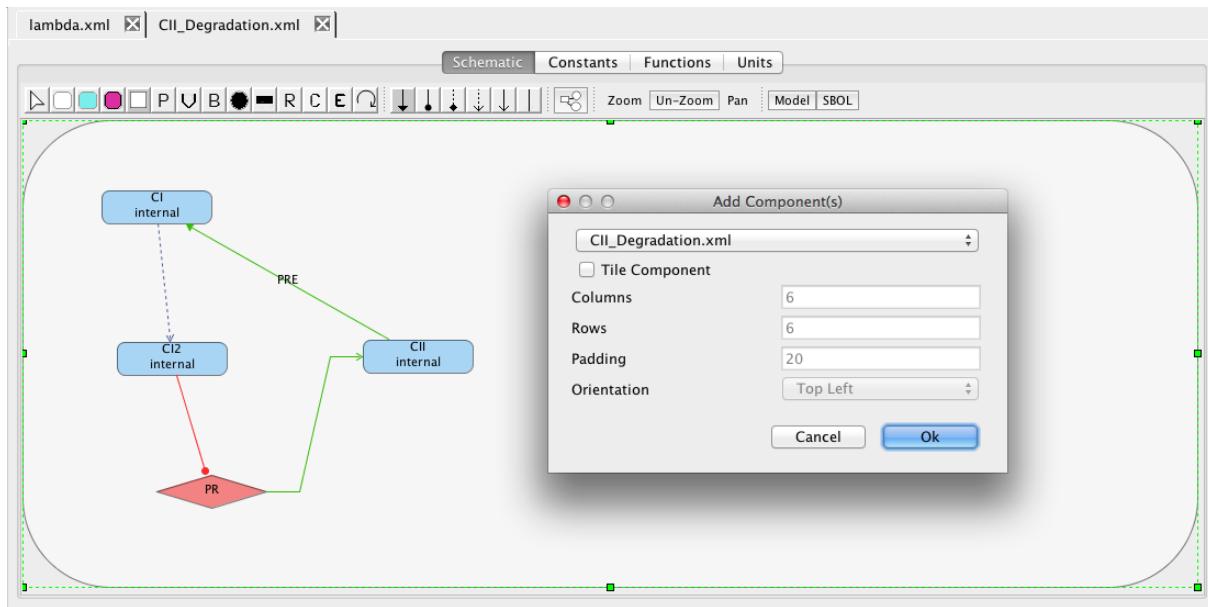
- The stoichiometry of all reactants and products for the reaction is 1.
- The reaction has either a single reactant OR a single product.
- The reaction has no modifier species.

The first condition is a default and cannot be changed. Adding edges can, however, cause a violation of the second or third condition. If this violation occurs, an explicit reaction is created to replace the implicit reaction.

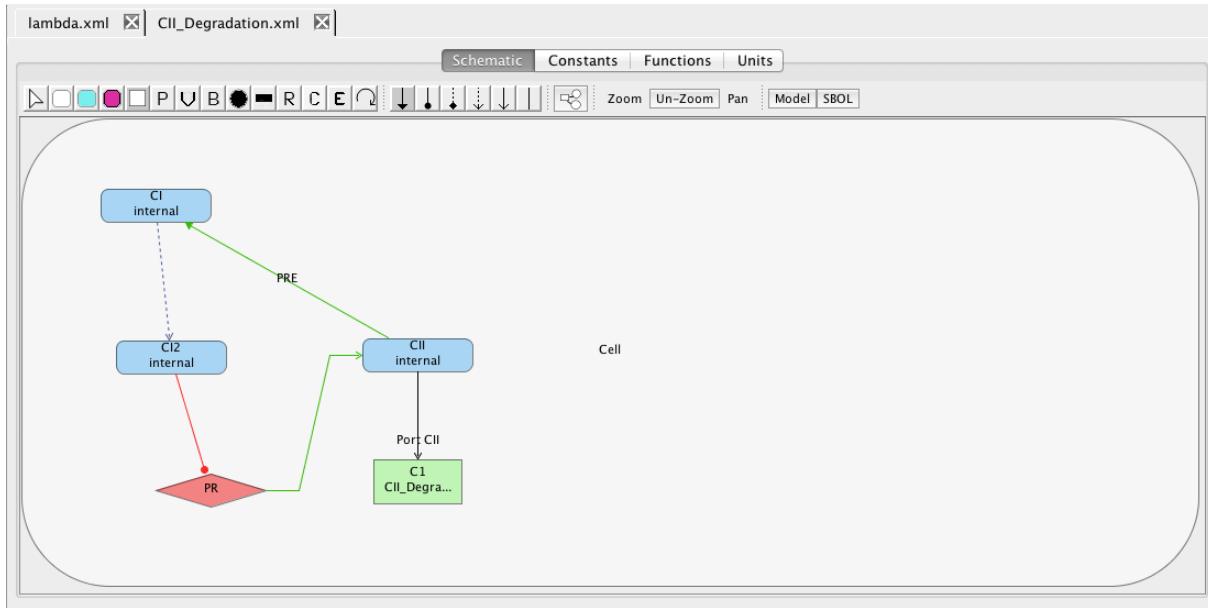


3.4 Components

Components are instances of other models within the project, and they are used to create hierarchical, multi-compartment, and cellular population models. To add a component, select the Add Component icon and click on the schematic canvas. You are then prompted to select the model for the component that you wish to add. This window also allows you to add multiple instances of the component at one time. To do this, select Tile Component and specify how many Columns, Rows, the Padding between them that you want, and whether to start the tiling in the top left corner of the schematic or at the mouse location.



When a component is added, it appears in the schematic either with sharp or rounded corners. A component with sharp corners is not enclosed within a compartment (i.e., its compartment is on a port). In other words, this is likely part of a hierarchical model. On the other hand, rounded corners indicate that it is enclosed within a compartment. In this case, the component is likely to be part of either a multi-compartment or cellular population model. If a diffusible species with the same ID exists inside and outside of a compartment-enclosed component, a membrane diffusion reaction will be created, allowing the species to be transported to/from the compartment-enclosed component. If the compartment-enclosed component is on a grid, the outside diffusible species will be automatically created if the inside diffusible species exists. When a component has ports, connections to input ports can be made by selecting an object and, while holding the mouse button, drag an arc to the component. If there are multiple input ports, you are asked which input port to connect this object to. To connect to output ports, begin by selecting the component and drag an arc to the object to be connected to the output port. Note that when you make these connections you are specifying that the object in the top-level model is the same object as the one in the component that it is connected too (i.e., it creates a *replacement*).

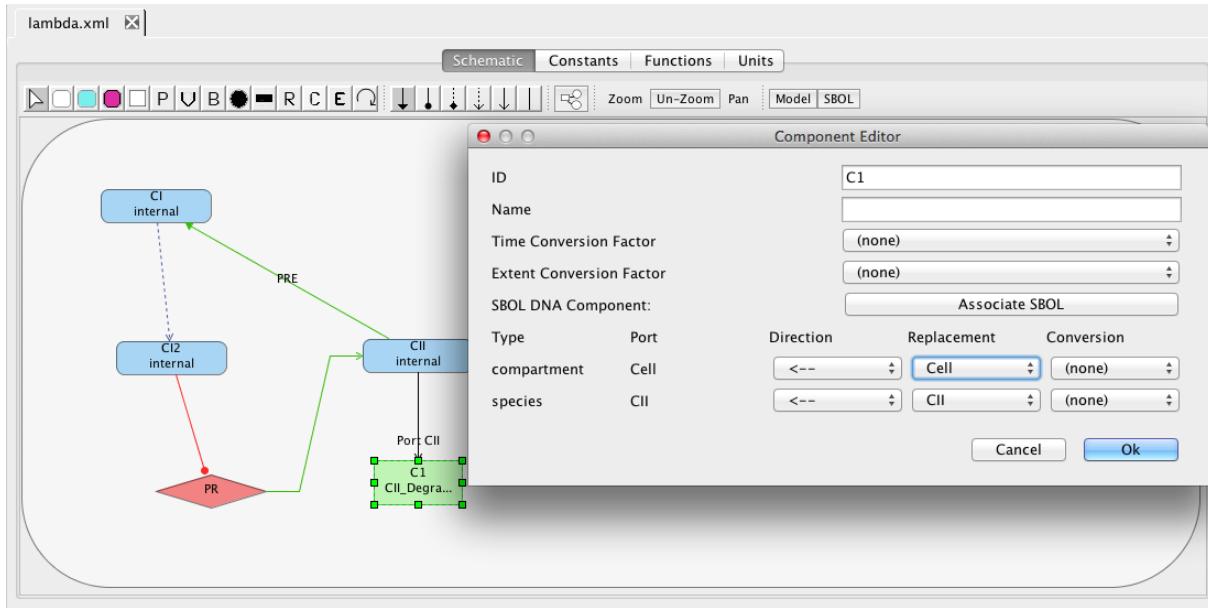


Clicking on the component opens the Component Editor which allows you to edit the following fields:

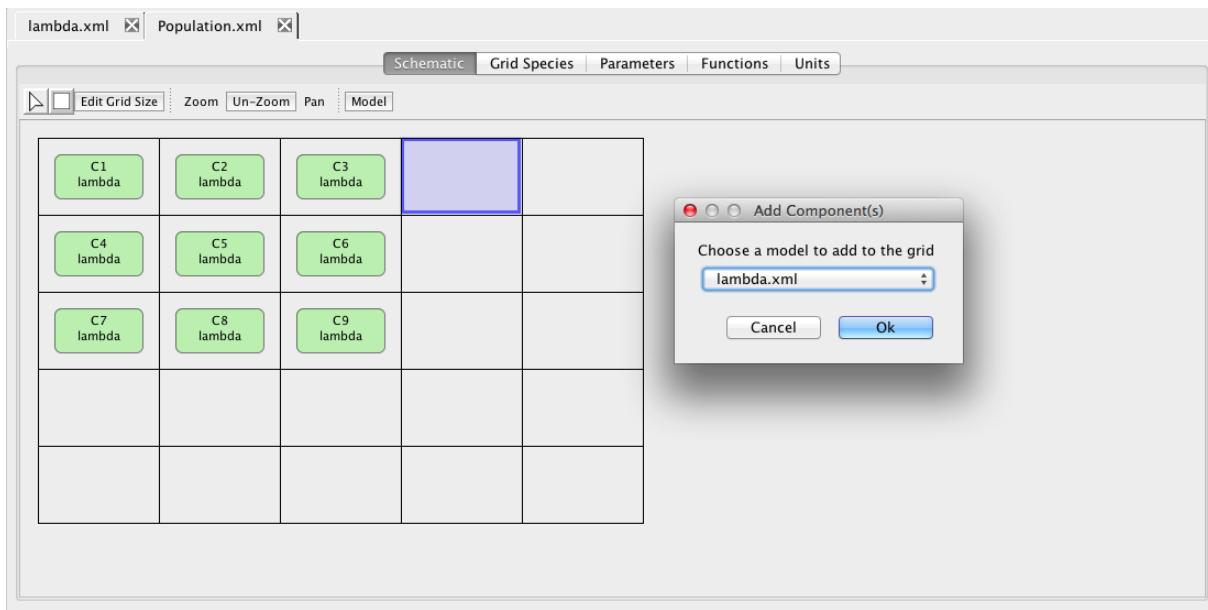
- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Time Conversion Factor: is a constant global parameter, the value of which is used to convert the submodel's time units into the units used in the model that includes it.
- Extent Conversion Factor: is a constant global parameter, the value of which is used to convert the submodel's extent units into the units used in the model that includes it.
- SBOL DNA Component: the Associate SBOL button allows the user to annotate the reaction with SBOL (see Section 4).

The last part of the Component Editor is a table of replacements and deletions. This table lists all items that are mapped to ports within the submodel. For each port, it includes:

- Type: the type of SBML element.
- Port: the ID of the SBML element.
- Direction: the direction of replacement. A \leftarrow indicates that the item in the enclosing model replaces the one in the submodel. A \rightarrow indicates that the item in the enclosing model is replaced by the one in the submodel.
- Replacement: this indicates the ID of the item in the enclosing model that is involved in the replacement. If the replacement selection is “delete” then the object in the submodel is deleted when this model is flattened. If the replacement selection is “none” then there is no replacement or deletion on this port.
- Conversion: is a constant global parameter, the value of which is used to convert the submodel's units for this element into the units used by the item replacing it within the enclosing model.



When adding a component to a grid, you are simply asked which model to instantiate, and you cannot connect to ports as species are not allowed on a grid.

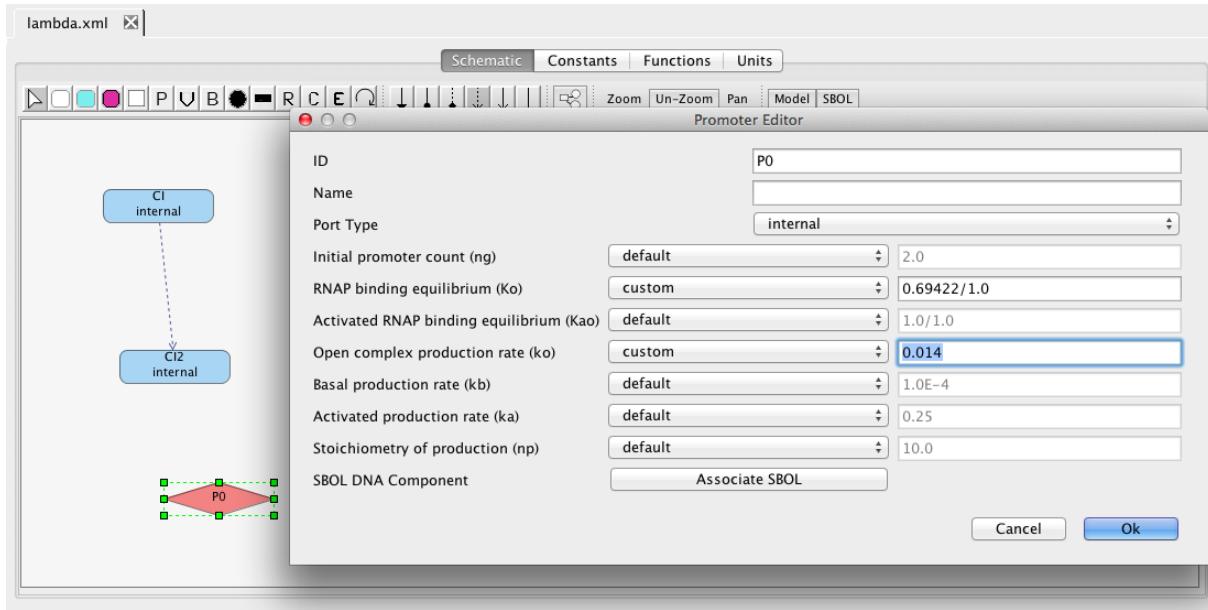


3.5 Promoters

Promoters are the locations on a DNA sequence in which transcription is initiated to create a *message* RNA (mRNA), which can then be translated by a ribosome into a protein. To add a promoter to a model, select the Add Promoter icon and click on the schematic canvas, which creates a new promoter with a default ID and parameter values. Clicking on the promoter opens the Promoter Editor and allows you to edit the following fields:

- ID: a unique ID composed only of alphanumeric characters and underscores.

- Name: an arbitrary string description (optional).
- Port Type: used to indicate how this object can be connected in hierarchical models. The *input* type is used to indicate a object that is produced outside this model, the *internal* type is used to indicate that a object is produced inside this model but cannot be used in other models, and the *output* type is used to indicate that a object is produced by this model and can be used in other models (default=internal).
- Initial promoter count (ng): the number of copies of this promoter. The default is the value of the global parameter ng.
- RNAP binding equilibrium (Ko): the equilibrium constant to use for the binding of RNAP to this promoter. The default value is the global parameter Ko. Note that this can either be specified as a single equilibrium constant or as a forward rate “/” reverse rate.
- Activated RNAP binding equilibrium (Kao): the equilibrium constant to use for the binding of RNAP to a promoter with an activator. The default value is the global parameter Kao. Note that this can either be specified as a single equilibrium constant or as a forward rate “/” reverse rate.
- Open complex production rate (ko): once RNAP binds to this promoter, this is the rate at which transcription is initiated. Note that this production rate is only used for promoters with no activators. The default value is the global parameter ko.
- Basal production rate (kb): once RNAP binds to an activated promoter, this is the rate at which transcription is initiated without the activator also being bound. The default value is the global parameter kb.
- Activated production rate (ka): once RNAP binds to an activated promoter, this is the rate at which transcription is initiated when the activator is also bound to the promoter. The default value is the global value ka.
- Stoichiometry of production (np): the average number of proteins that are produced by an mRNA before it degrades. The default value is the global parameter np.
- SBOL DNA Component: the Associate SBOL button allows the user to annotate the promoter with SBOL (see Section 4).

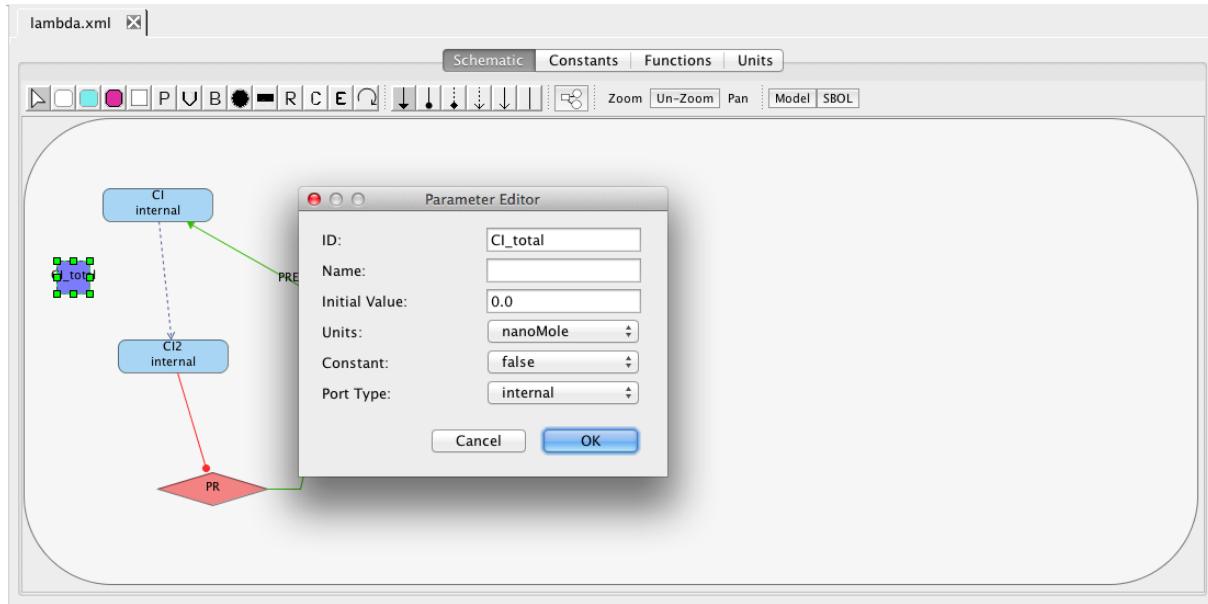


3.6 Real Variables

Real variables are non-constant global parameters that can be used in math formulas (see Section 10). A parameter includes the following:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Initial value: initial value for the parameter. This value can be a mathematical expression (see Section 10) which is called an *initial assignment*.
- Units: the units for the parameter value (default=none).
- Constant: Boolean indicating if the parameter value is constant (default=false).
- Port Type: used to indicate how this object can be connected in hierarchical models. The *input* type is used to indicate an object that is produced outside this model, the *internal* type is used to indicate that an object is produced inside this model but cannot be used in other models, and the *output* type is used to indicate that an object is produced by this model and can be used in other models (default=internal).

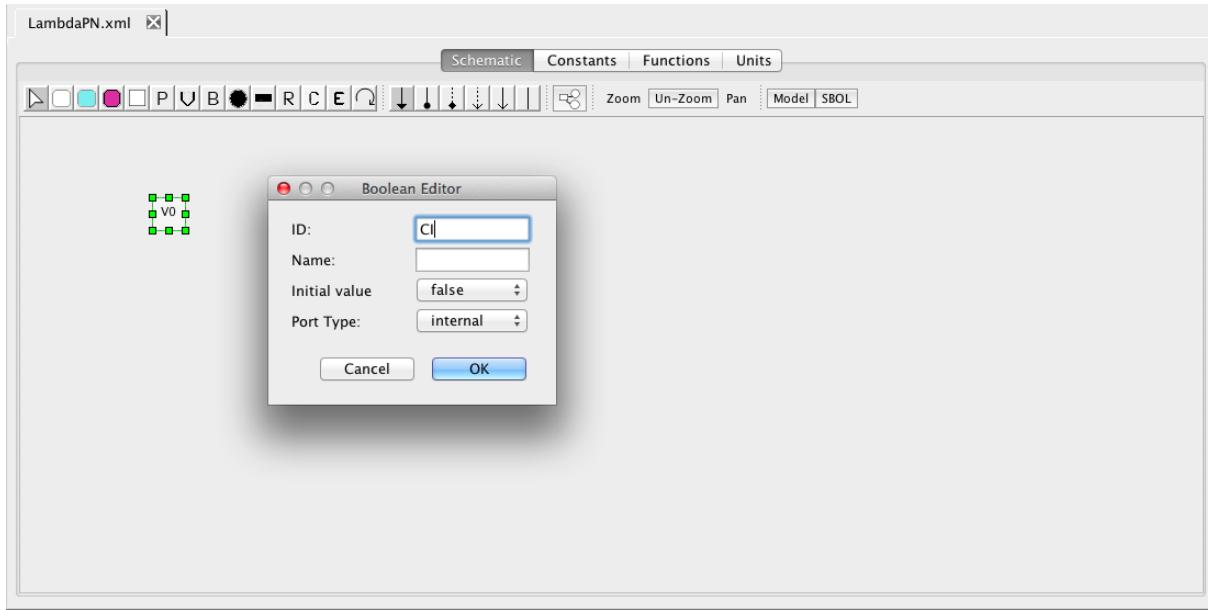
Note that changing the Constant attribute to true removes the parameter from the schematic, and it puts it in the list of Constants found in the Constant tab (see Section 3.15).



3.7 Boolean Variables

Boolean variables are global parameters that can only take the values of true and false. A Boolean variable includes the following:

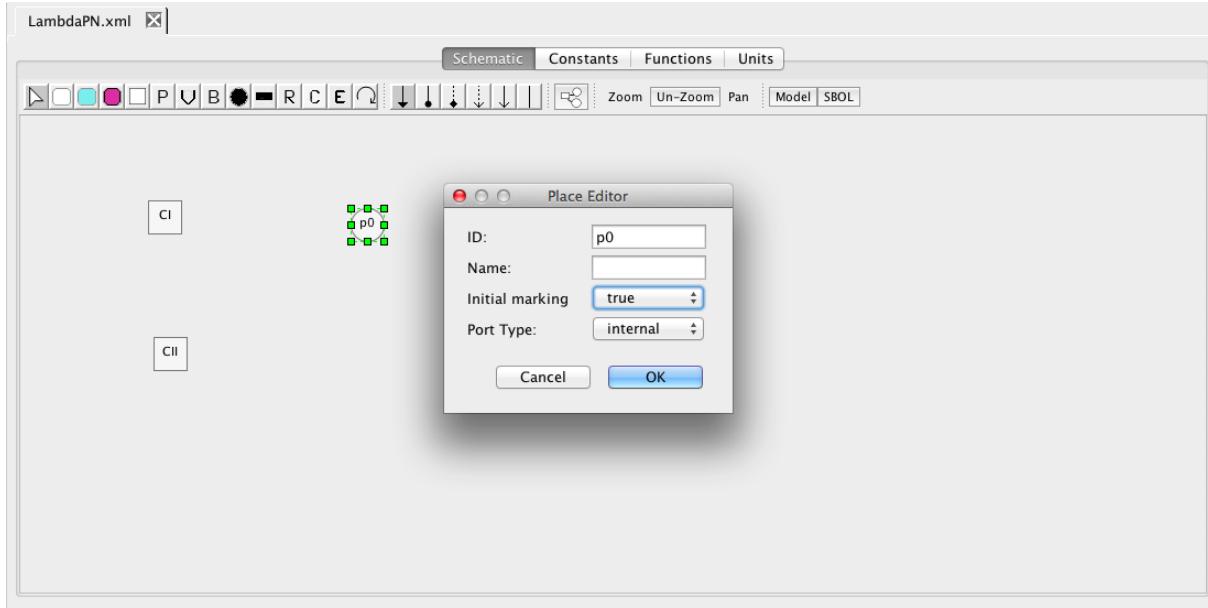
- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Initial value: initial value for the parameter (default=false).
- Port Type: used to indicate how this object can be connected in hierarchical models. The *input* type is used to indicate a object that is produced outside this model, the *internal* type is used to indicate that a object is produced inside this model but cannot be used in other models, and the *output* type is used to indicate that a object is produced by this model and can be used in other models (default=internal).



3.8 Petri Net Places

Places are a special type of global parameter that is used in Petri net models. A place can either be *marked* with a token or *not marked* with a token. A place includes the following:

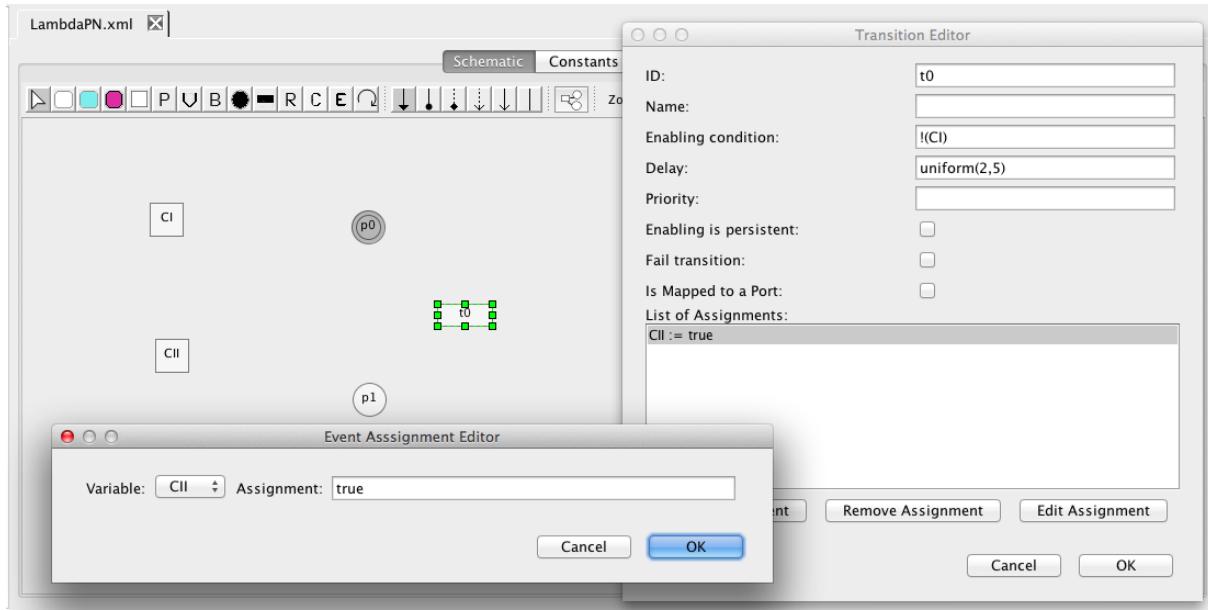
- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Initial marking: indicates if there is a token in this place initially (default=false).
- Port Type: used to indicate how this object can be connected in hierarchical models. The *input* type is used to indicate an object that is produced outside this model, the *internal* type is used to indicate that an object is produced inside this model but cannot be used in other models, and the *output* type is used to indicate that an object is produced by this model and can be used in other models (default=internal).



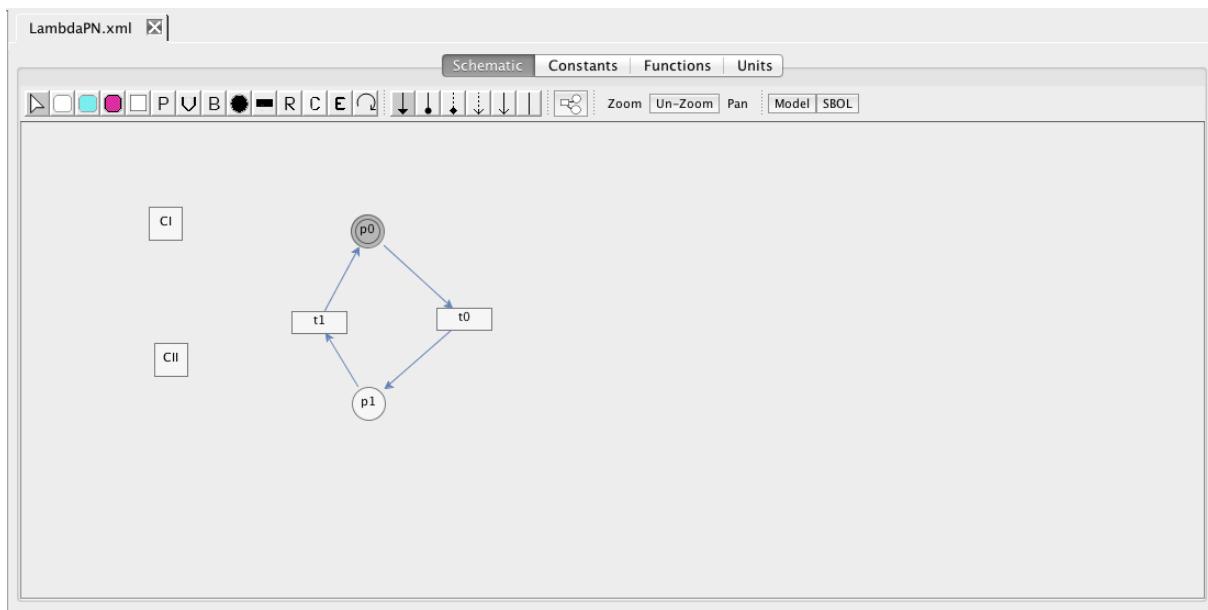
3.9 Petri Net Transitions

Transitions are a special kind of event used Petri net models. Each transition is composed of the following items:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Enabling condition: a mathematical formula which must evaluate to true for the transition to fire.
- Delay: a mathematical formula which is evaluated when a transition becomes enabled, and the result is the amount of time before the transition is fired.
- Priority: a mathematical formula which sets the priority for this transition when multiple transitions are scheduled to be executed simultaneously. In this situation, the transition with the highest priority is the next to be executed.
- Enabling is persistent: this Boolean indicates the behavior when an enabling condition becomes false before a transition is fired. If it is persistent, then it still fires when ready. If it is not persistent, then the transition is disabled and no longer scheduled to fire (default=false).
- Fail transition: this Boolean indicates if the firing of this transition should be considered as a failure or not (default=false).
- Is Mapped to a Port: this Boolean indicates whether this transition should be mapped to a port, so it can be replaced or deleted in hierarchical models.
- List of Assignments: these indicate how the state is supposed to change when the transition fires. Use the add, remove, and edit assignment buttons to update them. In the assignment editor, the variable to change must be selected and the assignment to be performed must be specified as a mathematical formula.



Places and transitions are connected to form a *token flow relation*. To connect a place to the *preset* of a transition, select the place and, while holding the mouse button, drag the connection to the transition. Similarly, to connect a place to the *postset* of a transition, select a transition and, while holding the mouse button, drag the connection to the place. A transition is enabled when all of its preset places are marked and its enabling condition is true. Once it is enabled, it can fire after the amount of time specified in the delay assignment has elapsed. When multiple transitions can occur at the same time, the priority expression is evaluated. The transition with the highest priority gets to fire first. When a transition fires, it removes tokens from all preset places, and places tokens in all postset places. Similarly, it will execute the assignments associated with the transition.



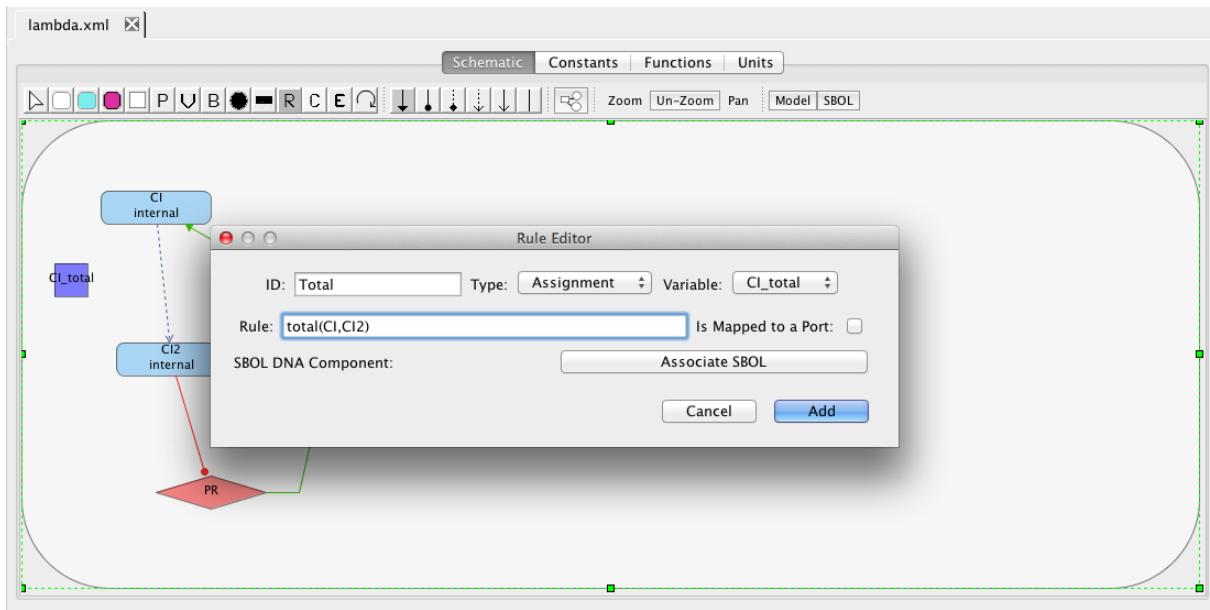
3.10 Rules

There are three types of rules: algebraic, assignment, and rate rules:

| | | |
|------------|------------------------------------|------------------------|
| Algebraic | left-hand side is zero | $0 = f(W)$ |
| Assignment | left-hand side is a scalar | $x = f(W)$ |
| Rate | left-hand side is a rate-of-change | $\frac{dx}{dt} = f(W)$ |

Algebraic rules specify relationships which must be maintained. Assignment rules specify the value of a compartment size, species amount or concentration, or parameter in terms of a mathematical formula (see Section 10). A quantity cannot be determined by both an assignment rule and an initial assignment (i.e., its initial value is an expression). Rate rules specify the rate of change of a compartment size, species amount or concentration, or parameter in terms of a mathematical formula. A quantity cannot be determined by both an assignment rule and a rate rule. A species that is a reactant or a product of any reaction cannot be updated by either an assignment rule or rate rule. When adding a rule, the user first selects the type of rule as shown below (note that when editing a rule, the user cannot modify the rule type). This automatically restricts the set of quantities available for the left-hand side to those that are valid. The user should then select a variable (except in the case of an algebraic rule) and enter a mathematical formula for the rule. In addition, the following attributes can be edited for a rule:

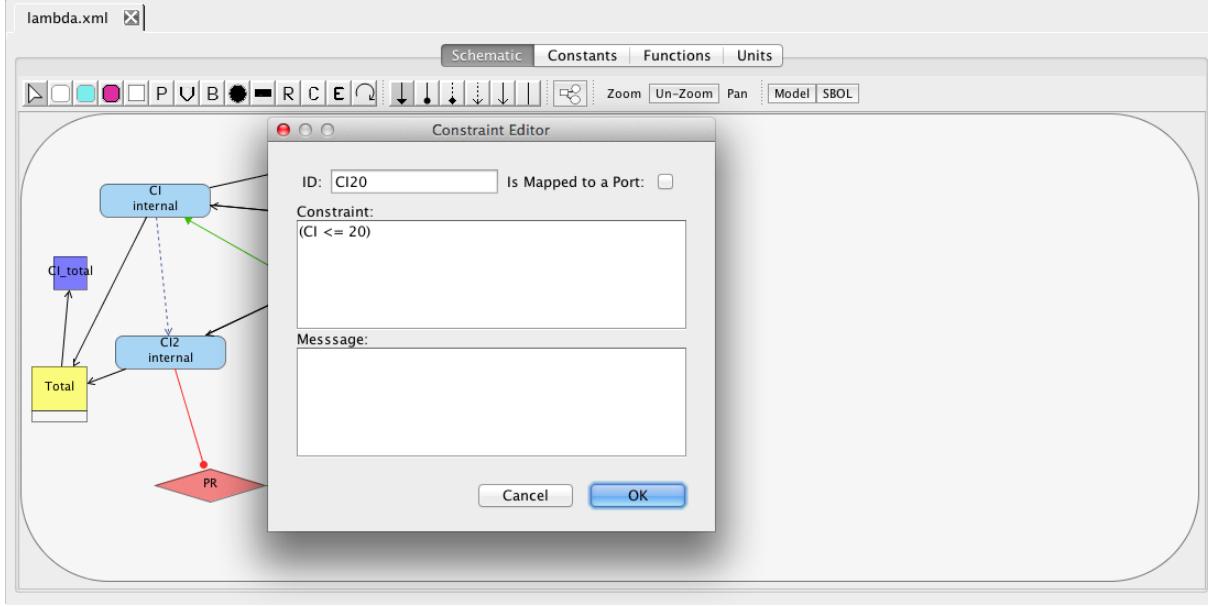
- ID: a unique ID composed only of alphanumeric characters and underscores.
- Is Mapped to a Port: this Boolean indicates whether this rule should be mapped to a port, so it can be replaced or deleted in hierarchical models.
- SBOL DNA Component: the Associate SBOL button allows the user to annotate the promoter with SBOL (see Section 4).



3.11 Constraints

Constraints are used to specify properties that should cause simulation to terminate. As shown below, each constraint is composed of an ID which is used to identify it in graphs, a constraint given

as a mathematical formula (see Section 10), and a message describing the constraint. A default ID is automatically generated when a new constraint is created. Special functions are provided to support a form of *Continuous Stochastic Logic* (CSL) within these constraints (see Section 10).



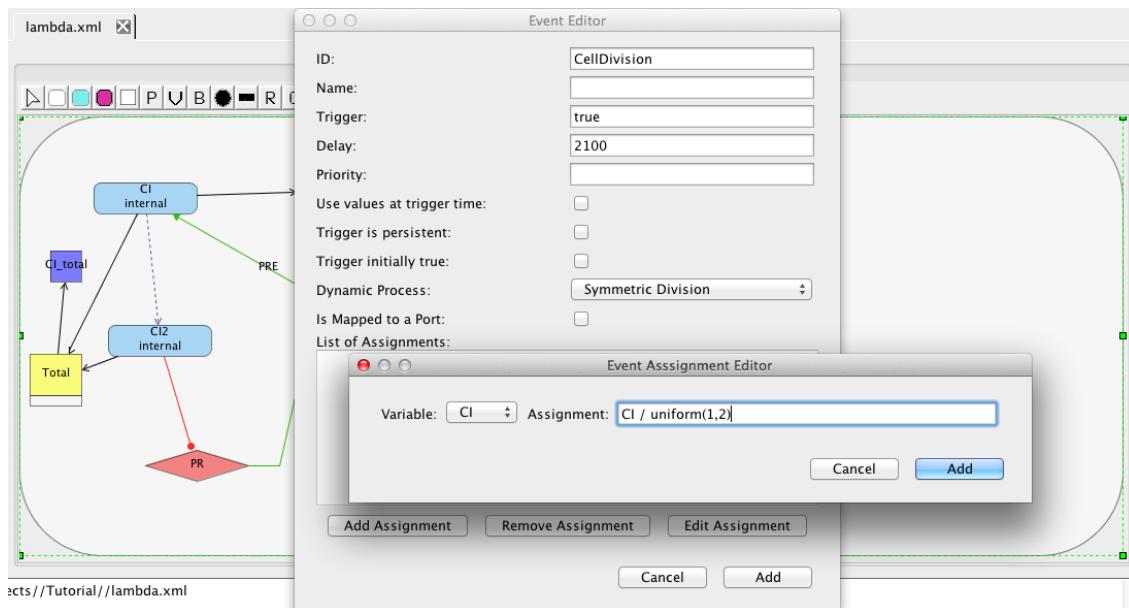
Our analysis methods can provide histograms that show the proportion of simulations that are terminated due to a constraint. Analysis can also show time-series plots showing the proportion of simulations at each time point that have terminated due to a constraint.

3.12 Events

Events are used to specify discrete changes of compartment sizes, species amounts or concentrations, and parameter values. Each event is composed of the following items:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Trigger: a mathematical formula which, when it changes evaluation from false to true, indicates that the event is triggered.
- Delay: a mathematical formula which is evaluated when an event is triggered, and the result is the amount of time before the event is to be executed.
- Priority: a mathematical formula which sets the priority for this event when multiple events are scheduled to be executed simultaneously. In this situation, the event with the highest priority is the next to be executed.
- Use values at trigger time: this Boolean indicates if values for the event assignments should be calculated at trigger time or execution time (default=false).
- Trigger is persistent: this Boolean indicates the behavior when a trigger expression becomes false before an event is executed. If it is persistent, then it still executes when ready. If it is not persistent, then the event is disabled and no longer scheduled to execute (default=false).

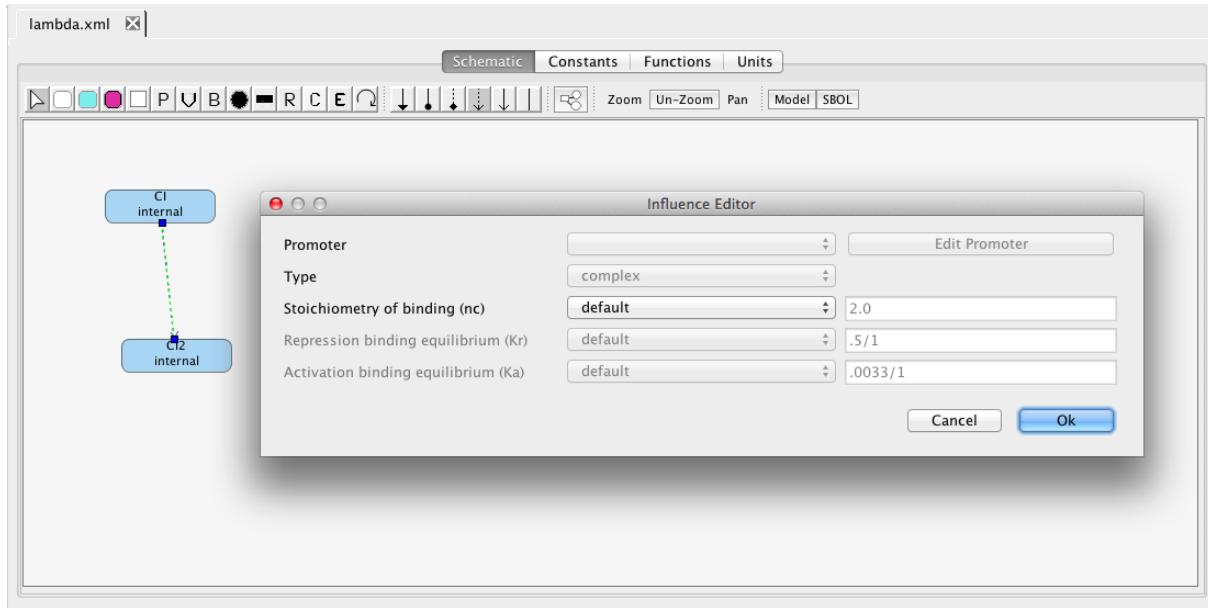
- Trigger is initially true: this Boolean indicates the behavior when a trigger evaluates to true at time 0. If it is initially true, there is no change in evaluation, so the event is not triggered. If it is not initially true, there is now a change in evaluation, so the event is triggered.
- Dynamic Process: indicates if this is a dynamic event. The possible dynamic even types are symmetric and asymmetric cell division, cell death, and cell movement in a specified direction.
- Is Mapped to a Port: this Boolean indicates whether this rule should be mapped to a port, so it can be replaced or deleted in hierarchical models.
- List of Event Assignments: these indicate how the state is supposed to change when the event executes. Use the add, remove, and edit assignment buttons to update them. In the event assignment editor, the variable to change must be selected and the assignment to be performed must be specified as a mathematical formula.



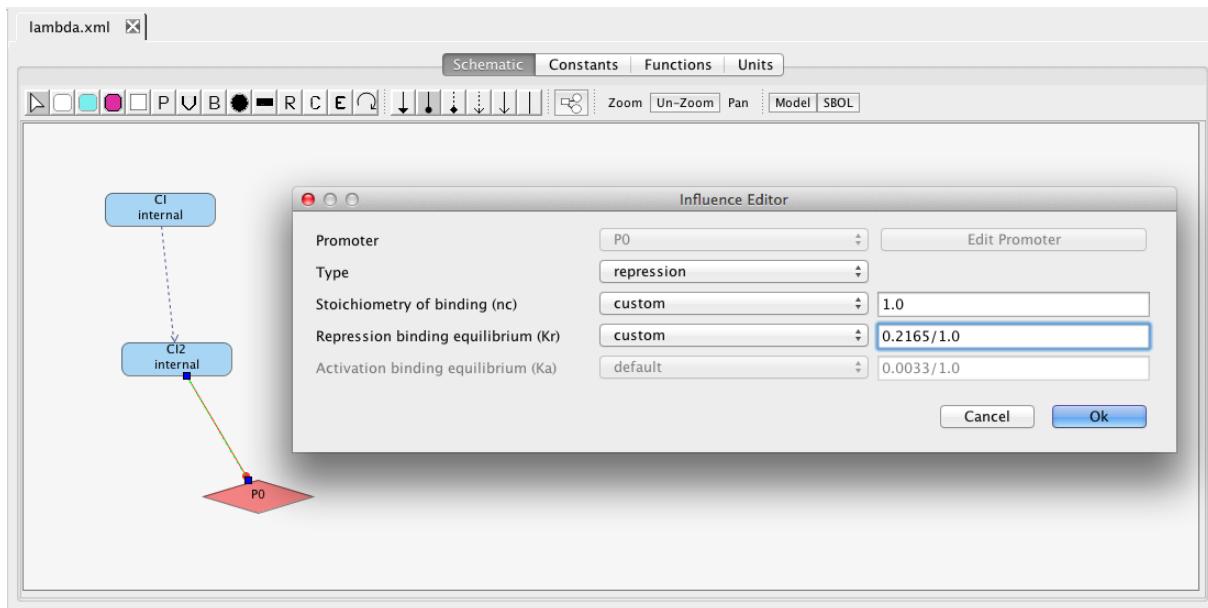
3.13 Influences

There are several types of influences between species that can be specified. A species may *activate* or *repress* the production of other species. This relationship is made from a species to the promoter that it activates or represses. It can also be made directly between two species with an implicit promoter annotating the edge. One can also specify when there is no influence between two species. An activating or repressing self-influence can also be specified. Species can also be related by complex-formation reactions, which indicate that multiple identical or different species can be combined to form a complex species.

To add a complex-formation reaction, select the complex formation icon , highlight a species that is part of a complex, and, while holding the mouse button, stretch the complex formation arc to the complex species. If you double-click on the complex formation arc, an influence editor will open, which indicates that this is a complex formation arc and the stoichiometry of binding (i.e., the number of molecules of the source species used to construct the sink species) is 2, which you can customize, if desired.

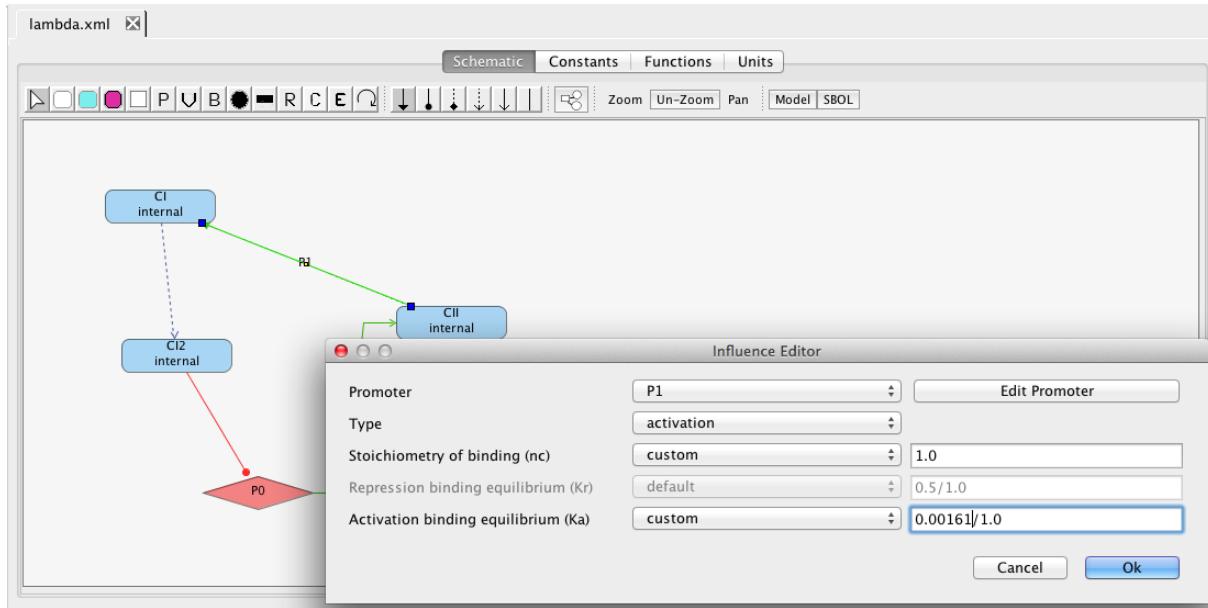


To create a repression influence, select the Repression icon , highlight the species that is acting as a repressor, and, while holding the mouse button, stretch the repression arc to either the promoter or another species. In the latter case, an implicit promoter is added. Double clicking on the repression arc brings up an influence editor. In this editor, you can customize the stoichiometry of binding as well as the repression binding equilibrium constant. You can also change the type of influence, as well as, the promoter on which this influence takes place. Note that pressing the Edit Promoter button brings up the Promoter Editor for the corresponding promoter (see Section 3.5).

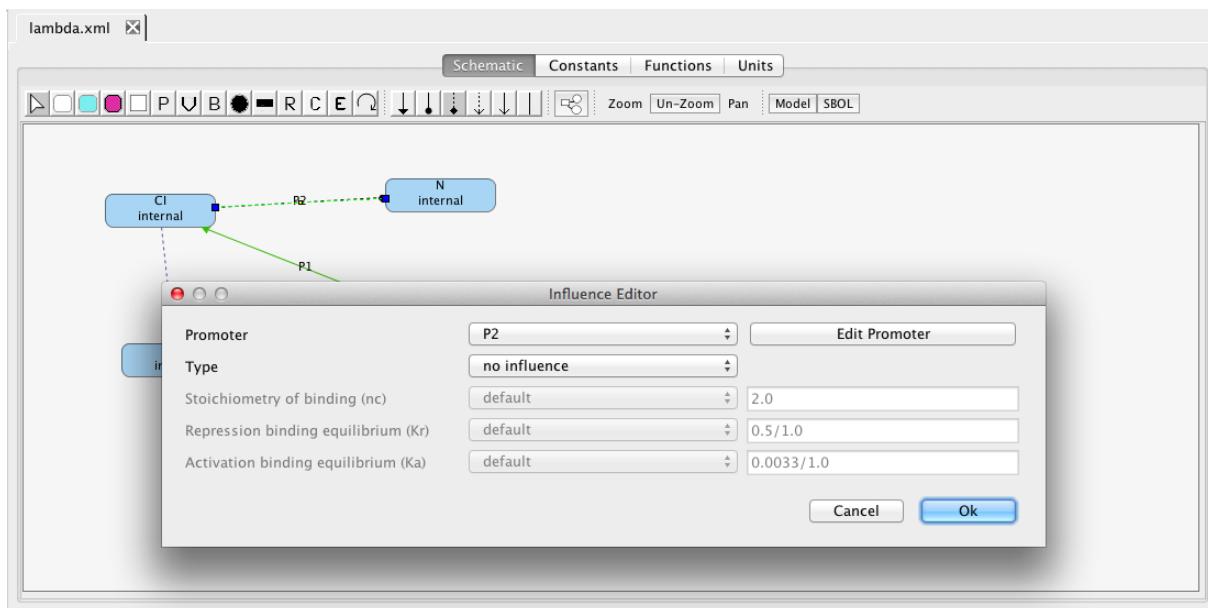


To create an activation influence, select the Activation icon , highlight the species that is acting as an activator, and while holding the mouse button stretch the activation arc to either the promoter or another species. In the latter case, an implicit promoter is added. Double-clicking on the activation arc brings up an influence editor. In this editor, you can customize the stoichiometry

of binding as well as the activation binding equilibrium constant. You can also change the type of influence, as well as, the promoter on which this influence takes place.

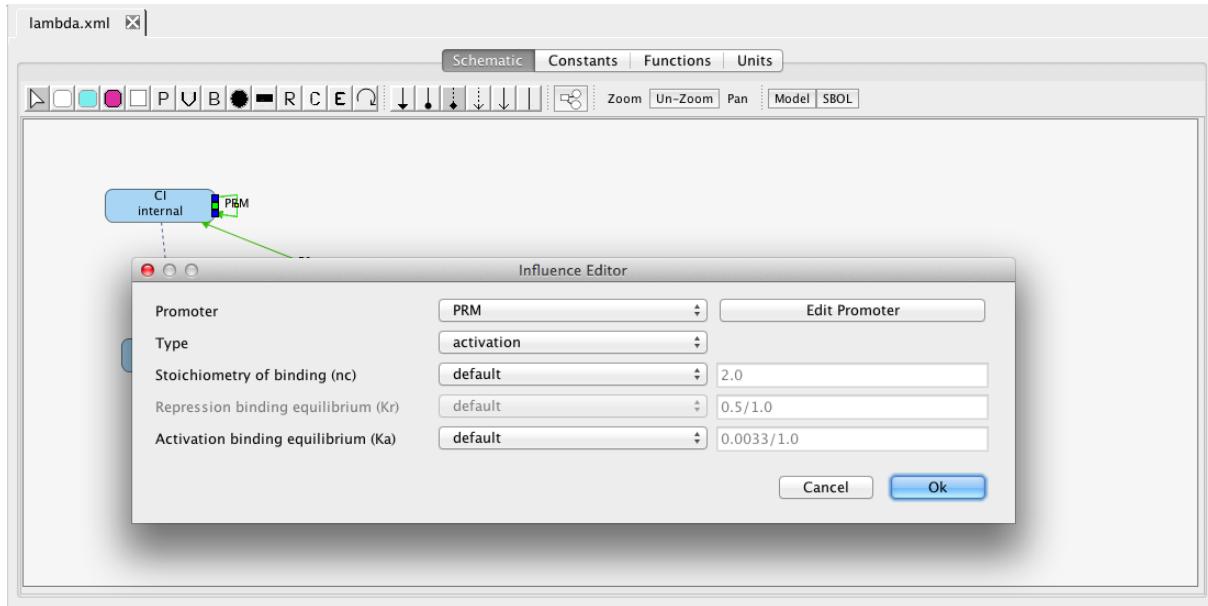


To indicate that a species has no influence on another species, select the No Influence icon , highlight the species that has no influence, and, while holding the mouse button, stretch the no-influence arc to the species it has no influence upon. Double-clicking on the no-influence arc brings up an influence editor. In this editor, you can change its type to indicate that it actually has an influence, if you like, and edit accordingly. No-influence arcs are used by the Learn Tool to avoid adding influences where you are certain there isn't one. You can also change the type of influence, as well as, the promoter on which this species has no influence.



The last type of influence is a self-influence. A self-influence can be either activation or repression. To create a self-influence, select the Self-influence icon , select the appropriate Activation

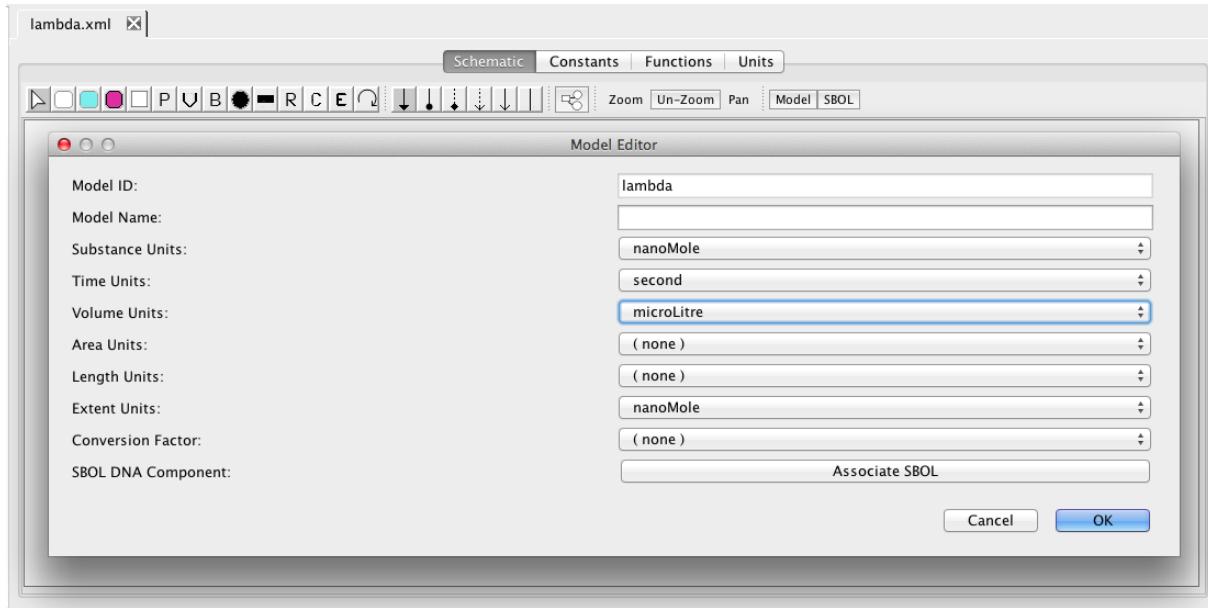
or Repression icon, and click on the species that influences itself. If you click on a self-influence, you will be able to edit its promoter, its type of influence, its stoichiometry of binding, as well as the appropriate binding equilibrium constant.



3.14 Model Editor

Clicking on the Model button on the schematic tab brings up the Model Editor. This editor includes:

- Model ID: a unique ID composed only of alphanumeric characters and underscores. This cannot be changed here. To change it, you must rename the model file.
- Name: an arbitrary string description (optional).
- Substance units: the default units for species.
- Time units: the default units for time.
- Volume units: the default units for 3-dimensional compartments.
- Area units: the default units for 2-dimensional compartments.
- Length units: the default units for 1-dimensional compartments.
- Extent units: the default units for the amount changed due to a reaction. Kinetic laws have default units of extent units / time units.
- Conversion factor: the constant global parameter to be used as the default conversion factor to convert species units into extent units.
- SBOL DNA Component: the Associate SBOL button allows the user to annotate the promoter with SBOL (see Section 4).

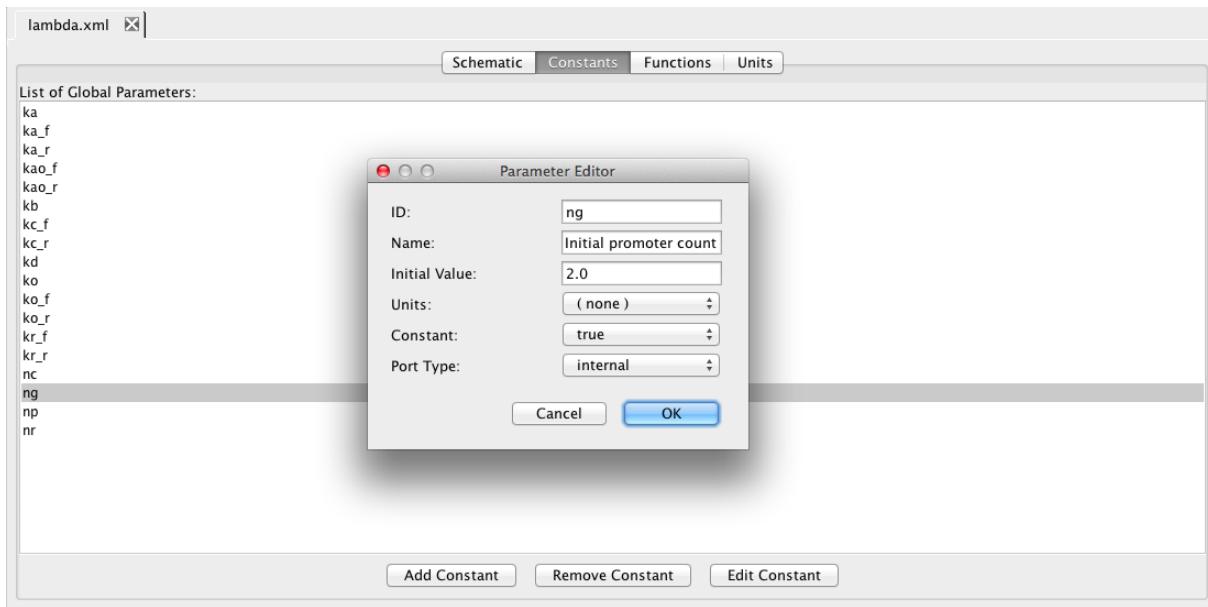


3.15 Constants

The Constants tab includes a list of all the constant global parameters. They are not included in the schematic, but they can appear in math formulas (see Section 10). This tab can be used to add, remove, or edit these constants. The following attributes can be edited for a global parameter:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Initial value: initial value for the parameter. This value can be a mathematical expression (see Section 10) which is called an *initial assignment*.
- Units: the units for the parameter value (default=none).
- Constant: Boolean indicating if the parameter value is constant (default=true).
- Port Type: used to indicate how this object can be connected in hierarchical models. The *input* type is used to indicate a object that is produced outside this model, the *internal* type is used to indicate that a object is produced inside this model but cannot be used in other models, and the *output* type is used to indicate that a object is produced by this model and can be used in other models (default=internal).

Note that changing the Constant attribute to false removes the parameter from the constants list, and it places the parameter on the schematic.



Some constants are added to the model automatically because they are needed to provide default parameter values when converting genetic regulation elements into a reaction-based model. In the Species, Promoter, and Influence Editors, any parameter set to default uses the value found in this list. These defaults can be modified in the user preferences (see Section 9). The user can also edit these parameters for an individual model here. The complete list of model generation parameters is listed below:

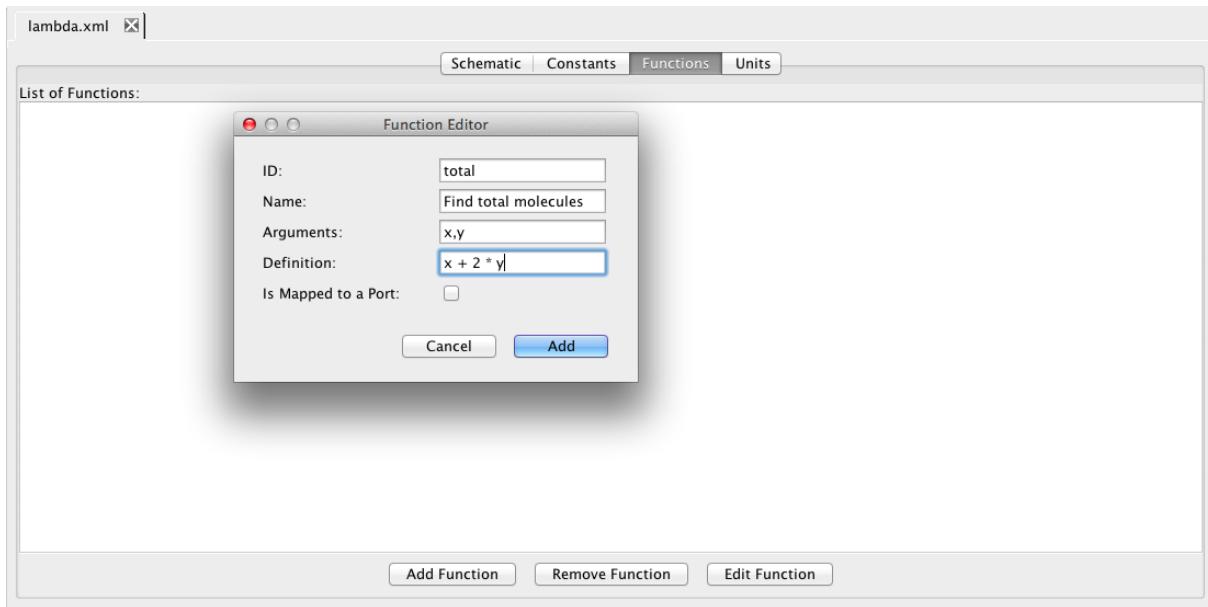
| ID | Default Value | Units | Structure | Description |
|----------|---------------|--------------------------------------|-----------|------------------------------------|
| ka | 0.25 | $\frac{1}{\text{sec}}$ | promoter | Activated production rate |
| Ka | 0.0033 | $\frac{1}{\text{molecule}^{(nc+1)}}$ | influence | Activation binding equilibrium |
| kb | 0.0001 | $\frac{1}{\text{sec}}$ | promoter | Basal production rate |
| kd | 0.0075 | $\frac{1}{\text{sec}}$ | species | Degradation rate |
| kecd | 0.005 | $\frac{1}{\text{sec}}$ | species | Extracellular degradation rate |
| nc | 2 | molecule | influence | Stoichiometry of binding |
| nr | 30 | molecule | model | Initial RNAP count |
| ng | 2 | molecule | promoter | Initial promoter count |
| ko | 0.05 | $\frac{1}{\text{sec}}$ | promoter | Open complex production rate |
| Ko | 0.033 | $\frac{1}{\text{molecule}}$ | promoter | RNAP binding equilibrium |
| Kao | 1 | $\frac{1}{\text{molecule}}$ | promoter | Activated RNAP binding equilibrium |
| Kr | 0.5 | $\frac{1}{\text{molecule}^{nc}}$ | influence | Repression binding equilibrium |
| np | 10 | molecule | promoter | Stoichiometry of production |
| Kc | 0.05 | $\frac{1}{\text{molecule}}$ | species | Complex formation equilibrium |
| kmdiff_f | 1.0 | $\frac{1}{\text{molecule}}$ | species | Forward membrane diffusion rate |
| kmdiff_r | 0.01 | $\frac{1}{\text{molecule}}$ | species | Reverse membrane diffusion rate |
| kecdiff | 1.0 | $\frac{1}{\text{molecule}}$ | species | Extracellular diffusion rate |

3.16 Functions

The functions tab is used to add, remove, and edit function definitions. Function definitions are used to create user-defined functions that can then be used in math formulas (see Section 10). As shown below, function definitions include:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Arguments: a comma-separated list of arguments for the function.
- Definition: a math formula, though it is restricted to the use of variable names which are arguments to the function.
- Is Mapped to a Port: this Boolean indicates whether this transition should be mapped to a port, so it can be replaced or deleted in hierarchical models.

While functions can call other functions, they cannot be recursive (i.e., call themselves) either directly or indirectly (i.e., through a cycle of function calls).



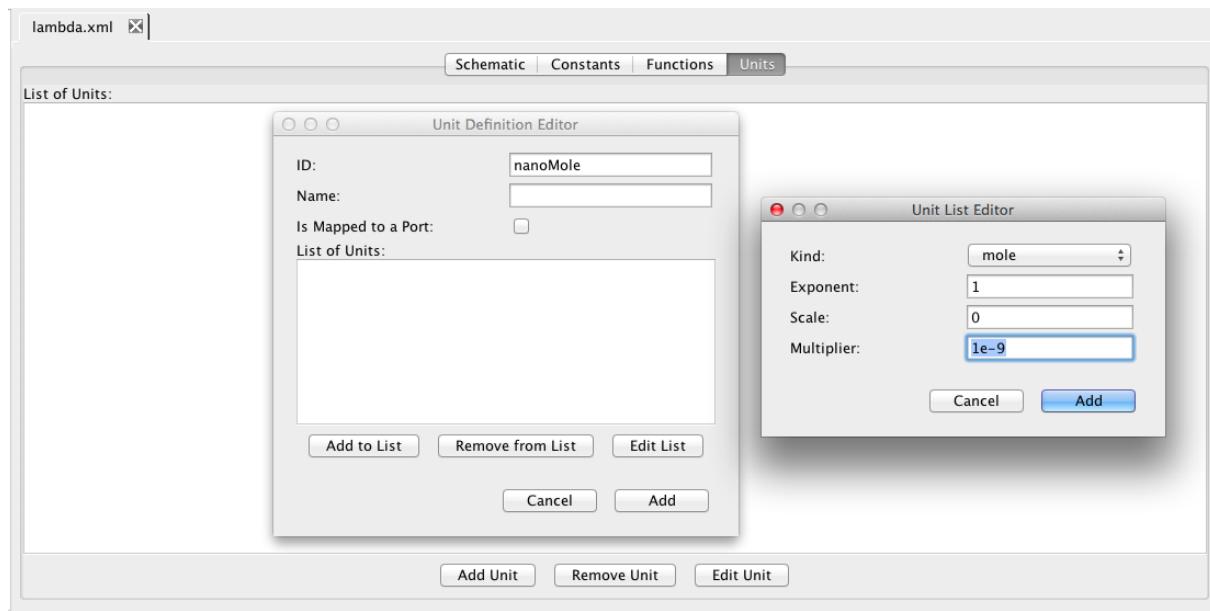
3.17 Units

The Units tab is used to add, remove, and edit unit definitions. Unit definitions are used to construct user-defined units which are derived from the set of base units. As shown below, a unit definition includes:

- ID: a unique ID composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- Is Mapped to a Port: this Boolean indicates whether this transition should be mapped to a port, so it can be replaced or deleted in hierarchical models.
- List of Units: the units that make up the unit definition. There are buttons to add, remove, and edit elements in the list of units. Each unit is composed of a kind, exponent, scale, and multiplier. The kind is selected from the list of base units in the table below. The exponent and multiplier are real numbers, and the scale is an integer that specifies the relationship between the derived unit and the base unit using the relation below:

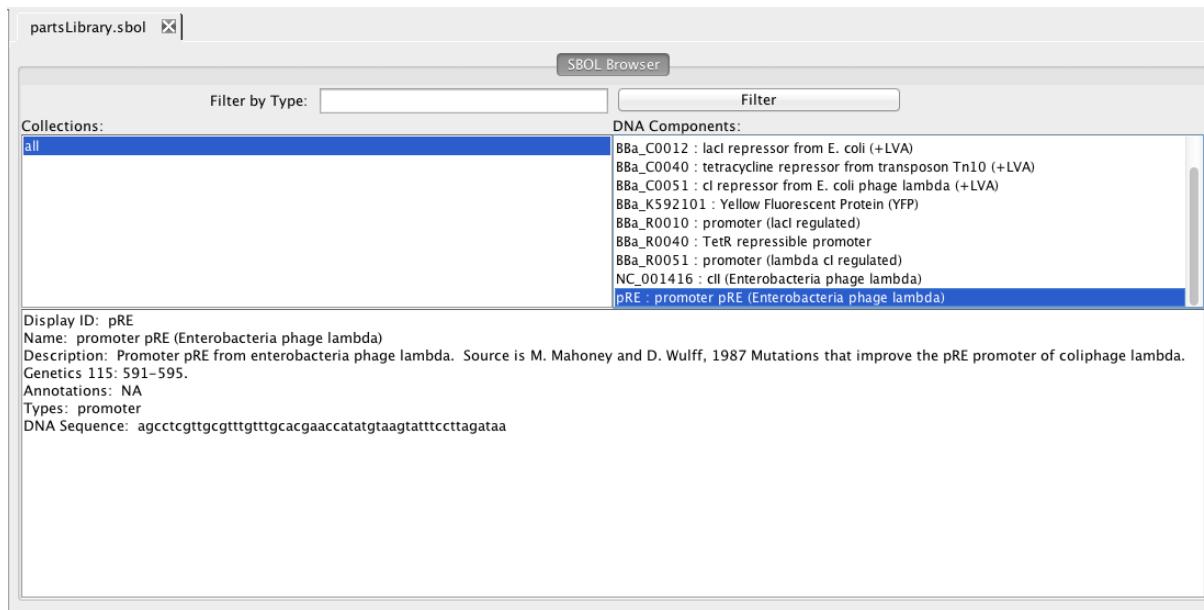
$$\text{unit} = (\text{multiplier} * 10^{\text{scale}} * \text{baseUnit})^{\text{exponent}}$$

| | | | | | |
|---------------|-------|----------|--------|-----------|-------|
| ampere | farad | joule | lux | radian | volt |
| avogadro | gram | katal | metre | second | watt |
| bacquerel | gray | kelvin | mole | siemens | weber |
| candela | henry | kilogram | newton | sievert | |
| coulomb | hertz | litre | ohm | steradian | |
| dimensionless | item | lumen | pascal | tesla | |

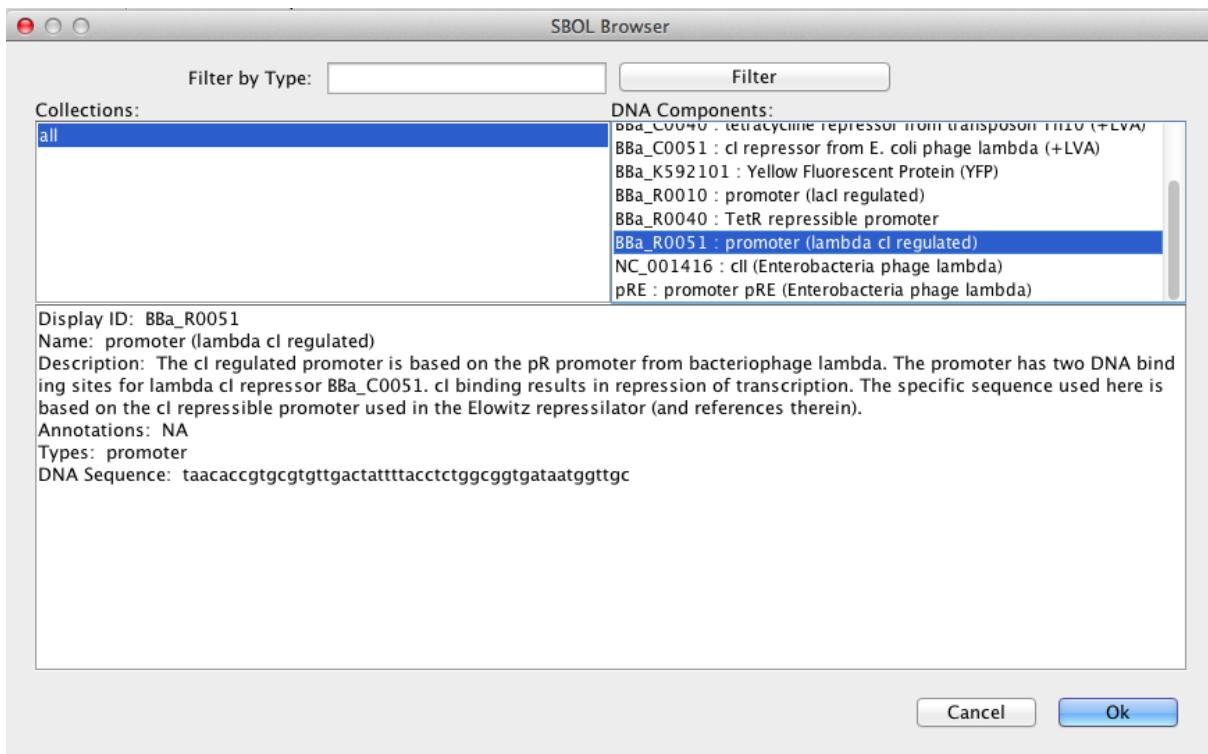
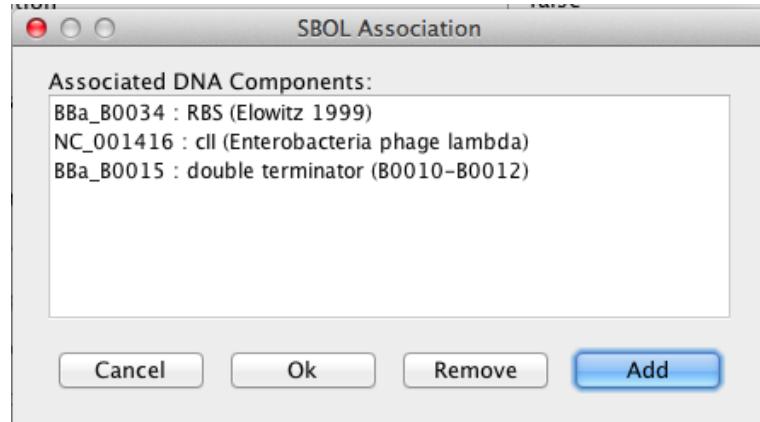


4 SBOL Editors

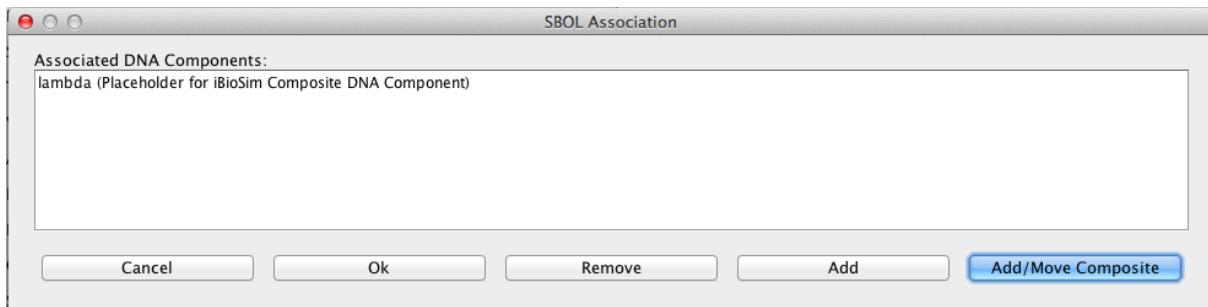
SBOL is an emerging standard under development for the exchange of information about synthetic biology designs. An SBOL file includes *Collections of DNA Components*. Each DNA component has a unique ID, name, description, *Sequence Ontology* type, and a DNA sequence. A DNA component can be a simple sequence feature such as a *promoter*, *ribosome entry site*, *coding sequence* (i.e., gene), or *terminator*. It can also be a DNA sequence annotated with several of these individual features. For example, a DNA component that is a gene may have its sequence annotated with a promoter DNA component followed by a ribosome binding site, an open reading frame, and a terminator. iBioSim includes an SBOL browser that allows the user to view an individual SBOL file by clicking on it in the list of project files. In this browser, the user can highlight individual DNA components and see the information stored about them. The user can also enter an SO type and press the Filter button to limit the list to only DNA components of the requested type.



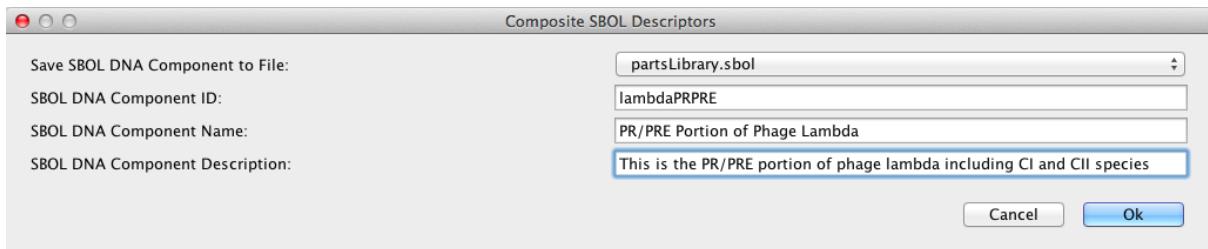
When the user clicks on an Associate SBOL button in the editor for a model element, it opens up a window that includes an ordered list of DNA components associated with that model element. In this window, highlighting a DNA component and pressing the Remove button removes this component from the list. Pressing the Add button opens an SBOL Browser that can be used to find a DNA component to add to this list. By selecting a DNA component and pressing OK, the selected component is added to the list. It is added to the end of the list when no component is highlighted in the list, and it is added before the highlighted component, otherwise.



The SBOL Association window is a bit different when associating SBOL to the top-level SBML model itself. In particular, it includes a button to Add/Move Composite. Upon save, any model which has SBOL associated with some of its elements automatically generates a composite DNA component which is typically associated with the model itself. It is possible though to add additional DNA components either before or after this composite component. One can also remove the composite DNA component and replace it with a DNA component from an SBOL file. The purpose of the Add/Move Composite is to restore the composite component to the list, if it has been removed. It has the added functionality of moving it before the highlighted component, if it is currently in the list.



The last SBOL window is the Composite SBOL Descriptors window which is opened by pressing the SBOL button on the schematic. In this window, the user can select which SBOL file in which to store the composite component generated for this model. The user can also specify the Id, name, and description for this composite component. When the user saves this model, this information is stored with the composite component in the specified SBOL file.



5 Analysis Tool

The analysis tool is used to analyze biochemical reaction network models. `iBioSim` comes with a number of simulation methods, ranging from continuous-deterministic (ODE) simulation methods to discrete-stochastic (Monte Carlo) simulation methods. In order to perform efficient temporal behavior analysis, various model abstractions can also be automatically applied. Many of the analysis and abstraction routines are implemented within the `reb2sac` tool described in Kuwahara's PhD Dissertation (UofUtah 2007). An analysis view includes several tabs. The Simulation Options tab, described in Section 5.1, selects the different simulation methods and parameters. The Abstraction Options tab described in Section 5.2 configures model abstraction. The Schematic tab, described in Section 5.3, allows the user to modify or sweep various parameters on the schematic. It also provides an alternative way of visualizing simulation data upon the schematic. The Parameters tab allows the user to modify or sweep global parameters, and it is described in Section 5.4. The SBML elements tab described in Section 5.5 allows the user to add or remove parts of the model during analysis. The TSD Graph tab is a graph editor for time series data, and it is described in Section 7. Finally, the Histogram tab is a graph editor for probability data, and it is described in Section 8.

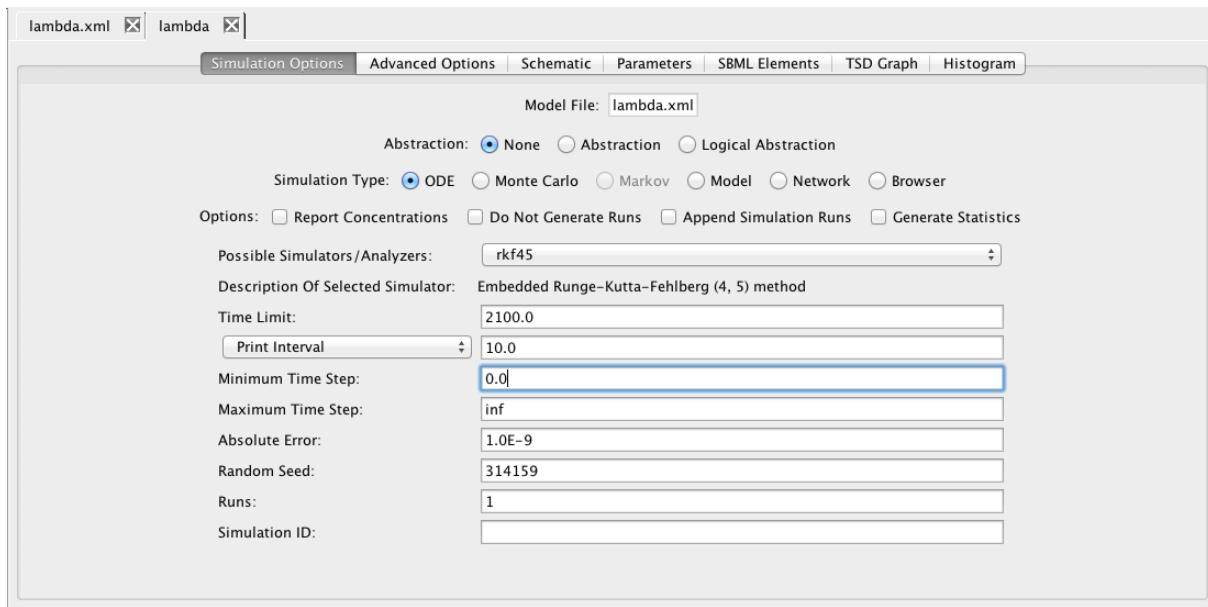
5.1 Simulation Options

The first set of radio buttons in this tab specifies the type of abstraction.

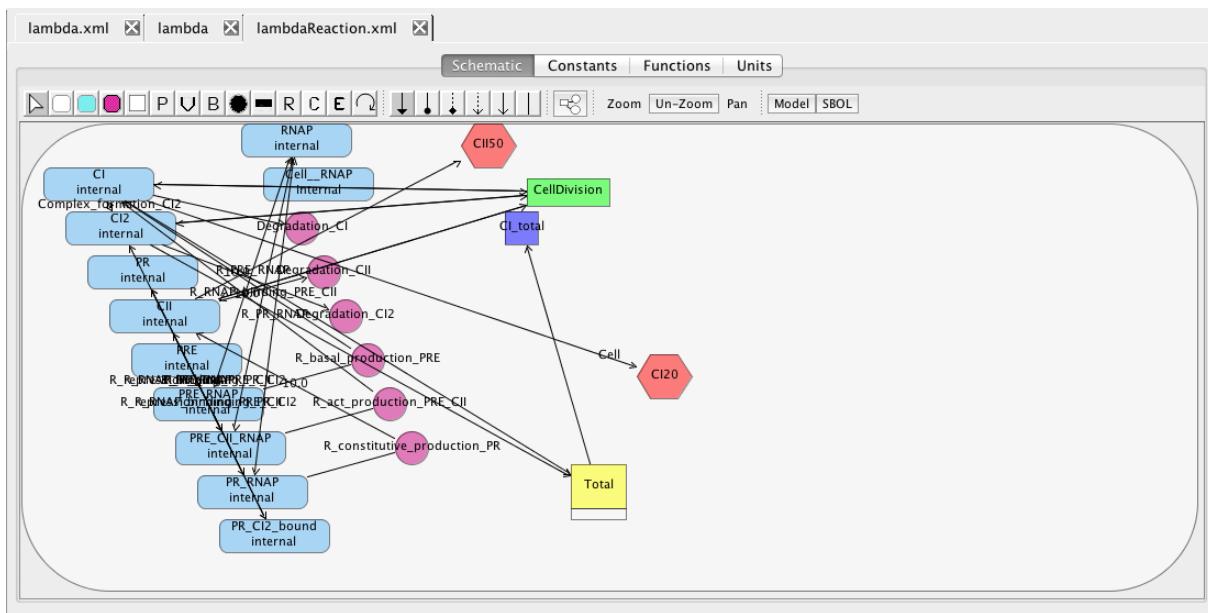
- None: use no abstraction.
- Abstraction: perform reaction-based abstraction.
- Logical Abstraction: perform both reaction-based and logical abstractions.

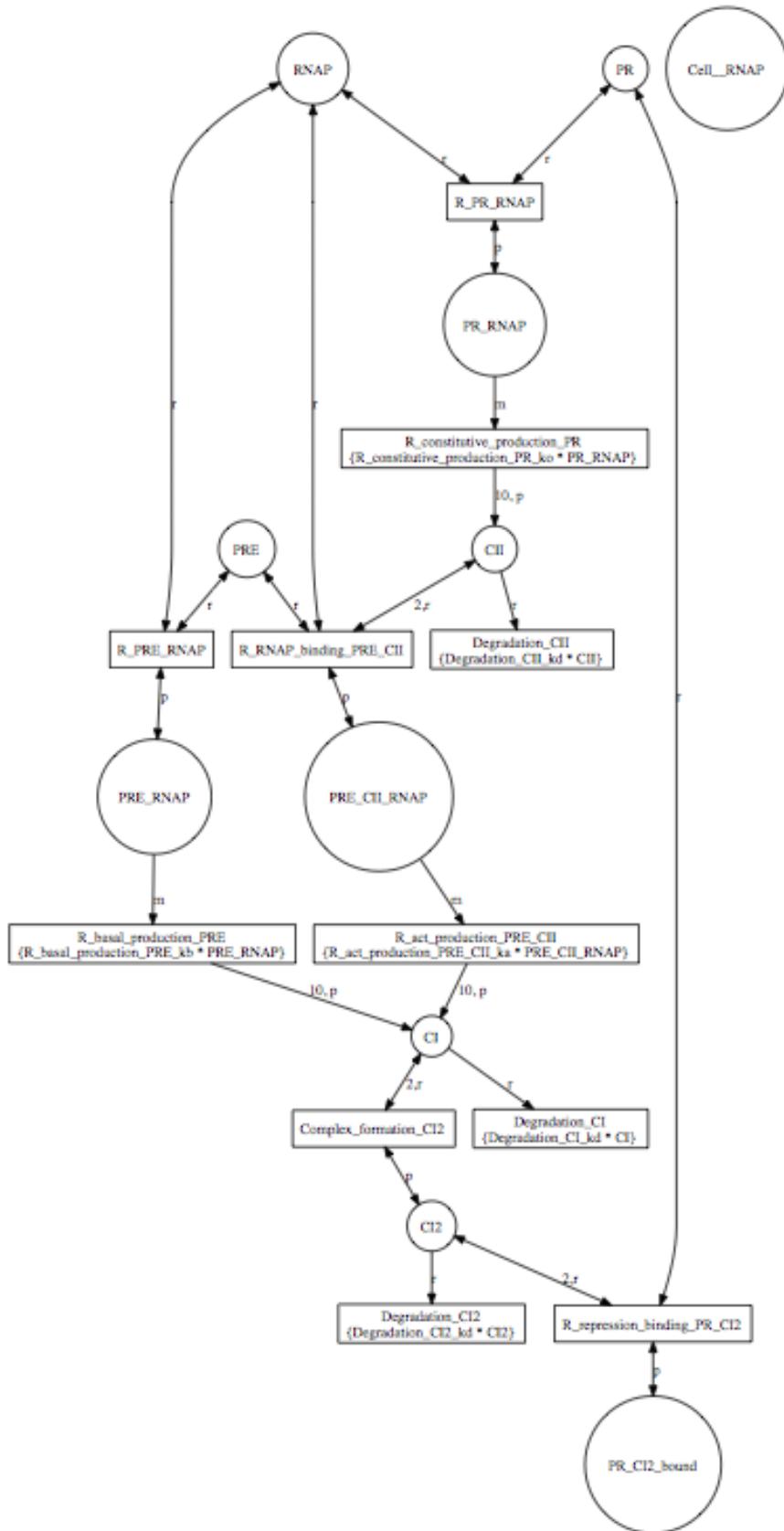
The second set of radio buttons specify the type of analysis.

- ODE: perform continuous-deterministic simulation.
- Monte Carlo: perform discrete-stochastic simulation.
- Markov: perform temporal probability distribution analysis on finite-state Markov chain models (this option requires logical abstraction).
- Model: produces a flattened model and puts it in the list of project files.
- Network: outputs the model in the GraphViz format for display by dotty.
- Browser: outputs the model in xhtml format for display in a web browser.



Some results selecting Model, Network, and Browser options are shown below.





| | |
|-----------------------|-------------------------|
| Model ID | lambda |
| Model Name | Created from lambda.gcm |
| Model Substance Units | mole |
| Model Time Units | second |
| Model Volume Units | litre |
| Model Extent Units | nanoMole |

| Functions |
|--|
| $\text{uniform}(a, b) = \frac{a+b}{2}$ |
| $\text{total}(x, y) = x+2*y$ |

| Unit ID | Definition |
|-------------------------|--|
| u_1_mole_n1 | (mole) ⁻¹ |
| microLitre | 0.001•litre |
| nanoMole | 1e-09•mole |
| u_1_second_n1 | (second) ⁻¹ |
| u_1_mole_n1_1_second_n1 | (mole) ⁻¹ •(second) ⁻¹ |

| Compartment ID | Dimensions | Initial Size | Units | Constant |
|----------------|------------|--------------|-------|----------|
| Cell | 3 | 1 | none | True |

| Species ID | Compartment | Initial Value | Units | Conversion Factor | Boundary | Constant | HasOnlySubstanceUnits |
|------------|-------------|---------------|-------|-------------------|----------|----------|-----------------------|
| CI | Cell | 0 (amount) | none | none | False | False | True |
| CI2 | Cell | 0 (amount) | none | none | False | False | True |
| PR | Cell | 2 (amount) | none | none | False | False | True |
| CII | Cell | 0 (amount) | none | none | False | False | True |
| DDP | Cell | 2 (amount) | none | none | False | False | True |

The next group of options are for simulation-based analysis only.

- Report concentrations: if checked, report concentrations rather than amounts (default=Amounts).
- Do Not Generate Runs: if checked, simulation data is not produced, though summary probabilistic data is still produced (default=Generate Runs).
- Append Simulation Runs: if checked, additional simulation runs are added to the existing set of simulation runs rather than first removing the previous runs (default=Overwrite). Note that the seed changes automatically when this gets checked, because otherwise the additional runs will be the same as the previous set.
- Generate Statistics: if checked, statistics such as average and standard deviation data are generated immediately after completing simulation. Otherwise, they are only generated on demand (i.e., for example, when included in a TSD graph).

The next field specifies the simulation method you want to use based on the simulation type you specified. The methods available are:

| Type | Method ID | Description |
|-------------|----------------|--|
| ODE | Euler | The forward Euler Method |
| ODE | gear1 | Gear Method M=1 |
| ODE | gear2 | Gear Method M=2 |
| ODE | rk4imp | Implicit 4th order Runge-Kutta at Gaussian points |
| ODE | rk8pd | Embedded Runge-Kutta Prince-Dormand (8,9) method |
| ODE | rkf45 | Embedded Runge-Kutta-Fehlberg (4, 5) method |
| ODE | rkf45 (java) | Native Java Runge-Kutta-Fehlberg method |
| Monte Carlo | gillespie | Gillespie's SSA-direct method |
| Monte Carlo | SSA-CR | SSA composition and rejection method |
| Monte Carlo | iSSA | incremental SSA |
| Monte Carlo | interactive | Interactive SSA-direct method |
| Monte Carlo | emc-sim | Uses jump count as next reaction time |
| Monte Carlo | bunker | Uses mean for next reaction time |
| Monte Carlo | nmc | Uses normally distributed next reaction time |
| Markov | steady-state | Steady-state Markov chain analysis |
| Markov | transient | Transient Markov chain analysis |
| Markov | reachability | Only perform reachability analysis |
| Markov | atacs | Use atacs steady-state Markov analysis engine |
| Markov | ctmc-transient | Transient distribution analysis |

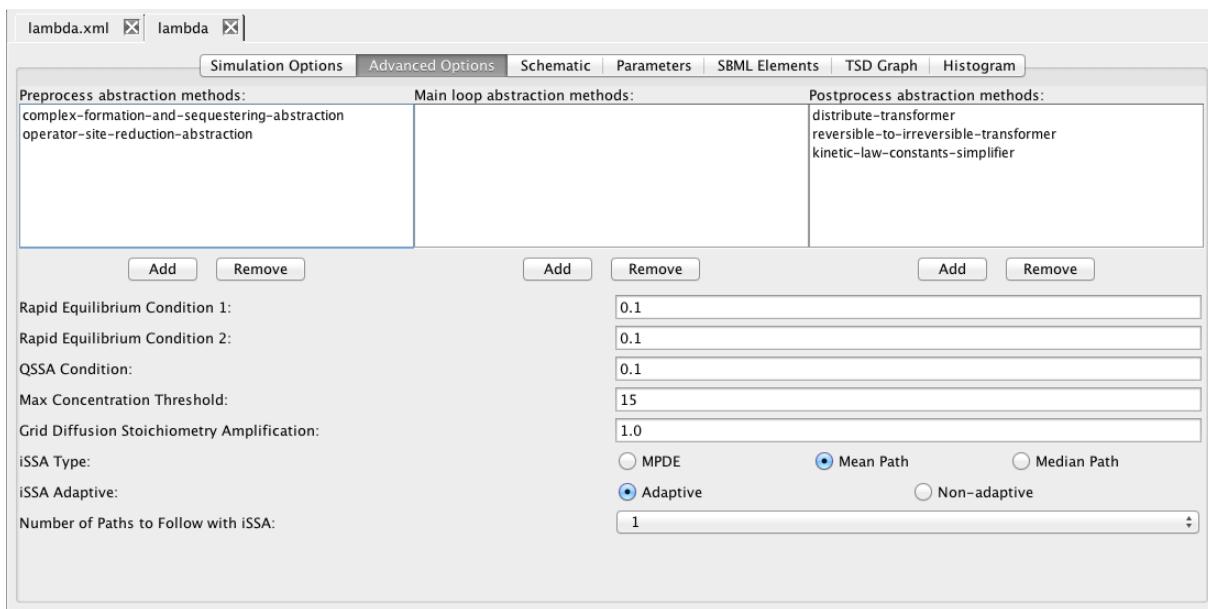
There are some properties that need to be set for simulation. The table below specifies these:

| Field | Description |
|------------------------|---|
| Time Limit | The simulation time limit |
| Choose one: | |
| Print Interval | The print time interval for each simulation run |
| Minimum Print Interval | Print on change but no more often than this interval |
| Number of Steps | Number of steps to print |
| Minimum Time Step | The smallest time step allowed |
| Maximum Time Step | The largest time step allowed (also the time step used for the Euler method) |
| Absolute Error | Used by the adaptive time step ODE methods |
| Random Seed | An integer number as a seed to generate random numbers |
| Runs | The number of random simulation runs to perform |
| Simulation ID | Creates a simulation directory with the ID name |

5.2 Abstraction Options

Abstraction is performed in three steps. There is a preprocess step, which is a sequence of abstractions performed once on the initial model. The main loop iterates through a group of abstractions until there is no change in the model. Finally, there is a postprocess step, which is a sequence of abstractions performed once at the end. Clicking on the Add button brings up a list of all the possible abstraction methods. The selected abstraction method is added after the highlighted abstraction method in the list. If no method is highlighted, it makes it the first abstraction in the list. **iBioSim** supports many different types of abstraction. However, there are a few critical ones that we list here:

- complex-formation-and-sequestering-abstraction: this attempts to remove complex formation reactions. It replaces the complex species in rate laws with a function of its constituent species.
- operator-site-reduction-abstraction: this removes reactions and corresponding species involved in the binding of transcription factors to operator sites.
- distribute-transformer: applies the distributive law to kinetic law formulas to make it easier to identify the forward and reverse rates within a reversible reaction.
- reversible-to-irreversible-transformer: this converts a reversible reaction into two irreversible reactions. This is critical for the Monte Carlo simulation methods that require there to be no reversible reactions.
- kinetic-law-constants-simplifier: this replaces constant parameters within kinetic laws with their values to improve their evaluation time.



This tab also includes a few parameters for the abstraction methods.

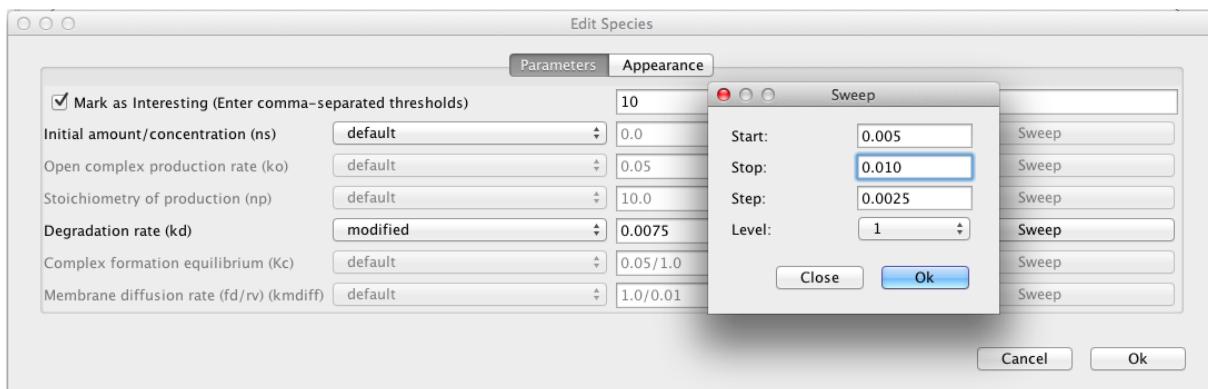
- Rapid Equilibrium Condition 1: specifies threshold T1 such that the rapid equilibrium condition fails when $T1 > E0/(S0 + k - 1/k1)$.
- Rapid Equilibrium Condition 2: specifies threshold T2 such that the rapid equilibrium condition fails when $T1 > k2/k - 1$.
- QSSA Condition: specifies threshold T used by the QSSA abstraction method where $T > E0/(S0 + KM)$.
- Max Concentration Threshold: specifies the maximum number of molecules that a species can have initially and still be considered an operator site by the operator site reduction.
- Grid Diffusion Stoichiometry Amplification: the number of molecules that are moved per diffusion reaction.

Note that these are not used by the common abstractions listed above.

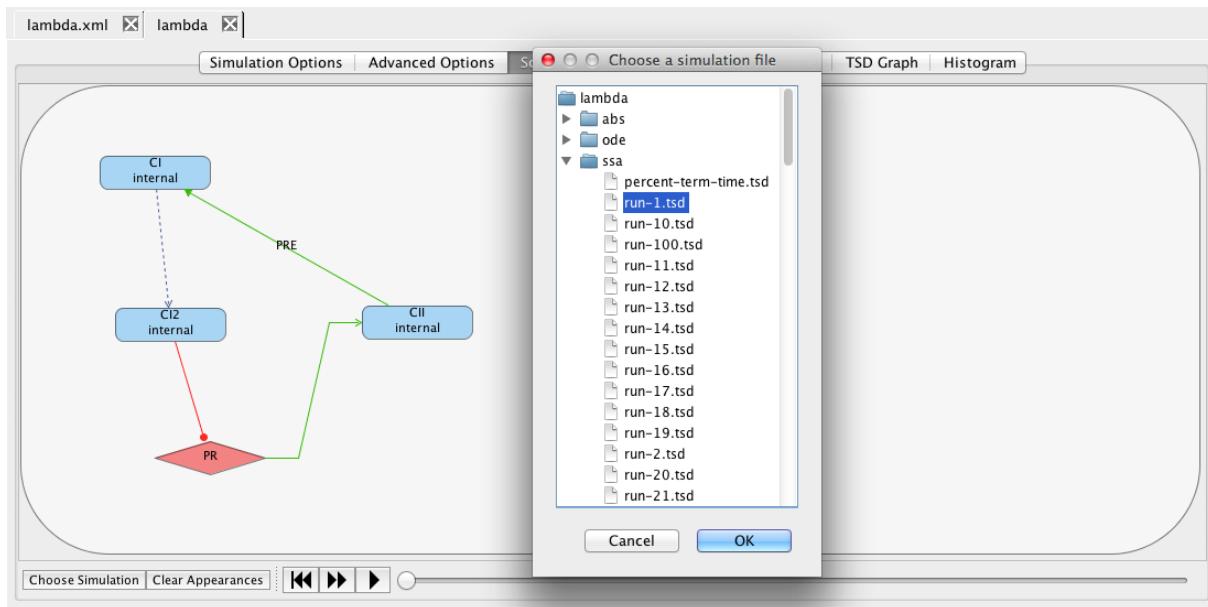
Finally, there are a few options used by the incremental SSA simulation method. The user can select the type of iSSA (MPDE, Mean Path, or Median Path). The user can also select whether or not to use an adaptive time step. Finally, the user can select the number of paths to follow during the iSSA.

5.3 Schematic

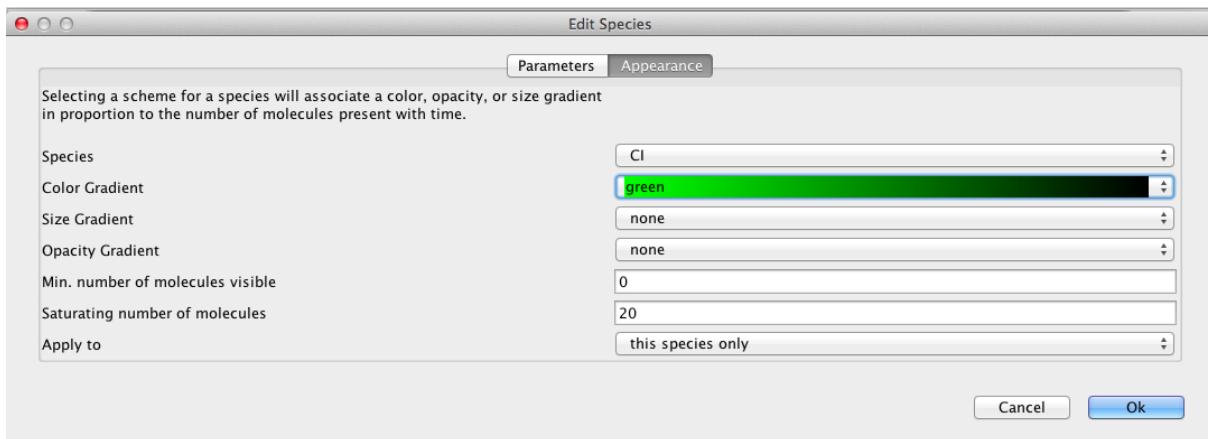
Using the schematic tab, you can adjust initial values and parameters, allowing one to perform simulations to determine the effect of these changes. Clicking on any species, promoter, reaction, or influence brings up the corresponding editor. To change a value, switch the corresponding combo box to modified, which will then allow you to change the value. In addition to making single changes, you can also sweep a value as shown below. When you click on the Sweep button, it brings up a window where you can select the start value, the stop value, and the step value. The level indicates how the sweep should perform when multiple variables are swept at the same time. Variables at the same level are changed at the same time. Furthermore, all variables on level 2 are stepped through all their values before changing the values of those variables on level 1. After the values on level 1 are changed, the values on level 2 are stepped through all their values again.



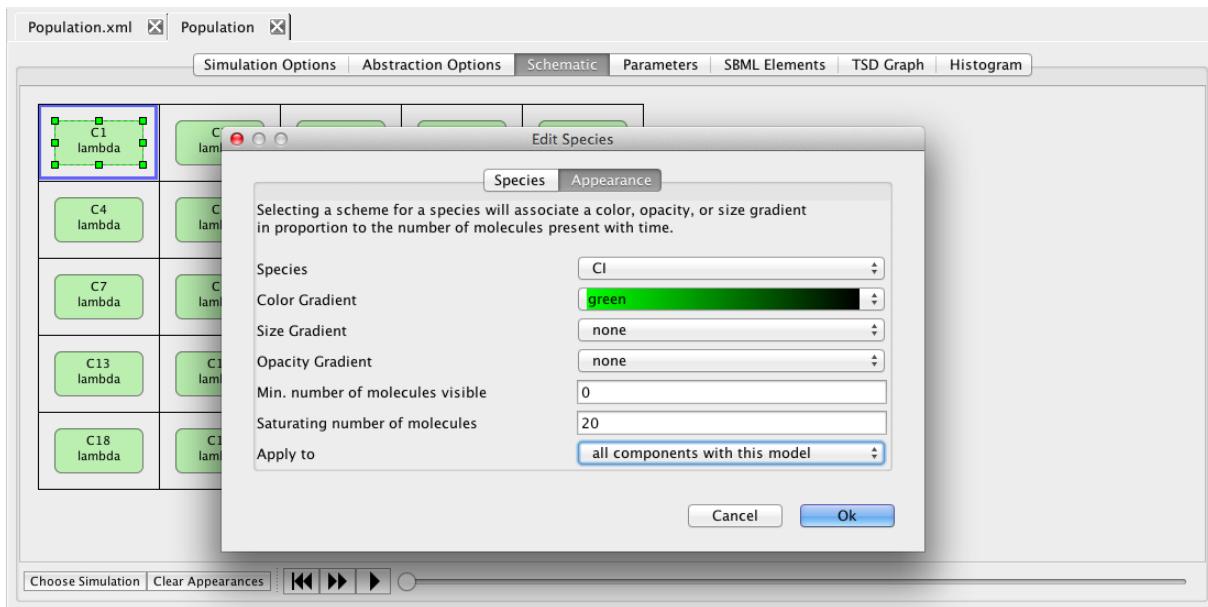
The schematic tab can also be used to visualize simulation results. To do this, after you have generated some simulation results, you can press the Choose Simulation button to bring up a window with all the simulations in this analysis view. You can then select the data that you would like to visualize.



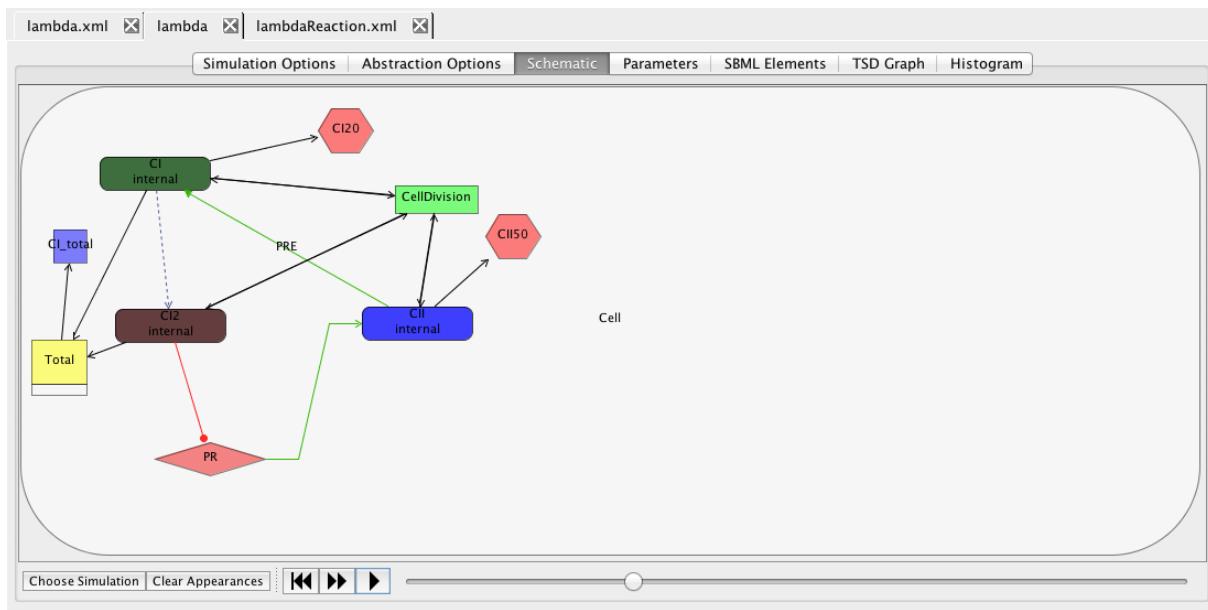
If you now click on a species, component, or grid location, you can modify how it appears as you playback the simulation. You can have it change color, size, and/or opacity on a gradient. You can also select the range of molecule counts to specify the ends of the gradient(s). Finally, you can indicate that these selections are either for this species or all species in the model. Appearances can be saved by clicking the save button.



The appearances editor for components and grid locations is similar. The differences are that you can select any species within the component (or diffusible species for the grid location). You also can copy the settings to all instantiations of the same model.

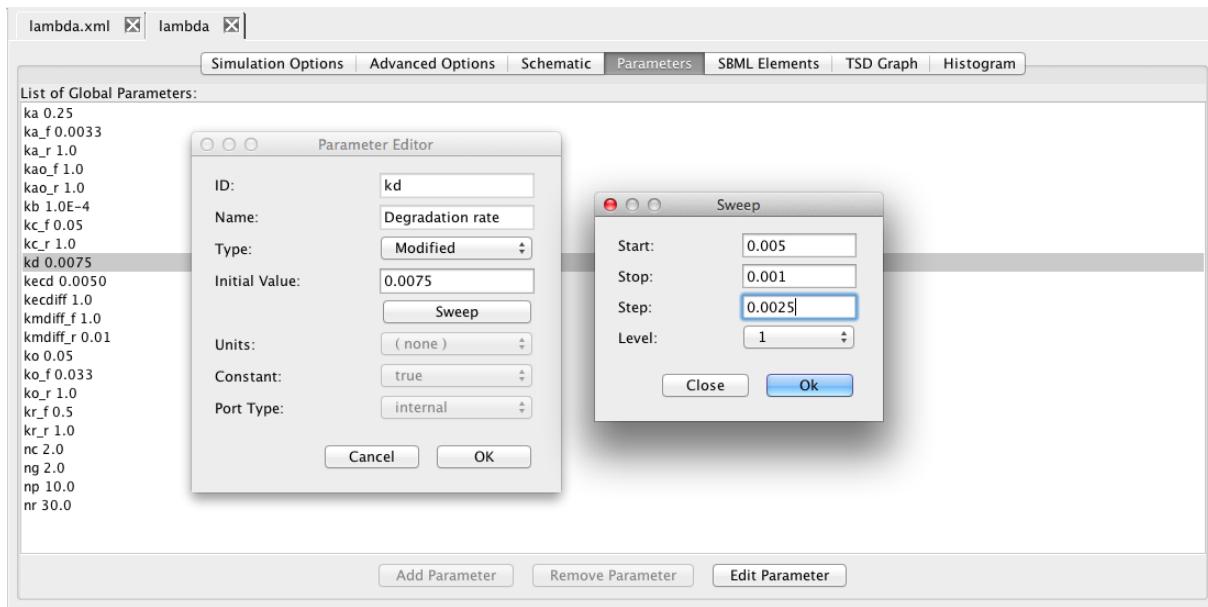


Once you have made your selections, you can now playback the simulation. You can either single-step the simulation by pressing the icon or play continuously by pressing the icon. The playback can also be paused by pressing the icon and restarted by pressing the icon. The slider can be dragged around to quickly reach a particular spot in simulation



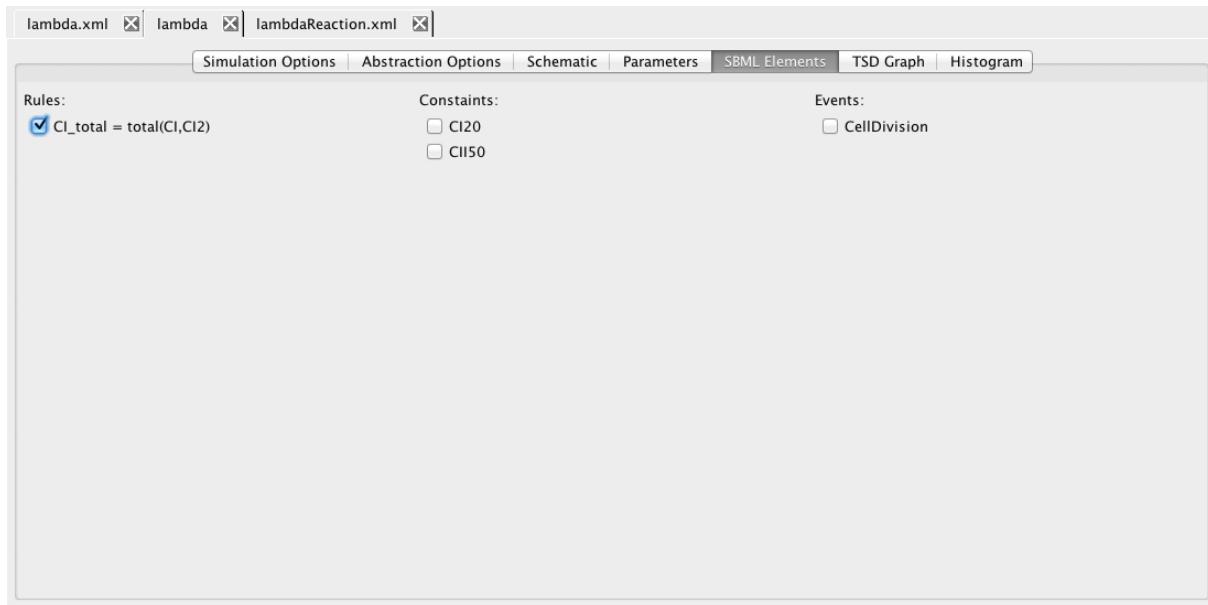
5.4 Parameters

The parameter tab allows the user to modify the value of any model generation or global parameter for a given analysis. Like the values on the schematic tab, they can also be swept to generate multiple simulation runs stepping through different values.



5.5 SBML Elements

The SBML elements tab allows you to select which SBML model elements to include in your analysis. This includes initial assignments, rules, constraints, and events. Elements that are checked are used during analysis. Otherwise, they are not used.



6 Learn Tool

The Learn Tool is used to discover genetic circuit connectivity from time series data. This tool uses the GeneNet algorithm described in Barker's PhD dissertation (University of Utah 2007). The first tab of the Learn Tool is the *data manager* which is shown below. It is used to both enter time series experimental data as well as bring data into the learn view. The Add button is used to create a new data file. After pressing this button, enter the name of the new data file, and then enter the data for this file using the data editor to the right. The Remove button deletes all highlighted files. Note that after highlighting one file, you can use the ctrl key to highlight additional files or the shift key to highlight a range of files. The Rename button is used to change the name of a data file. The Copy button copies a data file. The Copy From View button brings up a list of all analysis and learn views in the current project, and data from the selected view will be copied into this learn view. Finally, the Import button brings up a file browser, and it allows you to import a data file from outside this project. These files can be in time series data (TSD) format (see Section 11), comma separated value (CSV) format, or tab delimited format (DAT). The contents of the data file highlighted on the left appear in the data editor on the right. Individual data entries can be modified, new data points can be added using the Add Data Point button, data points can be removed using the Remove Data Point button, and data points can be copied using the Copy Data Point button. When you are satisfied with all your changes, you should press the Save button to record your changes.

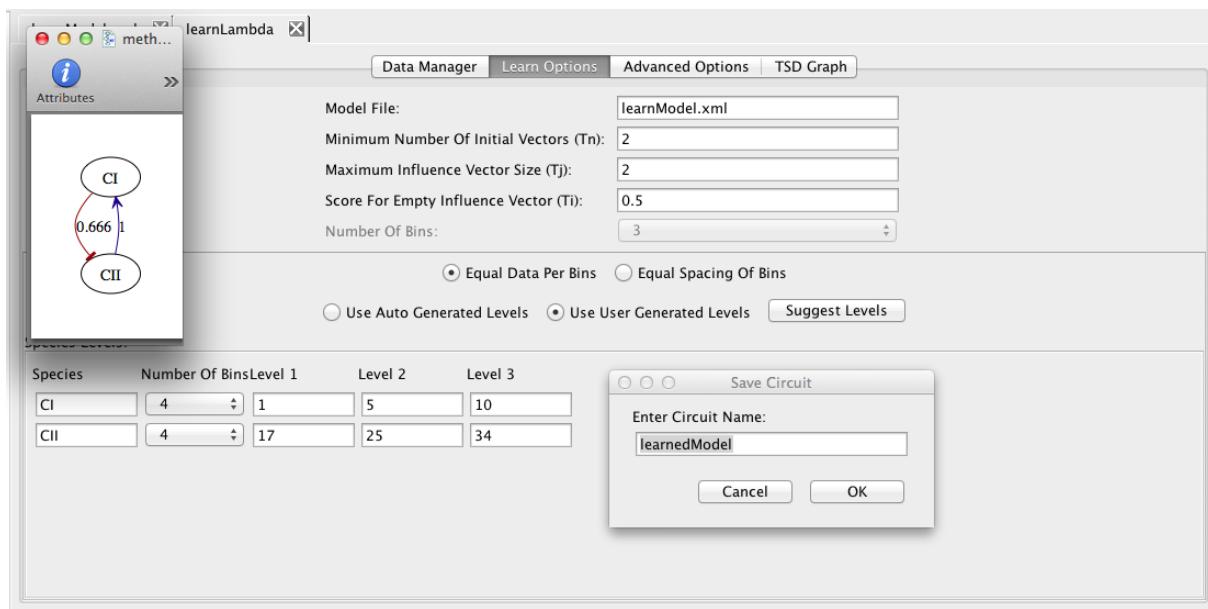
The screenshot shows the 'Data Manager' tab of the Learn Tool interface. On the left, a list of TSD files is displayed, with 'lambda/abs/run-1.tsd' currently selected. To the right is a data editor table with columns 'time', 'CI', and 'CII'. The table contains 25 rows of data, starting from time 0.0 and increasing by 10.0 units. The 'Save' button at the bottom is highlighted in blue, indicating it is the active button.

| time | CI | CII |
|-------|------|------|
| 0.0 | 0.0 | 0.0 |
| 10.0 | 0.0 | 17.0 |
| 20.0 | 0.0 | 16.0 |
| 30.0 | 0.0 | 15.0 |
| 40.0 | 10.0 | 25.0 |
| 50.0 | 10.0 | 34.0 |
| 60.0 | 8.0 | 31.0 |
| 70.0 | 8.0 | 27.0 |
| 80.0 | 8.0 | 26.0 |
| 90.0 | 7.0 | 26.0 |
| 100.0 | 7.0 | 21.0 |
| 110.0 | 7.0 | 18.0 |
| 120.0 | 6.0 | 17.0 |
| 130.0 | 6.0 | 16.0 |
| 140.0 | 4.0 | 16.0 |
| 150.0 | 4.0 | 16.0 |
| 160.0 | 4.0 | 15.0 |
| 170.0 | 4.0 | 15.0 |
| 180.0 | 3.0 | 14.0 |
| 190.0 | 3.0 | 14.0 |
| 200.0 | 3.0 | 24.0 |
| 210.0 | 3.0 | 22.0 |
| 220.0 | 3.0 | 22.0 |
| 230.0 | 3.0 | 21.0 |
| 240.0 | 3.0 | 20.0 |
| 250.0 | 3.0 | 18.0 |

The second tab allows the user to configure the basic options for the Learn Tool. The basic options include:

- Minimum Number of Initial Vectors (T_n) (default=2):
 T_n is a threshold value used in the CreateInfluenceVectorSet algorithm and represents the minimum number of influence vectors constructed in this algorithm.
- Maximum Influence Vector Size (T_j) (default=2):
 T_j is a threshold value used in the CombineInfluenceVectors algorithm to determine the maximal size of merged influence vectors.

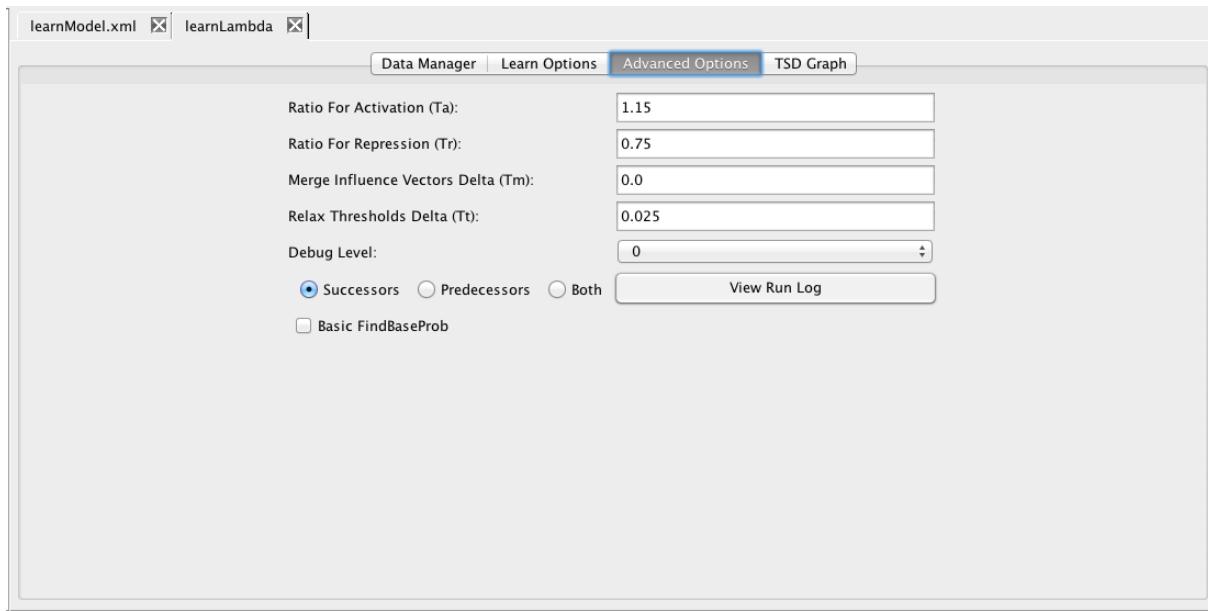
- Score for Empty Influence Vector (T_i) (default=0.5):
The score for an influence vector with no influences in it.
- Number of Bins (default=4):
The number of bins value specifies how many values the encoded time series data can assume.
- Equal Data Per Bins / Equal Spacing of Bins:
This radio button selects whether the auto generated levels should be determined by equally dividing the data between the bins or by equally dividing the range of the data.
- Use Auto Generated Levels / Use User Generated Levels:
This radio button allows the user to select whether they want the levels separating the bins to be auto generated or the user would like to provide them.
- When using user provided levels, the Suggest Levels button will provide the levels that would have been auto generated as a suggestion. These levels can then edited by the user. The number of bins for each species can also be individually adjusted.



The third tab allow the user to select some advanced options for the Learn Tool. These include:

- Ratio for Activation (T_a) (default=1.15):
A probability ratio above this value results in a vote for an influence vector that has a majority of activation influences.
- Ratio for Repression (T_r) (default=0.75):
A probability ratio above this value results in a vote for an influence vector that has a majority of repression influences.
- Merge Influence Vectors Delta (T_m) (default=0.0):
Two influence vectors cannot be merged unless the difference in their scores is less than this value.
- Relax Thresholds Delta (T_t) (default=0.025):
The values of T_a and T_r are modified by this amount when these thresholds are relaxed.

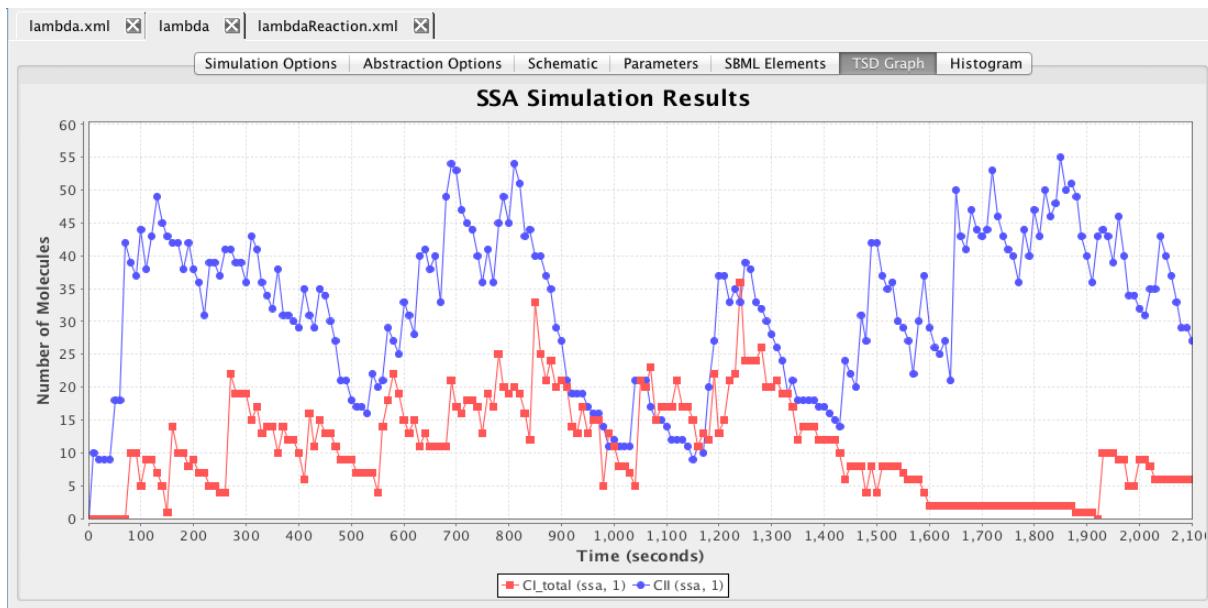
- Debug Level (default=0):
This controls how much information is displayed by the GeneNet algorithm when it runs.
- Successors / Predecessors / Both (default=Successors):
This radio button selects whether successor data point pairs, predecessor data point pairs, or both are used.
- Basic FindBaseProb (default=unchecked):
When selected, the basic FindBaseProb function is used.
- View Log button:
Opens a window containing the log file generated by GeneNet during the learn process for debugging purposes.



Once the Learn Tool has been provided data and configured as desired, pressing the Save and Run icon causes the tool to attempt to produce a model that may have produced this data. The resulting model is displayed using GraphViz, and the user is prompted to provide a model ID for the resulting model. If learning fails, an error message will be reported instead.

7 TSD Graph Editor

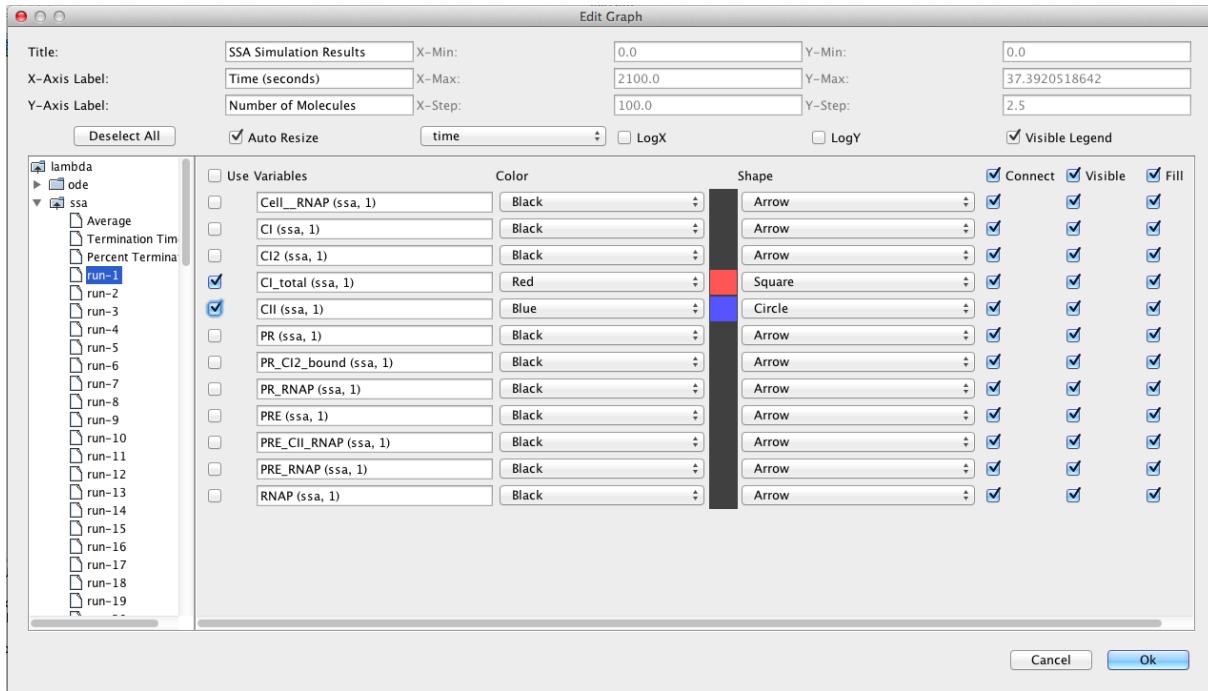
The TSD graph editor appears as a tab in both analysis and learn views. TSD graphs can also be created at the top-level of the project to allow you to integrate results from several analysis or learn views. These graphs can be created using the New → TSD Graph menu option in the File menu. Once created, they can be viewed and edited by double-clicking on the graph in the project window. An example graph is shown below.



In the TSD graph editor, a graph is created by double-clicking on the graph. You can then set various parameters and select what values you would like to have graphed. The parameters that you can select for a graph include:

- Title - the title of the graph.
- X-Axis Label - the label displayed for the x-axis.
- Y-Axis Label - the label displayed for the y-axis.
- X-Min - the starting value for the x-axis.
- X-Max - the ending value for the x-axis.
- X-Step - the increment for the x-axis.
- Y-Min - the starting value for the y-axis.
- Y-Max - the ending value for the y-axis.
- Y-Step - the increment for the y-axis.
- Auto Resize Check Box - determines whether to automatically resize the graph for best fit.
- X-Axis Combo Box - selects which variable to use for the x-axis (default=time).
- LogX - selects that the scale of the x-axis should be logarithmic.

- LogY - selects that the scale of the y-axis should be logarithmic.
- Visible Legend - selects that the legend should appear on the graph.



The data selection menu on the left displays all of the available sets of data that can be graphed. For a top-level graph, these data sets are organized hierarchically. Hierarchy is also introduced when different simulations in an analysis view are given different simulation IDs or after performing an analysis while sweeping parameter values. In addition to being able to plot results from individual simulation runs, the average, standard deviation, and variance are also provided. Finally, when constraints are used, the Termination Time and Percent Termination data is also computed. The Termination Time gives a plot of the number of runs that have terminated versus time, while Percent Termination gives the percentage of runs that have terminated versus time.

After selecting a data set, one can select individual variables (typically species) to graph and how they are to be displayed. In other words, for each species there are the following options:

- Use Check Box - determines whether or not this species is displayed on the graph. Checking or unchecking the box at the top changes the state for all species in the data set.
- Species Label - the name displayed in the legend.
- Drop Down Menu Of Colors - the color that is used for this species.
- Color Palette - clicking on the color palette lets one customize the color selection.
- Drop Down Menu Of Shapes - the shape that is used to mark the data points.
- Connect Check Box - determines whether to connect the points with a line. Checking or unchecking the box at the top changes the state for all species in the data set.
- Visible Check Box - determines whether shapes are visible on the line. Checking or unchecking the box at the top changes the state for all species in the data set.

- Fill Check Box - determines whether shapes are filled on the line. Checking or unchecking the box at the top changes the state for all species in the data set.

Note that a checkmark appears on a data set to indicate that some species have been selected in that data set. Also, all species can be deselected by pressing the Deselect All button.

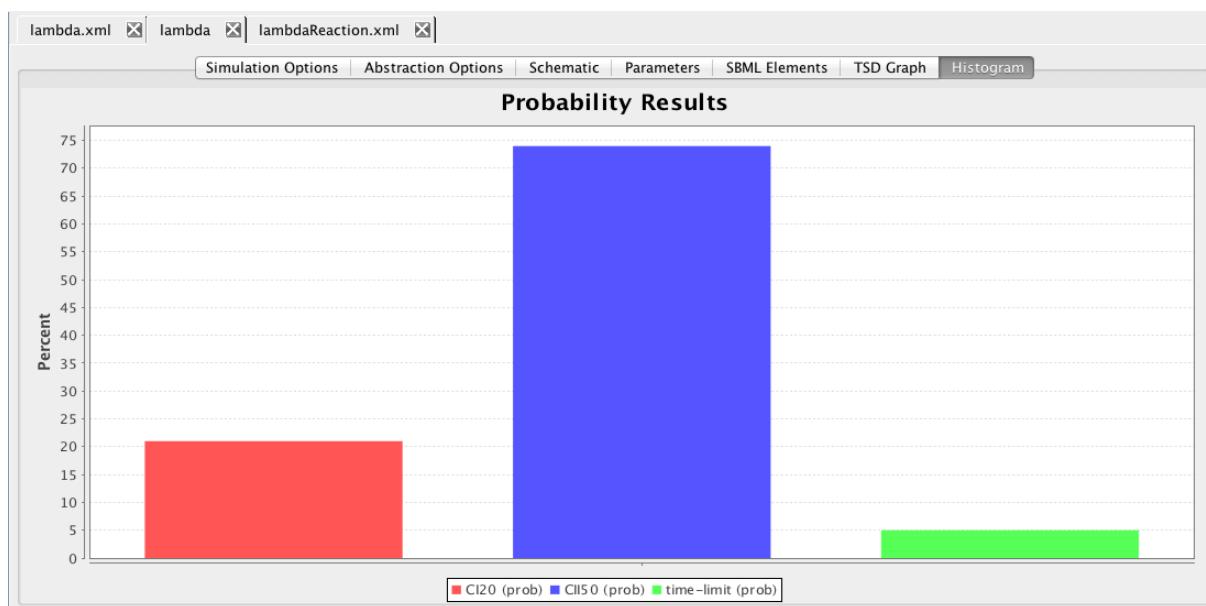
When in a TSD graph editor, pressing the Save icon  saves the settings for the graph to a file, so when you re-open the graph, it will reload this data and display in the same way as before. Pressing the Save As icon  prompts for a new filename and creates a new top-level graph with that name. Finally, pressing the Export icon  prompts for a filename and exports the data to the given name. The extension provided for the filename is used to determine how the graph is to be exported. The supported file types are:

- Time series data format (tsd), see Section 11..
- Comma separated value (csv).
- Column separated data (dat).
- Encapsulated postscript (eps).
- Joint Photographic Experts Group (jpg).
- Portable document format (pdf).
- Portable network graphics (png).
- Scalable vector graphics (svg).

If no extension is given, then the file type is the one specified in the file filter (default is pdf). For image (i.e., not data) file types, you will be prompted to give a desired pixel height and width for the file before the file is exported.

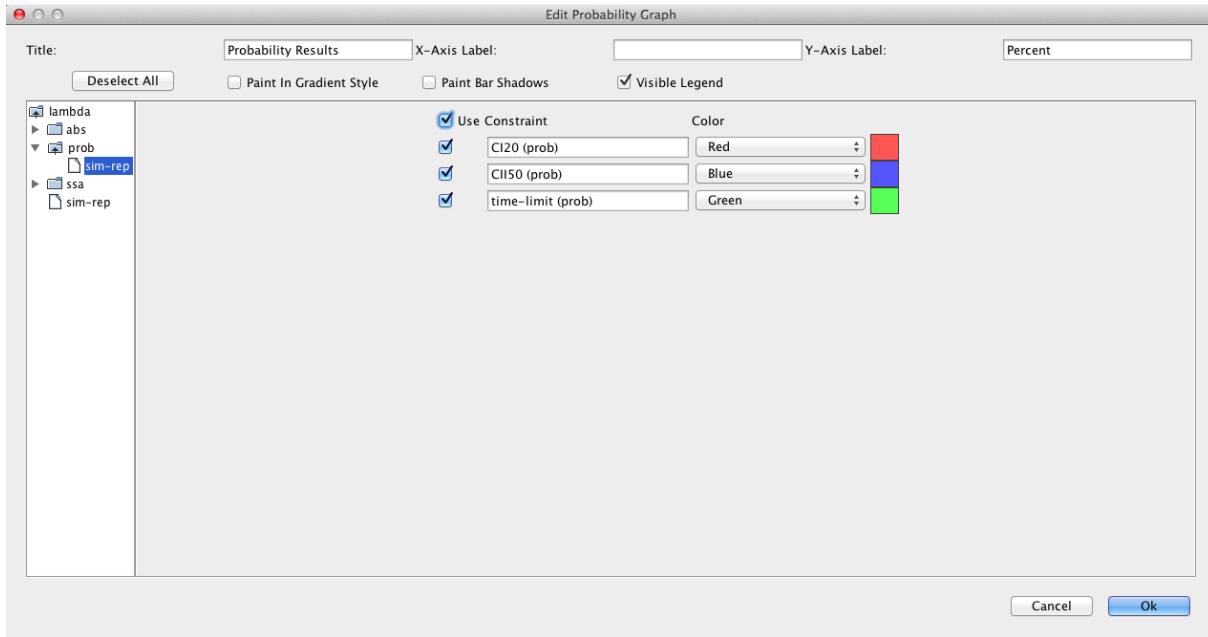
8 Histogram Graph Editor

iBioSim includes a histogram graph editor for visualizing probability data. In particular, this editor can display the reasons that a simulation terminated. Namely, it displays statistics on which constraints failed to hold, allowing the user to determine the likelihood of various conditions. The histogram graph editor appears as a tab in analysis views. Histograms can also be created at the top-level of the project to allow you to integrate results from several analysis views. These graphs can be created using the New → Histogram menu option in the File menu. Once created, they can be viewed and edited by double-clicking on the graph in the project window. An example histogram is shown below.



In the histogram graph editor, a graph is created by double-clicking on the graph. You can then set various parameters and select what values you would like to have graphed. The parameters that you can select for a graph include:

- Title - the title of the graph.
- X-Axis Label - the label displayed for the x-axis.
- Y-Axis Label - the label displayed for the y-axis.
- Paint in Gradient Style - creates a color gradient in the bars.
- Paint Bar Shadows - enables the bars to cast shadows.
- Visible Legend - selects whether a legend should be included in the graph.



The data selection menu on the left displays all of the available sets of data that can be graphed. For a top-level graph, these data sets are organized hierarchically. Hierarchy is also introduced when different simulations in an analysis view are given different simulation IDs or after performing an analysis while sweeping parameter values. After selecting a data set, one can select individual constraints to graph. The constraint labeled time-limit indicates that the simulation terminated while all constraints were still satisfied. One can also select how each result is to be displayed. In other words, for each constraint there are the following options:

- Use Check Box - determines whether or not this constraint is displayed on the graph. Checking or unchecking the box at the top changes the state for all constraints in the data set.
- Constraint Label - sets the name displayed in the legend.
- Drop Down Menu Of Colors - the color of the bar used for this constraint.
- Color Palette - clicking on the color palette lets one customize the color selection.

Note that a checkmark appears on a data set to indicate that some constraints have been selected in that data set. Also, all constraints can be deselected by pressing the Deselect All button.

When in a histogram graph editor, pressing the Save icon saves the settings for the graph to a file, so when you re-open the graph, it will reload this data and display in the same way as before. Pressing the Save As icon prompts for a new filename and creates a new top-level graph with that name. Finally, pressing the Export icon prompts for a filename and exports the data to the given name. The extension provided for the filename is used to determine how the graph is to be exported. The supported file types are:

- Encapsulated postscript (eps).
- Joint Photographic Experts Group (jpg).
- Portable document format (pdf).

- Portable network graphics (png).
- Scalable vector graphics (svg).

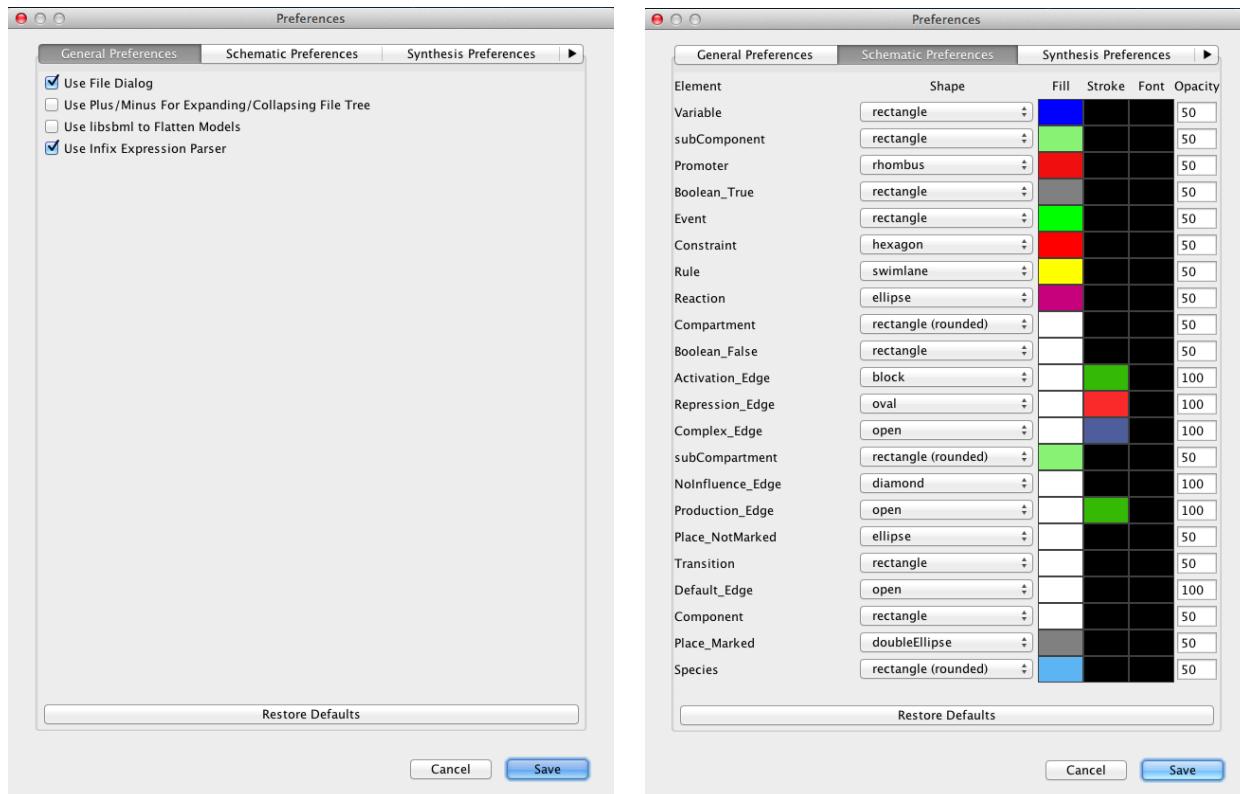
If no extension is given, then the file type is the one specified in the file filter (default is pdf). For image (i.e., not data) file types, you will be prompted to give a desired pixel height and width for the file before the file is exported.

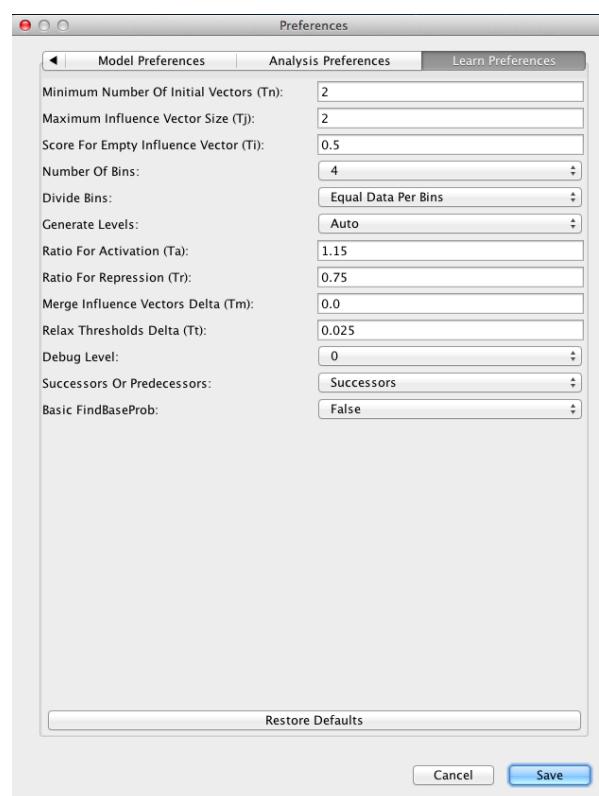
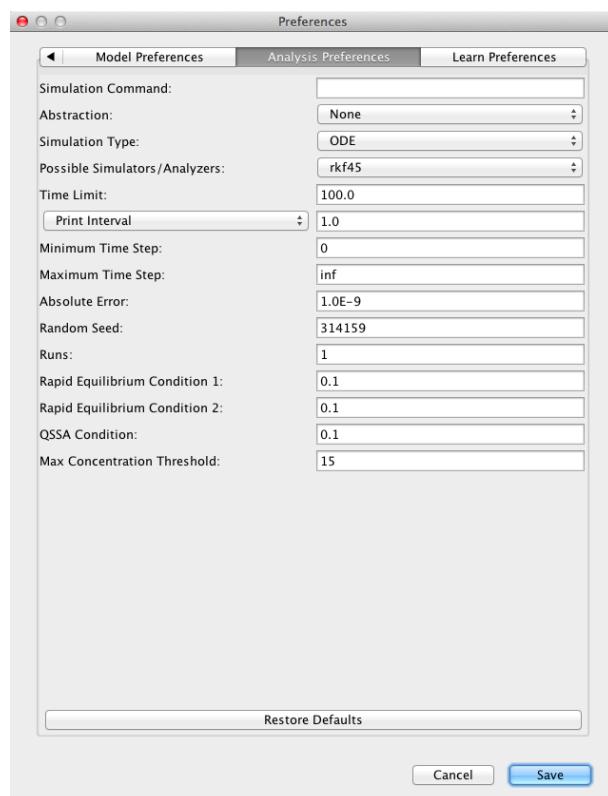
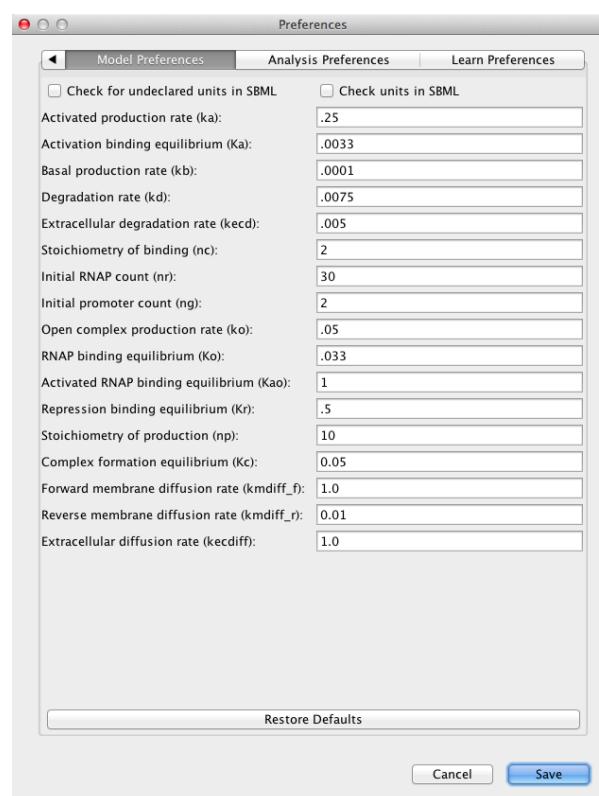
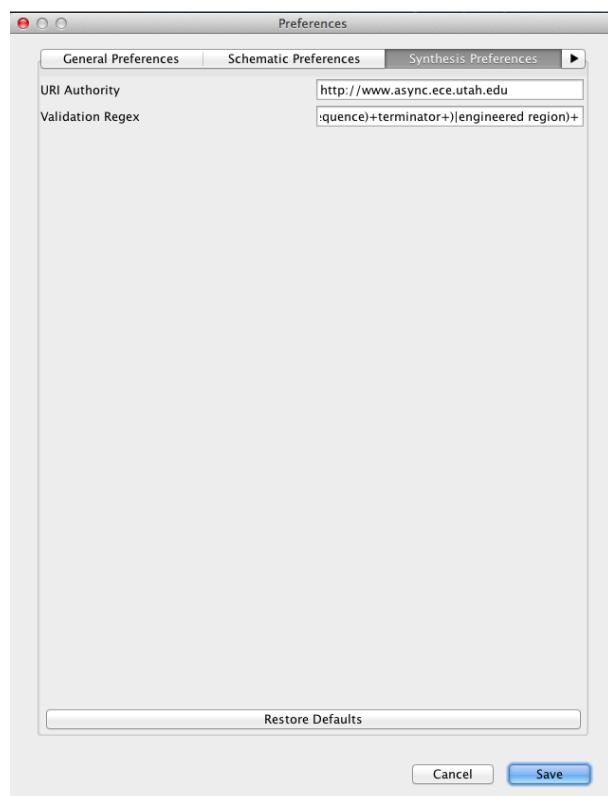
9 Preferences

User preferences can be set by selecting the **Preferences** option under the **File** menu on Linux and Windows or the **iBioSim** menu on MacOS. As shown below, under the General Preferences tab, the user can decide whether they wish to:

- Use a File Dialog for selecting files or the default Java File Chooser.
- Use plus/minus for expanding and collapsing file trees.
- Use the libsbml or the **iBioSim** flatten procedure.
- Use the infix expression parser in mathematical expressions.

The Restore Defaults button on this tab and the others restores the default preferences. In the Schematics preference tab, the user can decide the shape, color (fill, stroke, and font), and opacity for each schematic object. The Synthesis Preferences allows the user to select a default URI for new SBOL DNA components and a regular expression to validate generated SBOL DNA components. The Model Preferences tab allows users to select whether they wish to see warnings about undeclared units in SBML and whether they wish to check units at all. Finally, it allows users to change the default model generation parameter values. The Analysis Preferences tab allows users to change the default values used by the analysis tool. The simulation command in particular can be useful to select an alternative simulator. We've used it to select a script that executes the simulation on a compute server. Finally, the Learn Preferences tab allows users to change the default values used by the learn tool.





10 Mathematical Formulas

Math formulas appear in many model constructs. These formulas are expressed as text strings using a simple syntax. In particular, model math formulas can include:

- Variables (compartment, species, parameter IDs, and reaction IDs)
- Real Numbers
- Built-in constants: exponential, pi, true, and false.
- Special variable time or t, which returns the current simulation time.
- Mathematical operators including add (+), subtract (-), multiply (*), divide (/), and power (pow(x,y)).
- A function defined in the list of function definitions.
- Logical functions: and (&&), or (||), xor, not (!).
- Relational functions: eq (==), neq (!=), geq (\geq), gt ($>$), leq (\leq), and lt ($<$).
- Unary functions: abs, ceiling, exp, factorial, floor, ln, log, sqr, and sqrt.
- Trigonometric functions: cos, cosh, sin, sinh, tan, tanh, cot, coth, csc, csch, sec, sech, arccos, arccosh, arcsin, arcsinh, arctan, arctanh, arccot, arccoth, arccsc, arccsch, arcsec, and arcsech.
- The delay(expr1,expr2) function, which returns the value of expr1 at a time expr2 time units earlier (not currently supported by analysis).
- The piecewise(value1, case1, value2, case2, ..., otherwise) function returns value1 if case1 is true, value2 if case2 is true, etc. If no cases are true, it returns the otherwise value.

`iBioSim` simulators support a few special purpose built-in functions that are not standard parts of SBML. These include:

- `rate(a)` - returns the current rate of change for a .
- `BIT(a, b)` - assuming a is an expression that evaluates to an integer, this returns the bit at location specified by expression b .
- `BITNOT(a)` - assuming a is an expression that evaluate to an integer, this returns bitwise NOT of a .
- `BITAND(a, b)` - assuming a and b are expressions that evaluate to integers, this returns bitwise AND of a and b .
- `BITOR(a, b)` - assuming a and b are expressions that evaluate to integers, this returns bitwise OR of a and b .
- `BITXOR(a, b)` - assuming a and b are expressions that evaluate to integers, this returns bitwise XOR of a and b .
- `mod(a, b)` - assuming a and b are expressions that evaluate to integers, this returns the remainder after dividing a by b (i.e., $a - \text{floor}(a/b) * b$).

iBioSim's simulators also support several random functions which are added by default to any model file. The following random functions, therefore, can also be used in model math formula:

- Continuous random functions: uniform(a,b), normal(m,s), exponential(mu), gamma(a,b), log-normal(z,s), chisq(nu), laplace(a), cauchy(a), and rayleigh(s).
- Discrete random functions: poisson(mu), binomial(p,n), and bernoulli(p).

Finally, **iBioSim** includes built-in functions to support stochastic model checking of CSL formulas. CSL is a temporal logic which allows one to specify probabilistic temporal logic properties. Analysis of a CSL property should return the probability that the property is true. These properties can either be transient properties to ask if something happens within a certain amount of time, or steady-state properties to ask the probability of a given state as time goes to infinity. The built-in functions include:

- $PG(t, x)$ - returns the probability that the expression x remains true during the time period specified by the expression t .
- $G(t, x)$ - returns true if the expression x remains true during the time period specified by expression t (i.e., $\neg t \vee x$).
- $PF(t, x)$ - returns the probability that the expression x becomes true during the time period specified by the expression t .
- $F(t, x)$ - returns false if the expression x becomes true during the time period specified by the expression t (i.e., $\neg t \vee \neg x$).
- $PU(t, x, y)$ - returns the probability that the expression x remains true until y becomes true, and y becomes true during the time period specified by the expression t .
- $U(t, x)$ - returns true if the expression x remains true until y becomes true, and y is true by the time specified by expression t (i.e., $G(t, x) \vee F(t, y)$).

11 Time Series Data Format

The time series data (tsd) format is composed of a parenthesized and comma-separated set of time points. Each time point is composed of a parenthesized and comma-separated set of data for that time point. This first time point is composed of a set of strings that are the labels for the data entries. The first entry in each time point is, by convention, the time for that time point. Below is an example simulation of the species CI and CII from 0 to 1000 seconds with time points separated by 100 seconds.

```
(("time","CI","CII"), (0,0,0), (100,0,19), (200,20,25), (300,19,18), (400,17,20), (500,17,46),  
(600,26,40), (700,43,43), (800,63,28), (900,72,34), (1000,72,28))
```

12 Tutorial

A detailed tutorial and a separate grid tutorial are available in the `docs` directory that comes with the distribution.

13 Reporting Bugs and Feature Requests

In order to report a bug or to request a change or feature, please send an email to:
`atacs-bugs@vlsigroup.ece.utah.edu`. The subject line must begin with one of the following keywords or the mail will be filtered by our spam filters:

- BUG - error or crash of the software
- CHANGE - something which can be improved
- FEATURE - something new