Engineering Genetic Circuits

Chris J. Myers

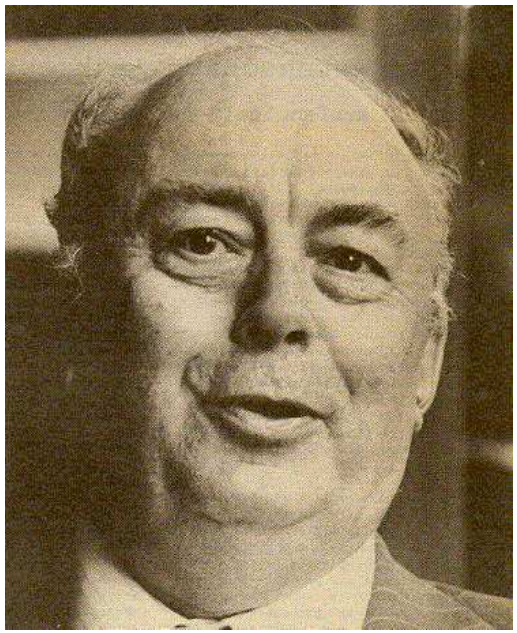
Lecture 5: Genetic Circuit Models

Models

Johann Von Neumann

The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.

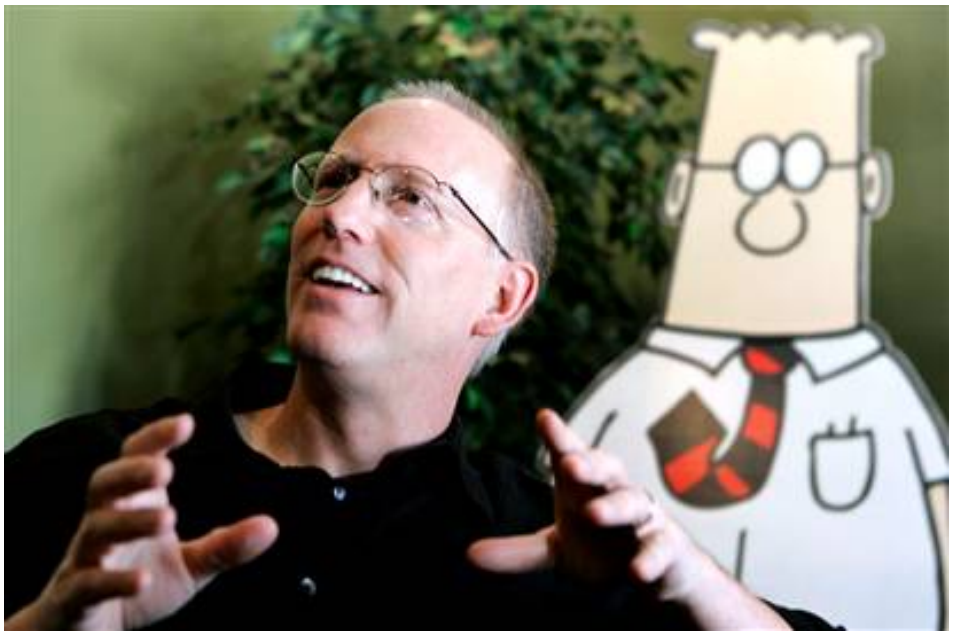
Models



Henri Theil

Models are to be used, not believed.

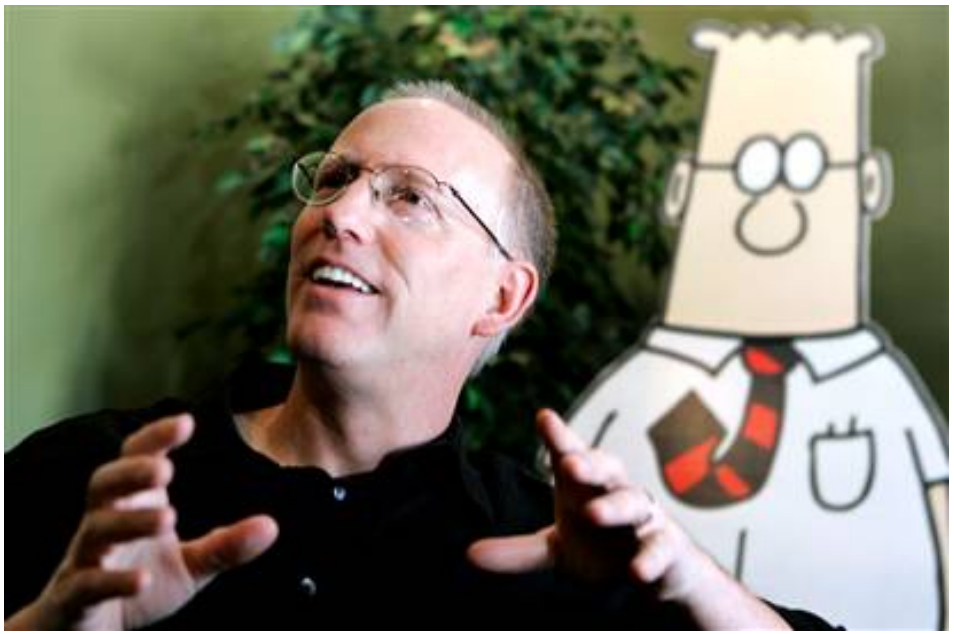
Models



Scott Adams

There are many methods for predicting the future. For example, you can read horoscopes, tea leaves, tarot cards, or crystal balls. Collectively, these methods are known as "nutty methods."

Models



Scott Adams

There are many methods for predicting the future. For example, you can read horoscopes, tea leaves, tarot cards, or crystal balls. Collectively, these methods are known as "nutty methods."

Or you can put well-researched facts into sophisticated computer models, more commonly referred to as "a complete waste of time."

Chemical Reaction Models

- *Atoms* are the basic building block for all matter.
- About 98 percent of any living organism consists of: hydrogen (H), carbon (C), nitrogen (N), oxygen (O), phosphorus (P), and sulfur (S).
- All material is created or destroyed via *chemical reactions*.
- Chemical reactions combine *atoms* to form *molecules* and combine simpler molecules to form more complex ones.
- Atoms form molecules via *covalent*, *ionic*, and *hydrogen bonds*.
- Chemical reactions can also work in reverse.

Chemical Reaction Example

$$2H_2 + O_2 \xrightarrow{k} 2H_2O$$

- H_2, O_2, and H_2O are *chemical* (or *molecular*) *species*.
- Subscripts indicate H and O are present in *dimer* form.
- The molecules H_2 and O_2 are known as the *reactants*.
- The water molecule, H_2O, is known as the *product*.
- The 2's indicate 2 H_2 molecules are used to produce 2 water molecules.
- These numbers are known as the *stoichiometry* of the reaction.
- Since matter is conserved, atom counts on each side must equal.
- Some reactions in this course may not have this property.

Rate Constants

- The *k* above the arrow is known as the *rate constant*.
- It indicates the probability or speed of this reaction.
- Used in many of the modeling techniques in this course.
- Often difficult to determine for bio-chemical reactions.

Law of Mass Action

- Rate of a chemical reaction is governed by rate constant and concentrations of reactants raised to power of stoichiometry.
- This is known as the *law of mass action*.
- The rate of water formation is:

$$\frac{d[H_2O]}{dt} = 2k[H_2]^2[O_2]$$

  where $[H_2O]$, $[H_2]$, and $[O_2]$ represent the concentration of water, hydrogen dimers, and oxygen dimers.
- 2 in front of k is due to this reaction producing two water molecules.

Laws of Thermodynamics

- Chemical reactions must obey the *laws of thermodynamics*.
- First law is that energy can be neither created nor destroyed.
- Second law is *entropy* (disorder in the universe) must increase.
- These two laws can be combined into a single equation:

$$\Delta H = \Delta G + T\Delta S$$

where ΔH is change in bond energy, ΔG is change in free energy, T is the absolute temperature, and ΔS is change in entropy.

Gibb's Free Energy

- ΔG is also known as the *Gibb's free energy* after J. Willard Gibbs who introduced this concept in 1878.
- Consider a *reversible reaction* of the form:

$$2H_2 + O_2 \overset{K_{eq}}{\leftrightarrow} 2H_2O$$

where $K_{eq} = k_f / k_r$ is the *equilibrium constant*.

- The Gibb's free energy for the forward reaction is:

$$\Delta G = \Delta G^\circ + RT \ln\{([H_2O]^2)/([H_2]^2[O_2])\}$$

where $R = 1.987$ cal/mol is the *gas constant* and T is the temperature.

Gibb's Free Energy (cont)

- The value of ΔG° is related to K_{eq}:

$$\Delta G^\circ = -RT \ln K_{eq}$$

- Combining equations results in:

$$\Delta G = RT \ln \frac{k_r [H_2O]^2}{k_f [H_2]^2 [O_2]}$$

- When negative, forward reaction can occur spontaneously.
- When positive, reverse reaction can occur spontaneously.
- When zero, the reaction is in a *steady state*.

Hydrolysis of ATP

- How do chemical reactions with positive free energy occur?
- Free energies of chemical reactions are additive.
- Coupling with other reactions allows them to occur.
- *Hydrolysis* of ATP releases energy:

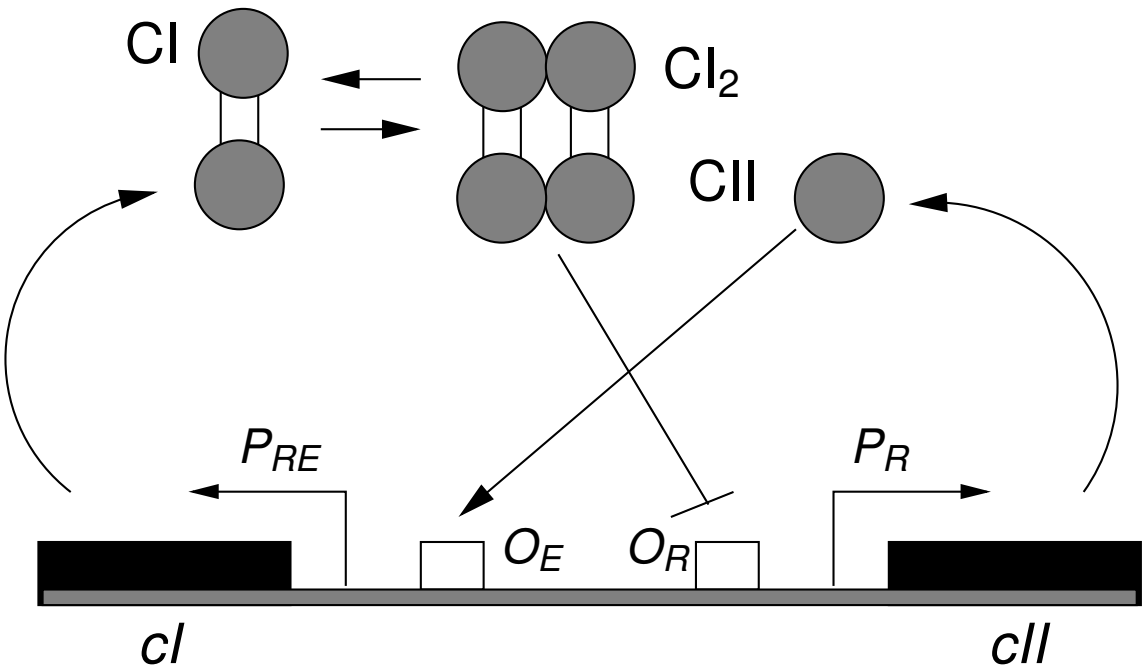
$$ATP + H_2O \leftrightarrow HPO_4^{2-} + ADP.$$

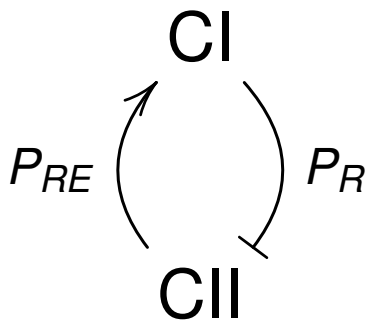
- These types of ATP reactions occur in all living organisms.
- ATP is the universal energy currency of living organisms.

Enzymes

- *Activation energy* barrier must be overcome.
- An *enzyme*, or catalyst, can accelerate a reaction without being consumed by the reaction.
- *Modifier* is a species that is not consumed by a reaction.
- Often enzyme amount much smaller than other reactants.
- Enzymes do not effect free energy of the reaction, but only help the reaction overcome its activation energy barrier.

Genetic Circuit Models

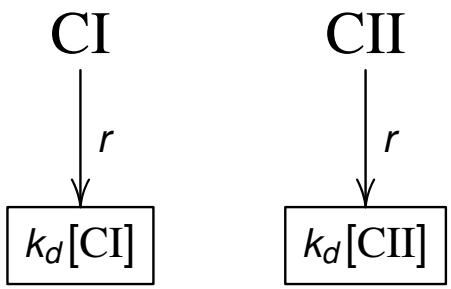




Creating a Chemical Reaction Model

- Create a species for RNAP as well as for each promoter and protein.
- Create degradation reactions for each protein.
- Create open complex formation reactions for each promoter.
- Create dimerization reactions, if needed.
- Create repression reactions for each repressor.
- Create activation reactions for each activator.

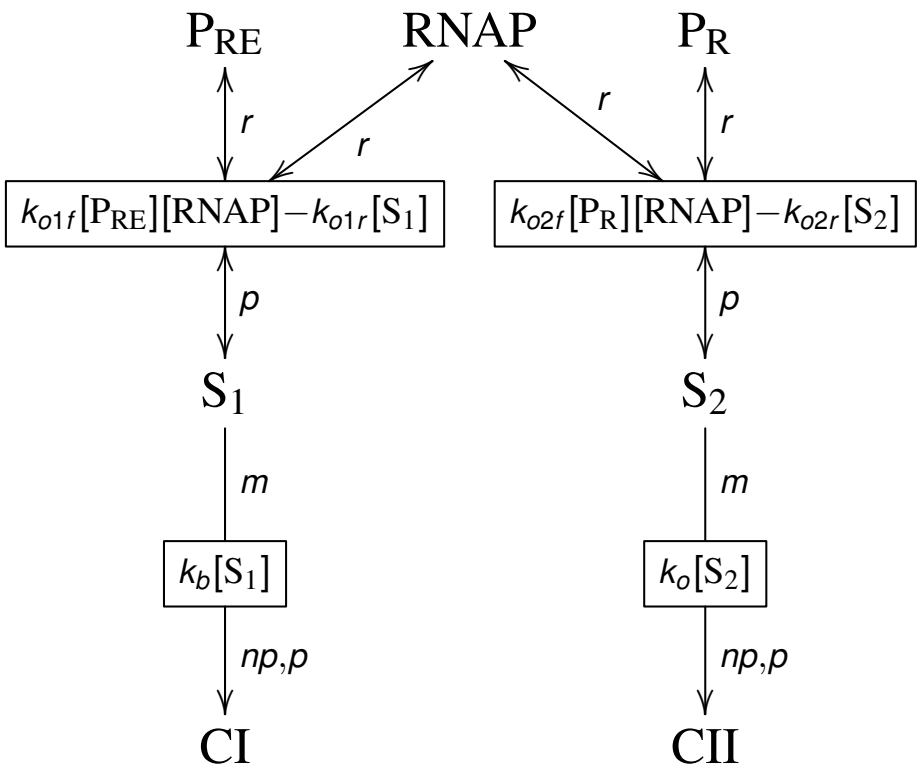
Degradation Reactions



$$\mathrm{CI} \xrightarrow{k_d} ()$$
$$\mathrm{CII} \xrightarrow{k_d} ()$$

Constant	*Value*
k_d	$0.0075\ \mathrm{sec}^{-1}$

Open Complex Formation Reactions



$$P_{RE} + \text{RNAP} \xleftrightarrow{K_{o1}} S_1$$

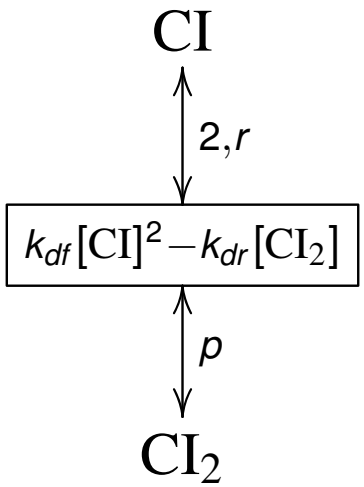
$$P_R + \text{RNAP} \xleftrightarrow{K_{o2}} S_2$$

$$S_1 \xrightarrow{k_b} S_1 + np\ \text{CI}$$

$$S_2 \xrightarrow{k_o} S_2 + np\ \text{CII}$$

Constant	*Value*
RNAP_0	30 nM
$K_{o1} = k_{o1f}/k_{o1r}$	0.01 M^{-1}
$K_{o2} = k_{o2f}/k_{o2r}$	0.69422 M^{-1}
k_b	0.00004 sec^{-1}
k_o	0.014 sec^{-1}
np	10

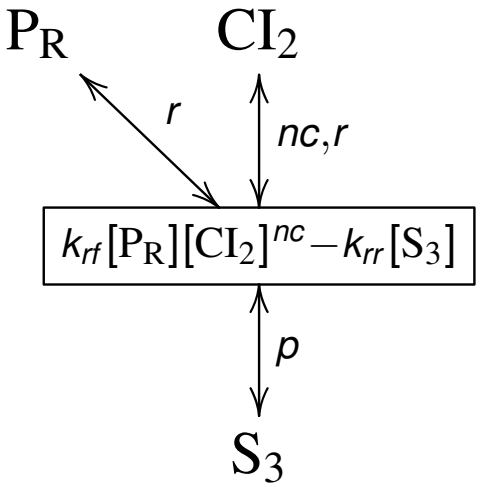
Dimerization Reactions



$$2CI \overset{K_d}{\longleftrightarrow} CI_2$$

Constant	*Value*
$K_d = k_{df}/k_{dr}$	$0.1 M^{-1}$

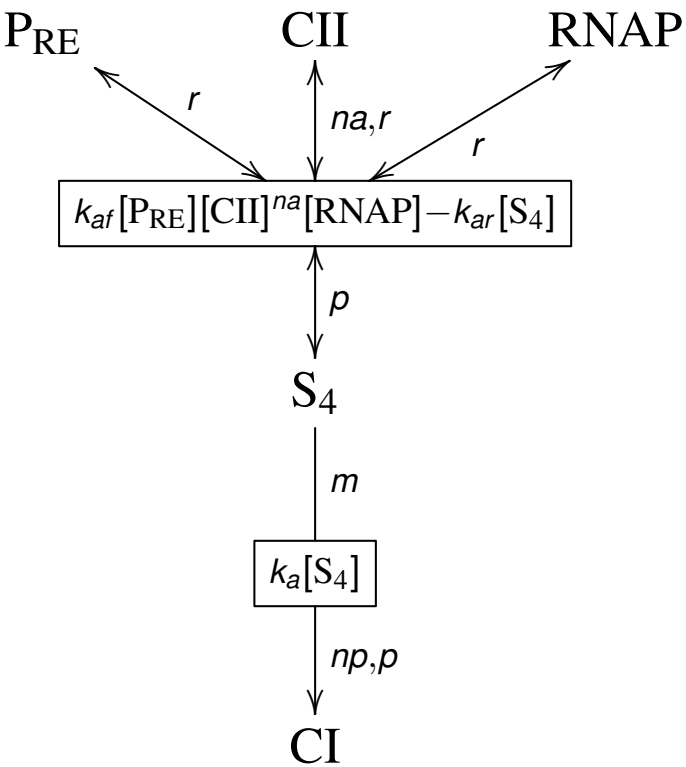
Repression Reactions



$$P_R + nc\ CI_2 \overset{K_r}{\longleftrightarrow} S3$$

Constant	*Value*
$K_r = k_{rf}/k_{rr}$	$0.2165\ M^{-nc}$
nc	1

Activation Reactions

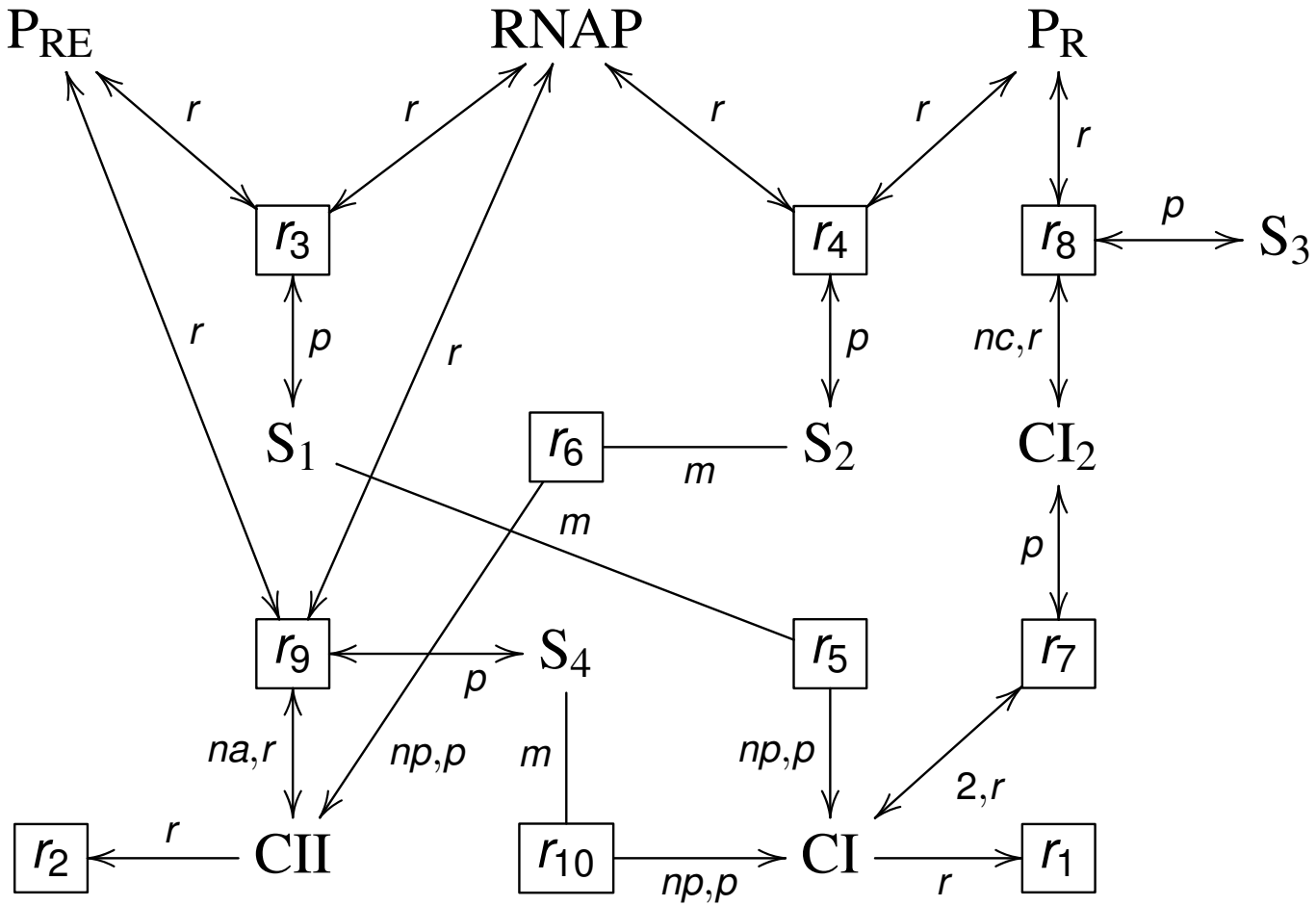


$$P_{RE} + na\,\mathrm{CII} + \mathrm{RNAP} \overset{K_a}{\longleftrightarrow} S4$$

$$S4 \xrightarrow{k_a} S4 + np\,\mathrm{CI}$$

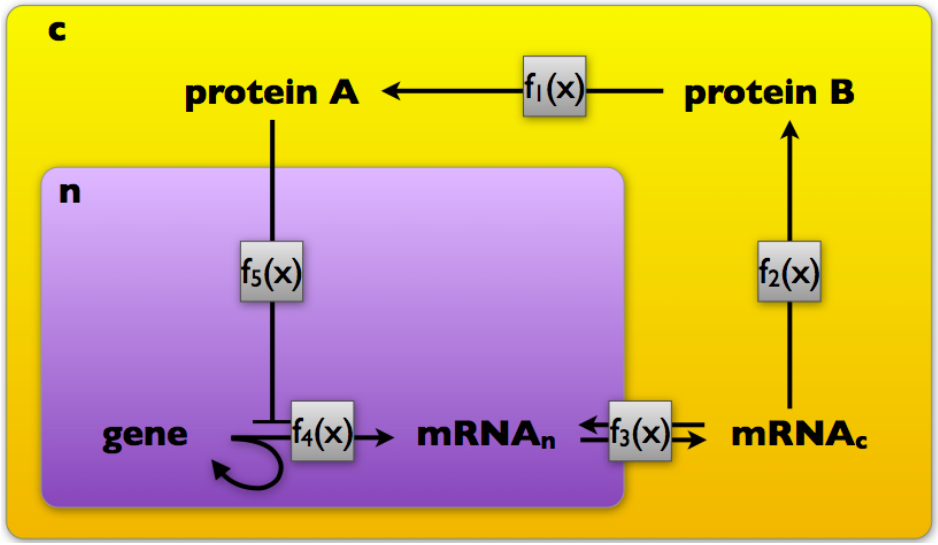
Constant	*Value*
$K_a = k_{af}/k_{ar}$	$0.00161\ M^{-(na+1)}$
k_a	$0.015\ \mathrm{sec}^{-1}$
na	1
np	10

Complete Reaction-Based Model



Systems Biology Markup Language (SBML)

- *Systems Biology Markup Language* (SBML) has been proposed as a standard representation for the modeling of biological systems.
- SBML models biological systems at the molecular level.
- A typical SBML model is composed of a number of chemical *species* (i.e., proteins, genes, etc.) and *reactions* that transform these species.



- SBML is supported by more than 290 tools, enabling researchers to create, annotate, simulate, and visualize models.
- SBML models can also be archived in the BioModels database.

Additional SBML Constructs

- *Unit definitions* are used to construct user-defined units which are derived from the set of base units.
- *Function definitions* are used to create user defined functions that can then be used in SBML math formulas.
- *Initial assignments* provide a formula to determine the initial value of a compartment size, species amount or concentration, or parameter.
- *Algebraic rules* specify relationships which must be maintained.
- *Assignment rules* specify the value of a compartment size, species amount or concentration, or parameter with a formula.
- *Rate rules* specify the rate of change of a compartment size, species amount or concentration, or parameter with a formula.
- *Events* are used to specify discrete changes of compartment sizes, species amounts or concentrations, and parameter values.
- *Constraints* specify properties that should never be violated.

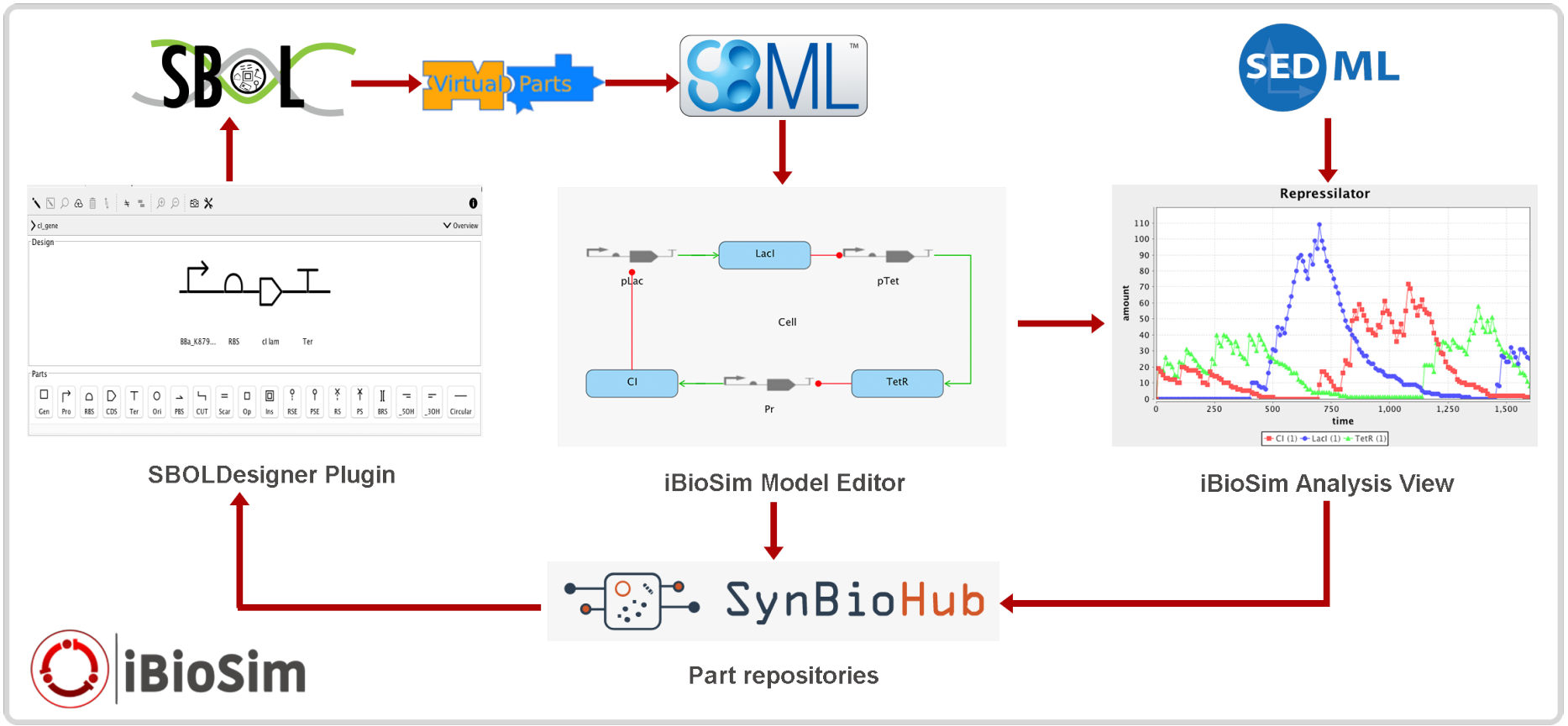
SBML Level 3 and Packages

- One of the major enhancements of SBML Level 3 is the support for packages to extend the modeling capabilities.
- A package can either add completely new elements or simply new parts to existing elements.
- For example, a layout package may add to a species its *x* and *y* coordinates for a graphical editor.
- Numerous packages have been proposed including *layout*, *hierarchical model composition*, *arrays*, *dynamic modeling*, *qualitative modeling*, *flux balance constraints*, etc.

Intelligent Biological Simulator (iBioSim)

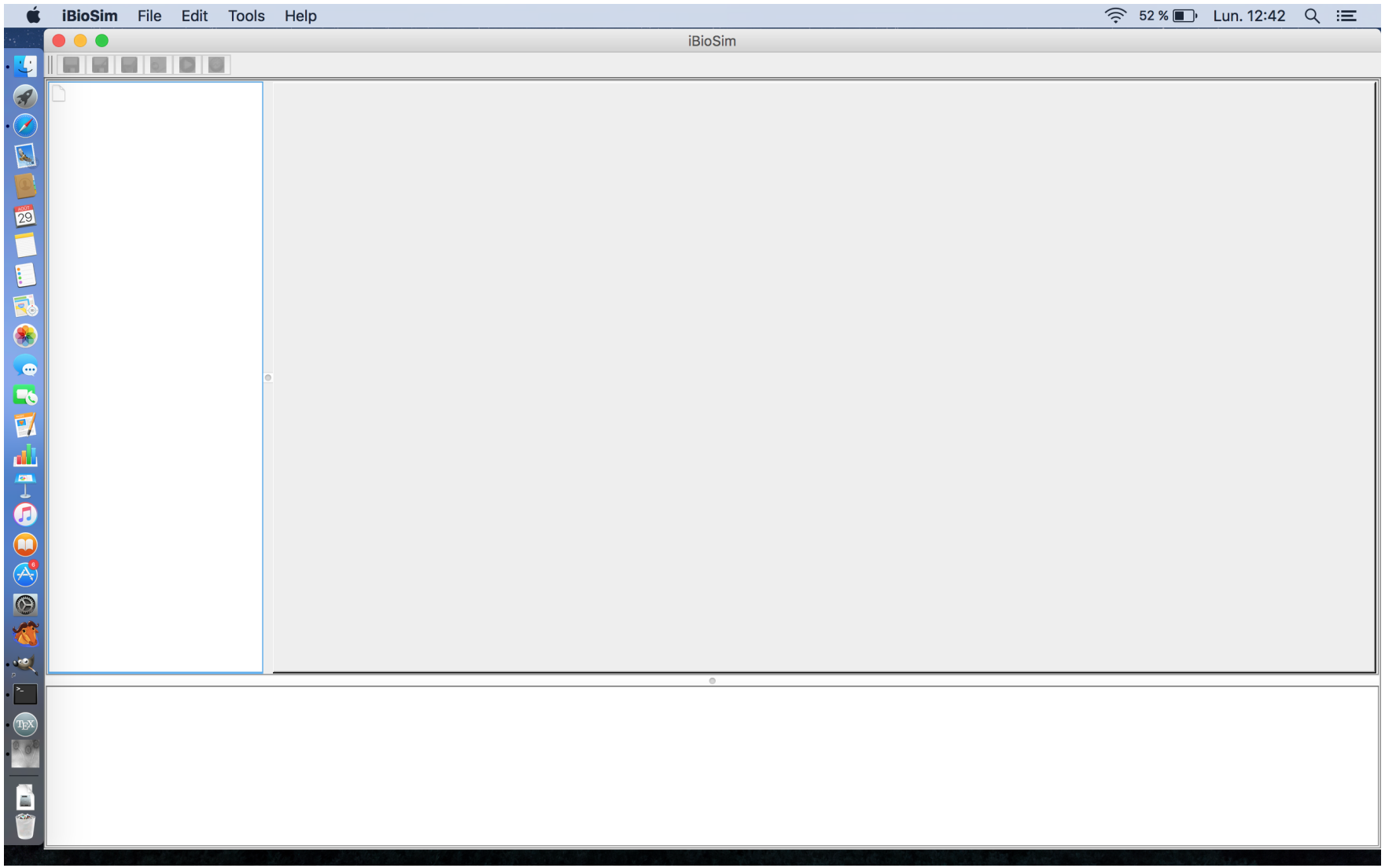
- `iBioSim` is for the modeling, analysis, and design of genetic circuits.
- Can also be used for any biochemical reaction-based model.
- Makes extensive use of standard data representations.
  - Models specified in the *Systems Biology Markup Language* (SBML).
  - Can import all levels and versions of SBML and is able to export L3 V1.
  - Supports all core SBML constructs, as well as, hierarchical model composition, layout, flux balance constraints, distributions, and arrays.
  - First tool to produce correct results for entire SBML testsuite.
  - Also tested on stochastic benchmark suite and curated BioModels.
  - Imports Simulation Experiment Description Markup Language (SED-ML) files that specify the analysis to perform.
  - Supports import/export of COMBINE Archives.
  - One of the first tools to support *Synthetic Biology Open Language* (SBOL).
  - Can fetch parts from and store designs in `SynBioHub`.

iBioSim Overview

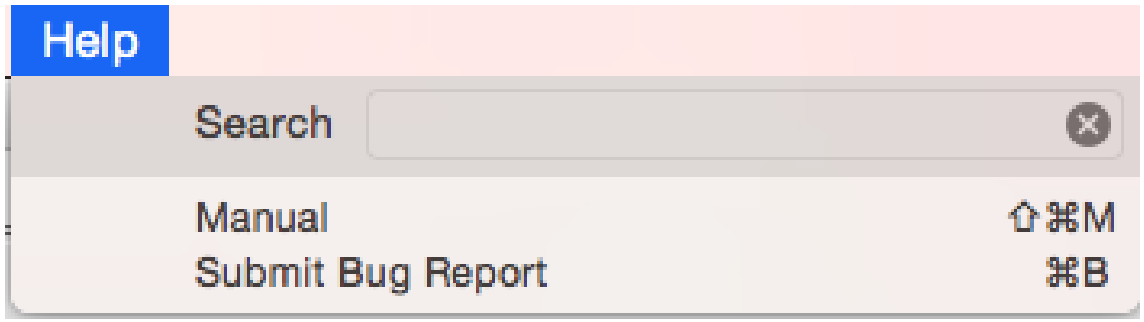


http://www.async.ece.utah.edu/ibiosim

iBioSim Window Layout

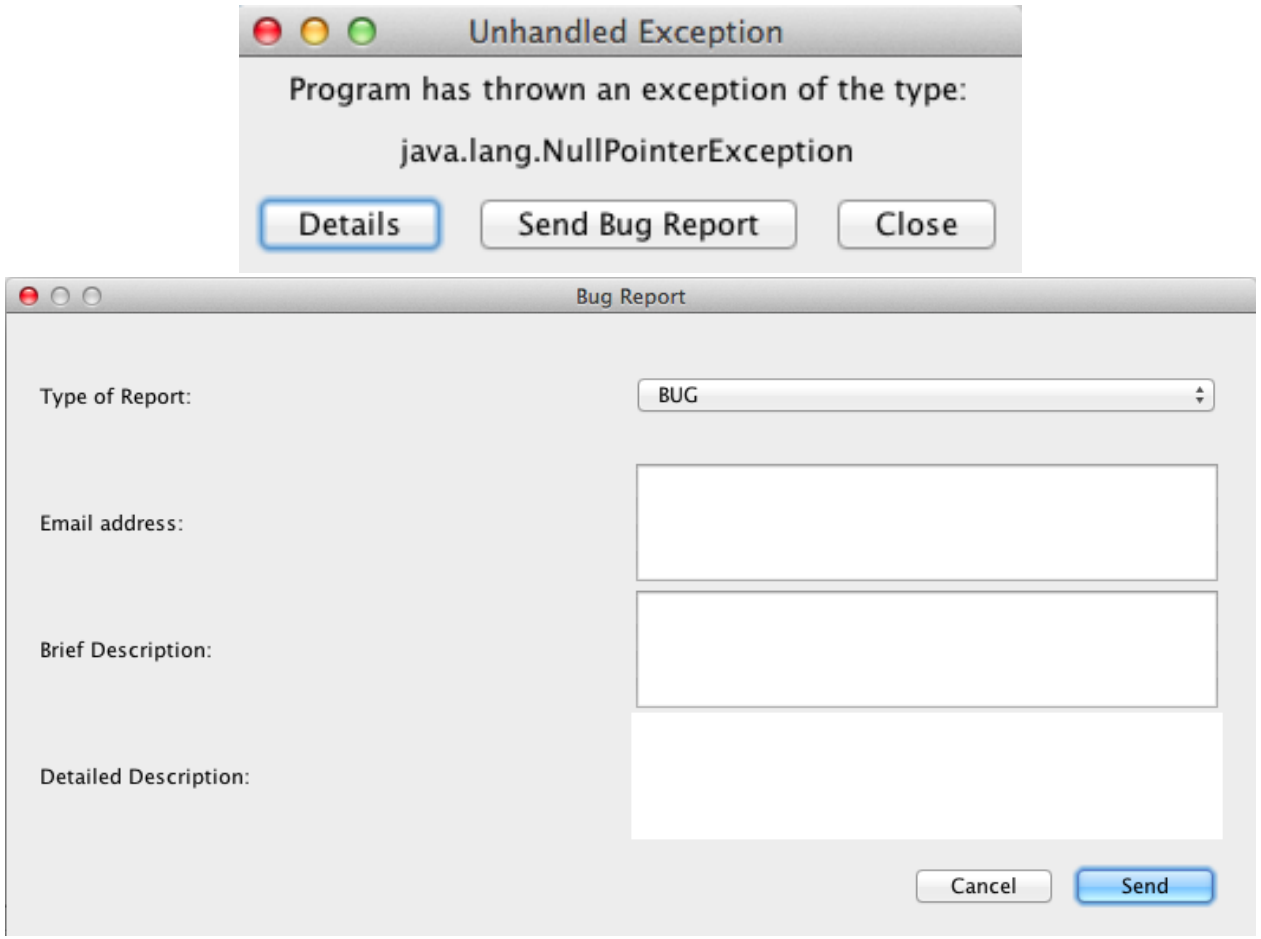


iBioSim Manual and Tutorials



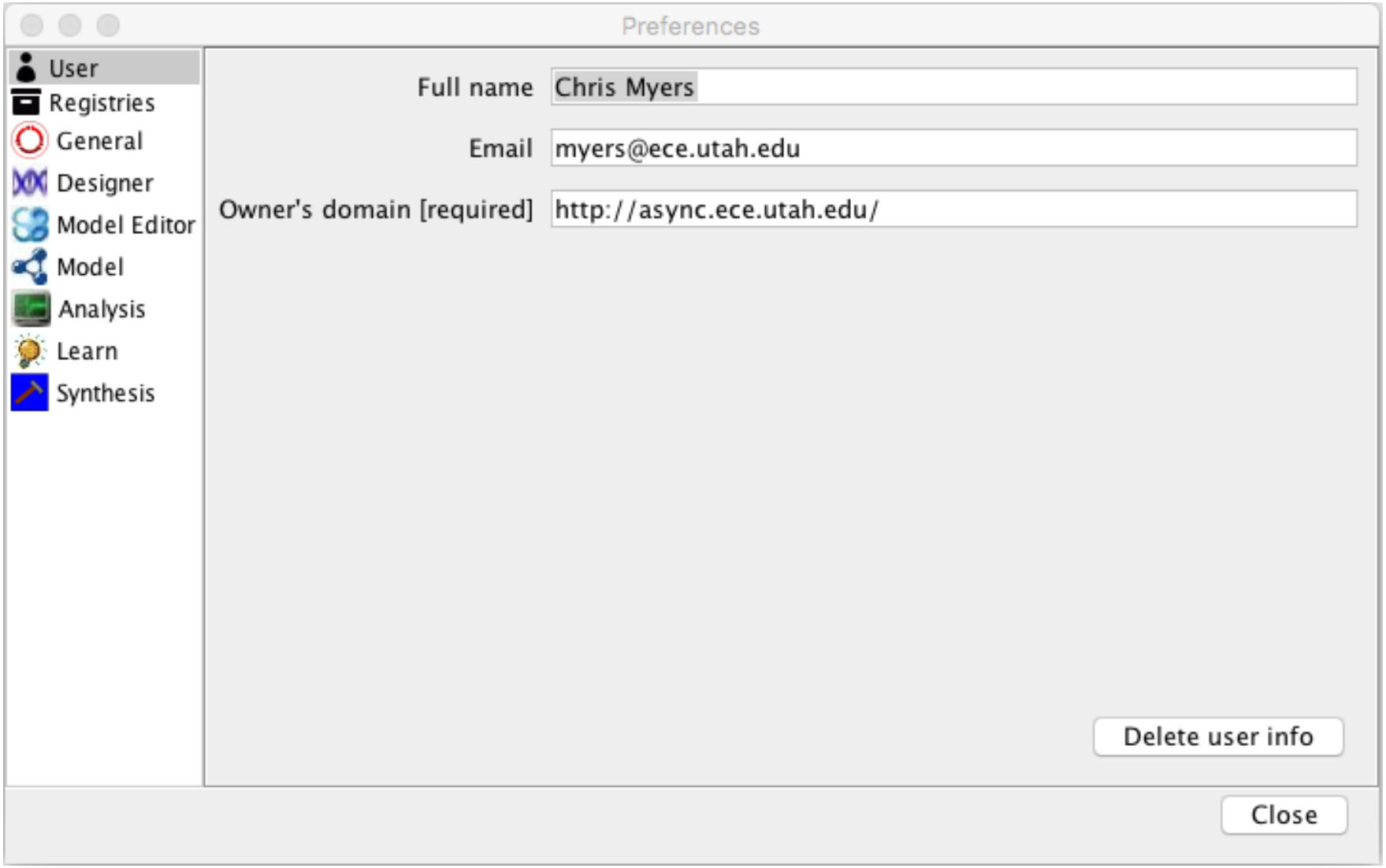
- Manual opens in browser, unfortunately a bit of out-of-date.
- Systems biology tutorial:
  - See `iBioSim_SysBio_Tutorial.pdf` found in `iBioSim` docs directory.
  - Illustrates `iBioSim`'s features using the phage λ decision circuit.
- Synthetic biology tutorial:
  - See `iBioSim_SynBio_Tutorial.pdf` found in `iBioSim` docs directory.
  - Illustrates `iBioSim`'s features using the genetic toggle switch.

Submitting Bugs and Feature Requests

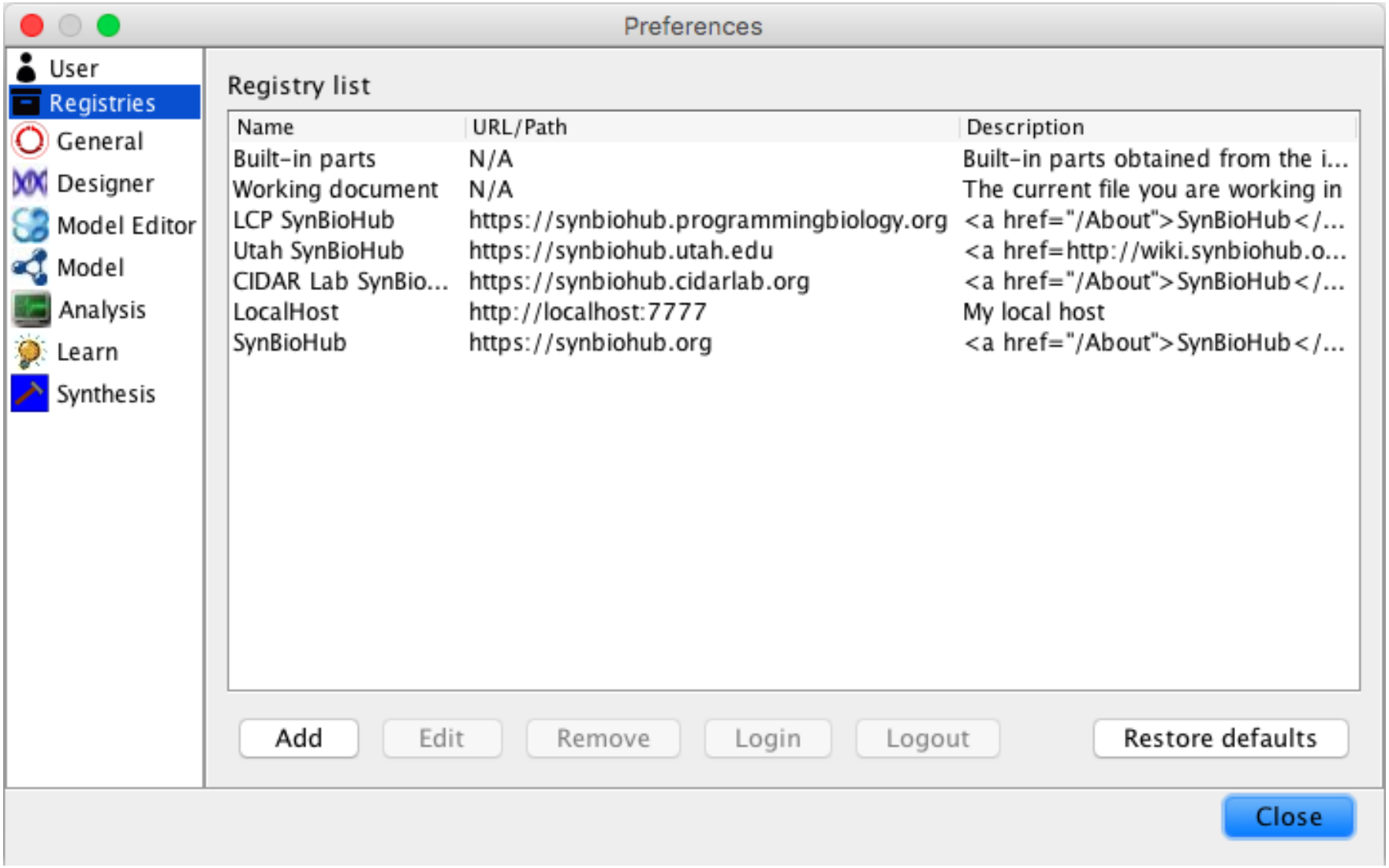


- Can also submit on github:
  `https://github.com/MyersResearchGroup/iBioSim/issues`.

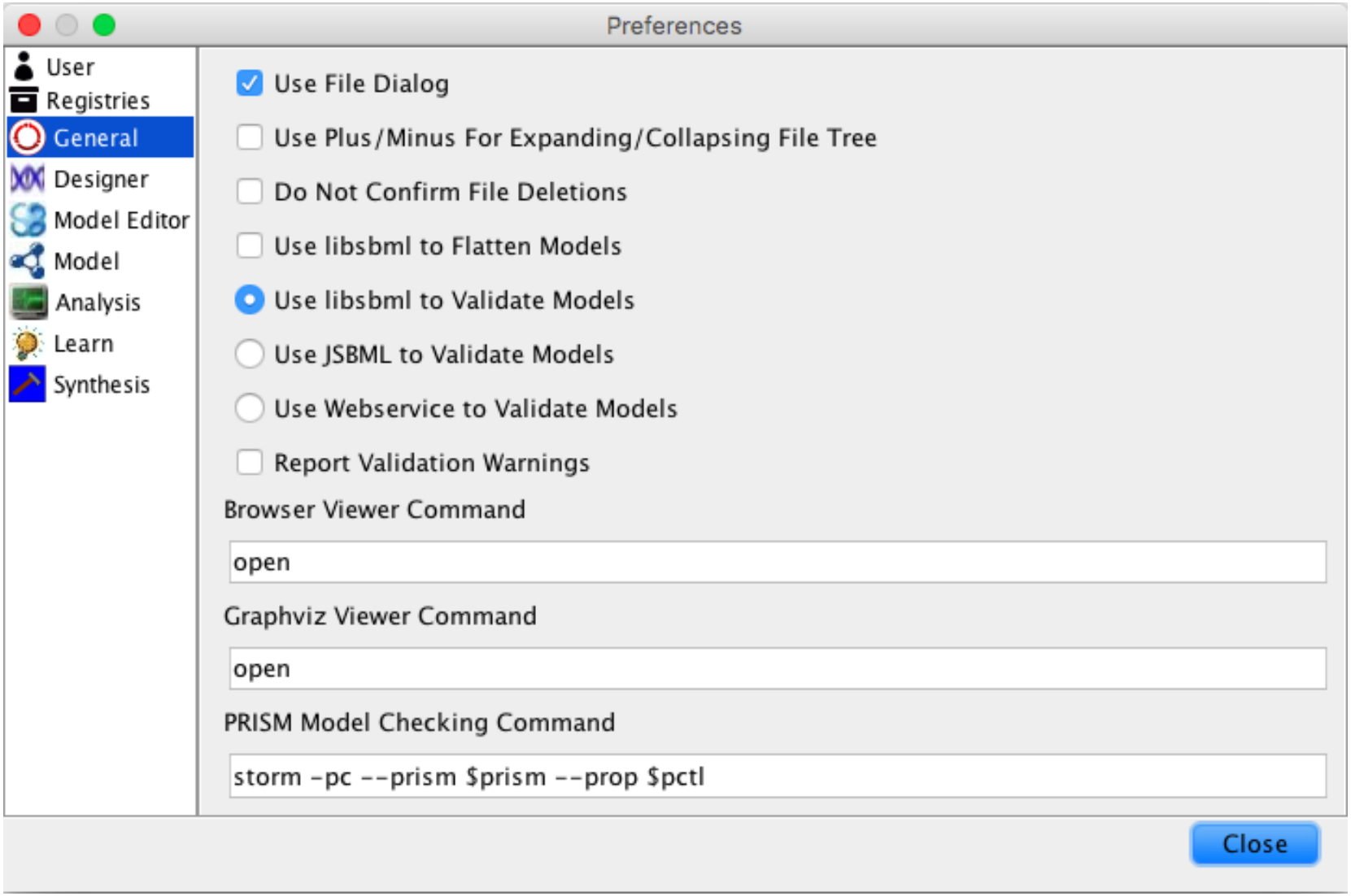
User Preferences



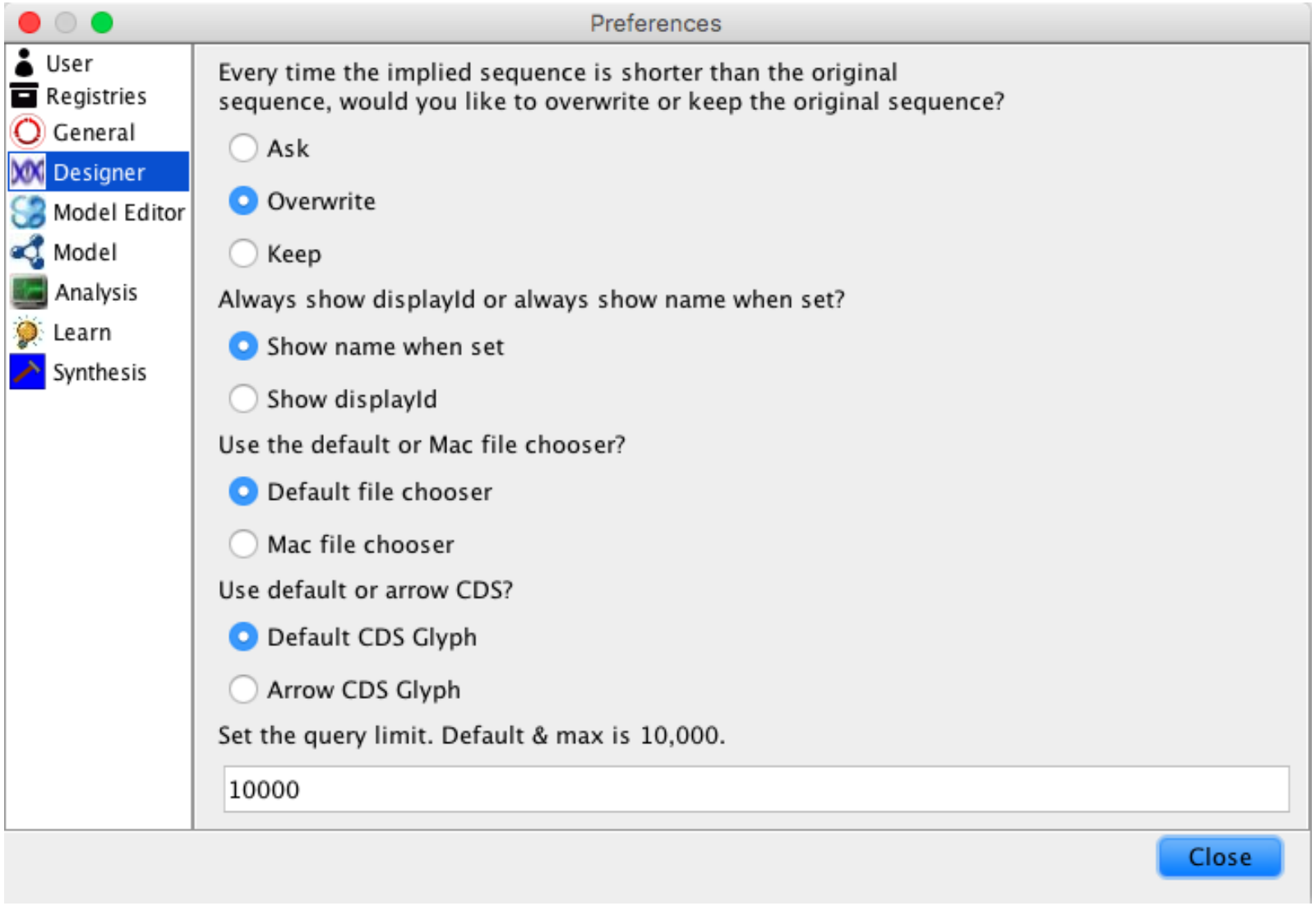
Registries Preferences



General Preferences



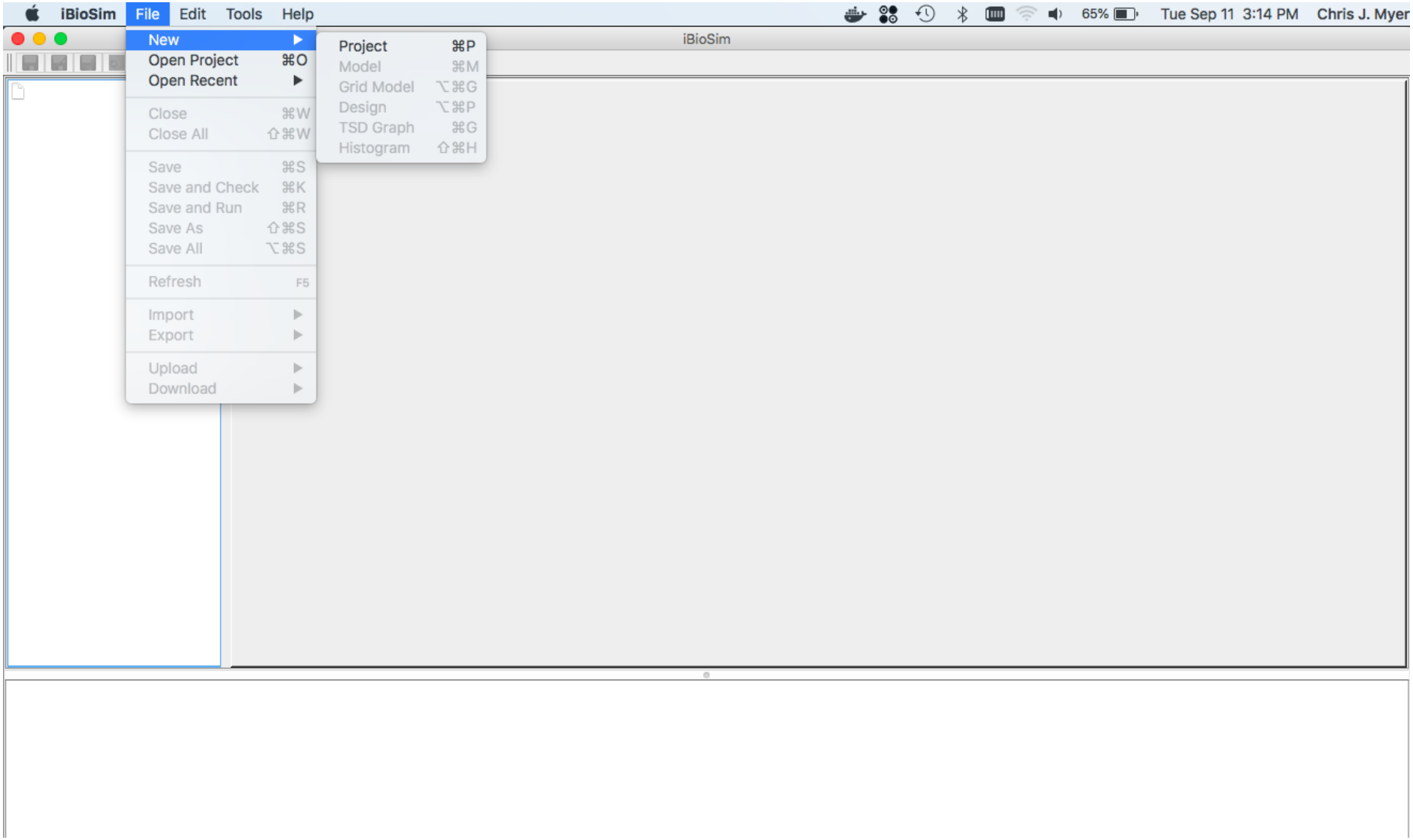
Part Editor Preferences



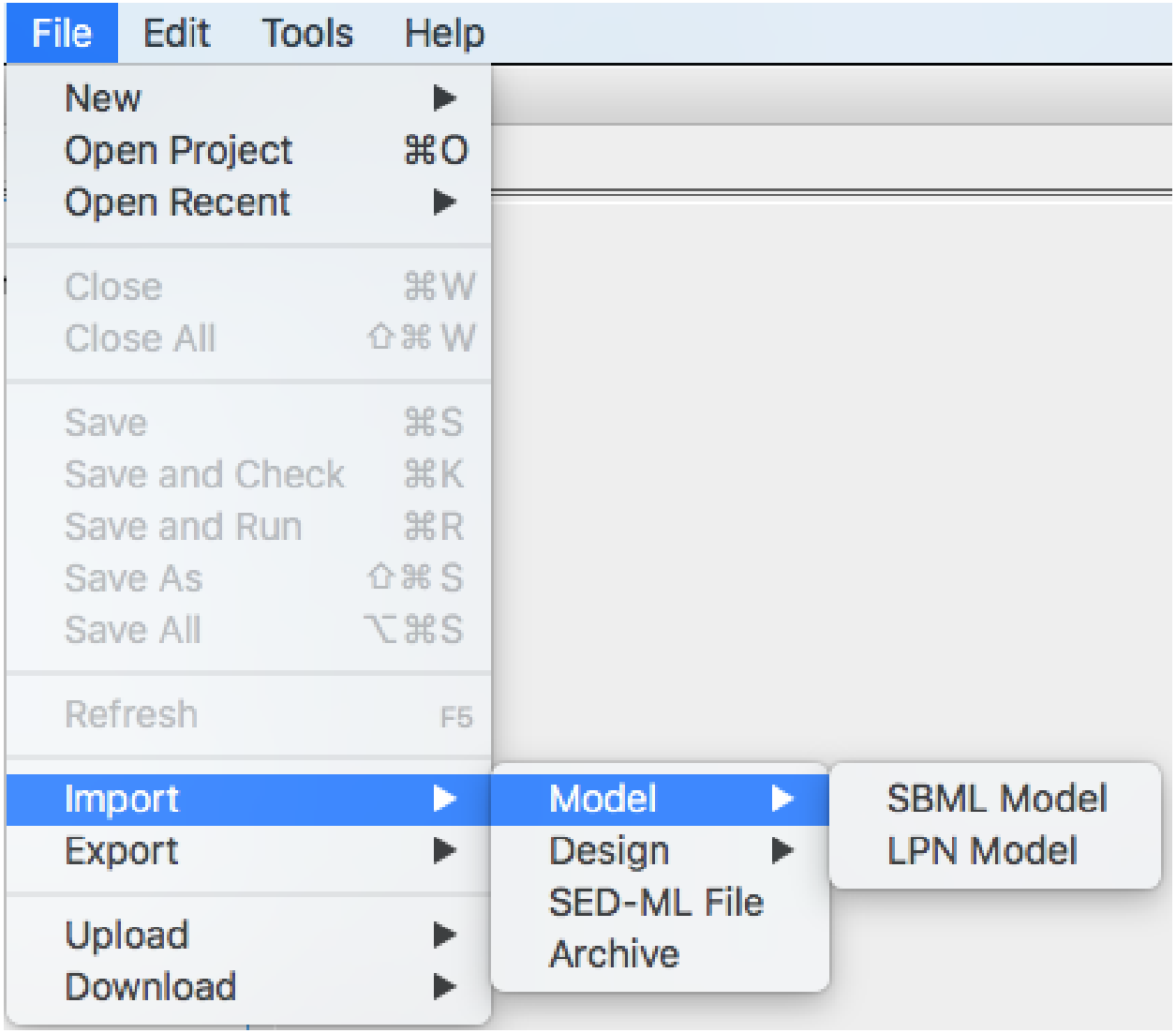
Project Management

- A *project* is a collection of models, views, and graphs.
- To get started, a new project must be created or an existing one opened.
- The ten most recently opened projects are easily accessed.
- After opening a project, new models, views, and graphs can be created.
- Models and other files can also be imported into the project.

Creating and Opening a New Project



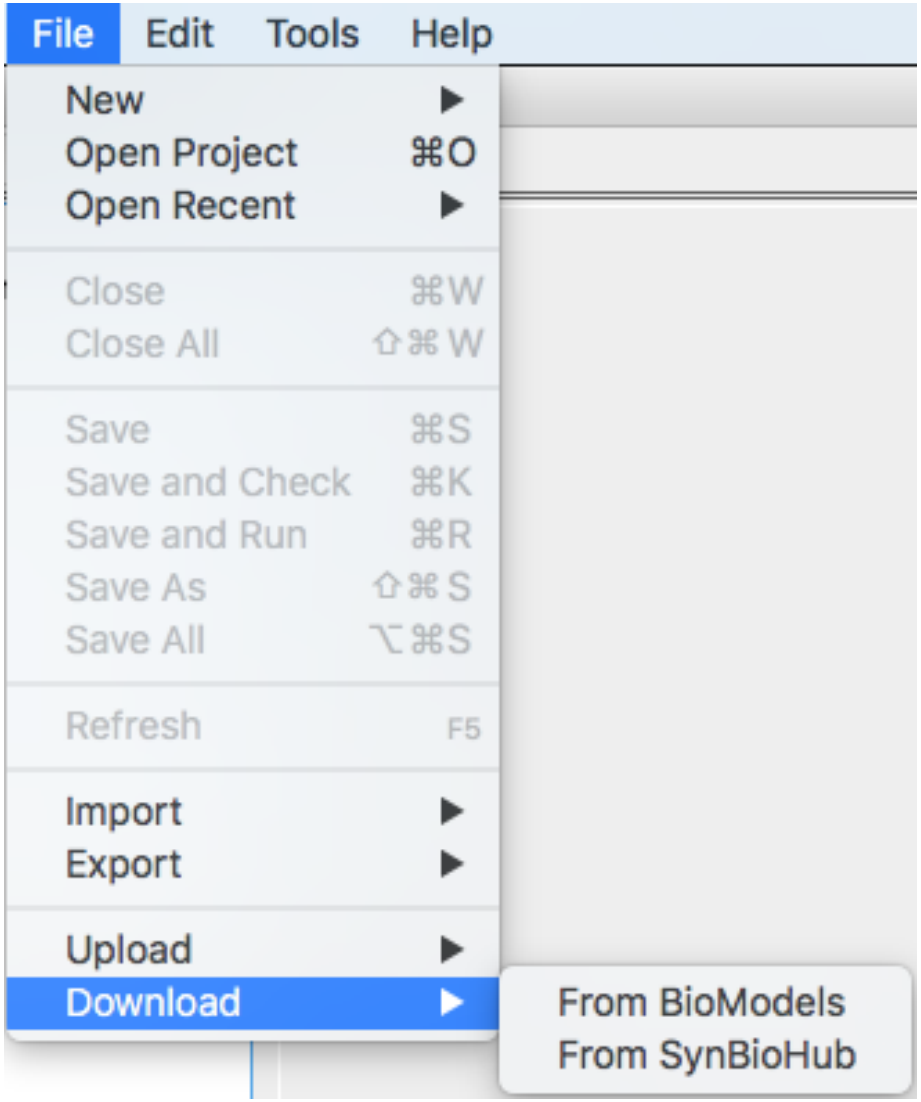
Importing Models and Other Files



Importing Models and Other Files

- You can import:
  - *Systems Biology Markup Language* (SBML) models
  - *Labeled Petri net* (LPN) models.
  - *Synthetic Biology Open Language* (SBOL) files.
  - Sequences in *GenBank* and *FASTA* formats.
  - *Simulation Experiment Description Markup Language* (SED-ML) files.
  - *COMBINE Archive* files.
- Before bringing a model into the project, it is checked for validity.
- Serious errors in the file prevent the file from being imported.
- Less serious errors and warnings are reported, and it is highly recommended that they are corrected before analysis.

Downloading Models and Archives



BioModels Database (http://biomodels.caltech.edu)

BioModels Database

Search

Advanced

BioModels Home | Models | Submit | Support | About BioModels | Contact us

BioModels Database is a repository of computational models of biological processes. Models described from literature are manually curated and enriched with cross-references. All models are provided in the Public Domain. More information about BioModels Database can be found in the FAQ.

Models published in the literature

Browse

Manually curated
(640 models)

Non curated
(1000 models)

Patient-derived genome-scale metabolic models
(PDGSMM)
(6753 models)

Alternative access

Gene Ontology classification

Gene Ontology tree

Advanced search

Models automatically generated from pathway resources (Path2Models)

Browse

Metabolic (112,898 models)
Non-metabolic (27,531 models)
Whole genome metabolism (2,641 models)

Alternative access

Taxonomy

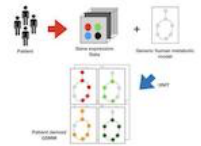
Dedicated search

Contact us | Main instance at EMBL-EBI, UK | Mirror at Caltech, USA | Model archives | Web Services

Model of the month

August, 2017

Uhlen et al. 2017 built personalised genome-scale metabolic models for 6753 patients by combining transcriptomics data of tumour samples available from TCGA with generic human metabolic model (HMR2).



Access this model of the month.

News

21 August 2017 Patient-derived genome-scale metabolic models

BioModels now hosts 6753 models derived from biological samples collected from patients.

26 June 2017 31st release of BioModels

We are extremely happy to announce the 31st release of BioModels. The resource now provides 1640 literature-based models and 143,070 models automatically generated from pathway resources. For more information, please refer to the release notes.

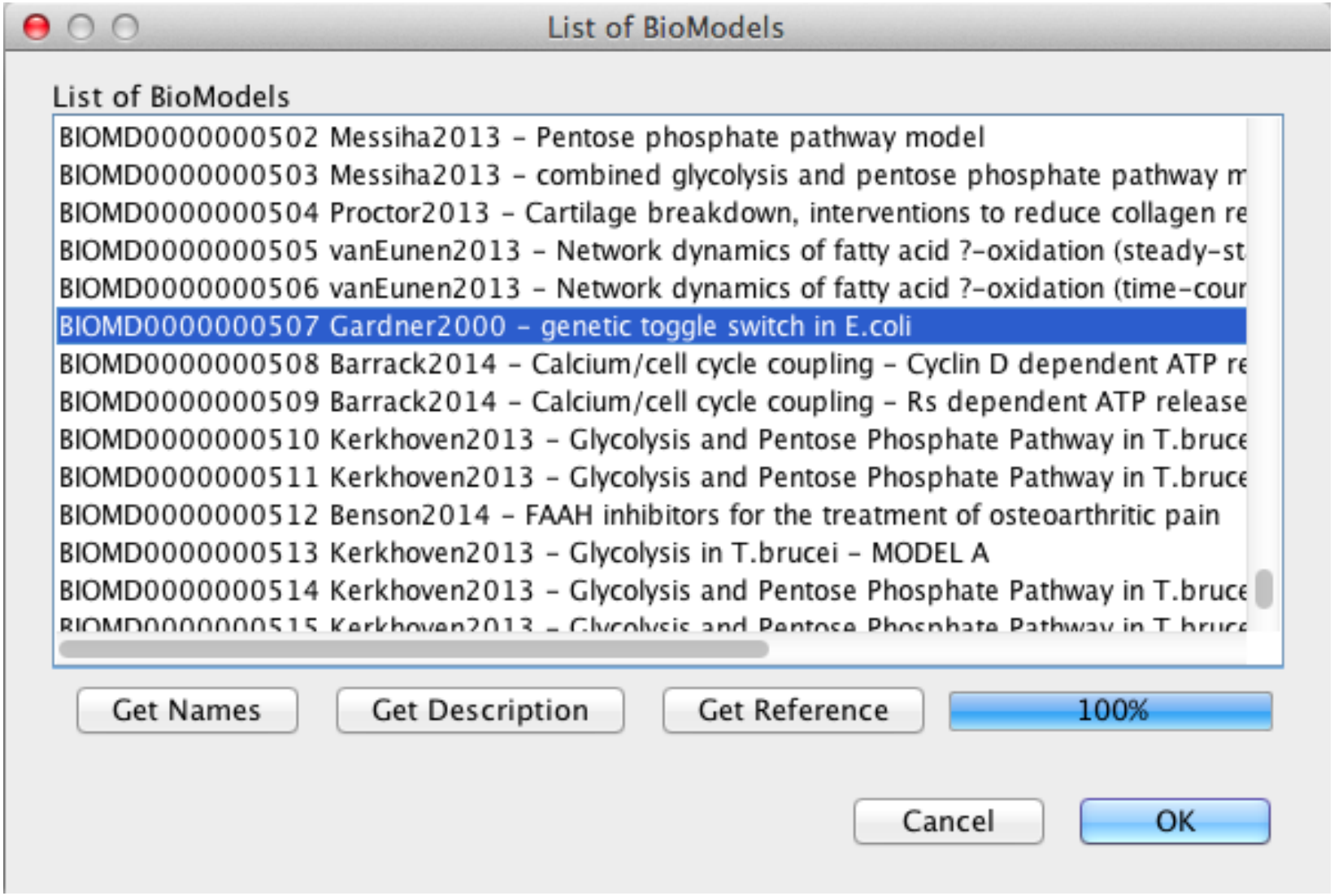
16 January 2017 Metabolic network and logical models

Archives of metabolic network and logical models are available for download.

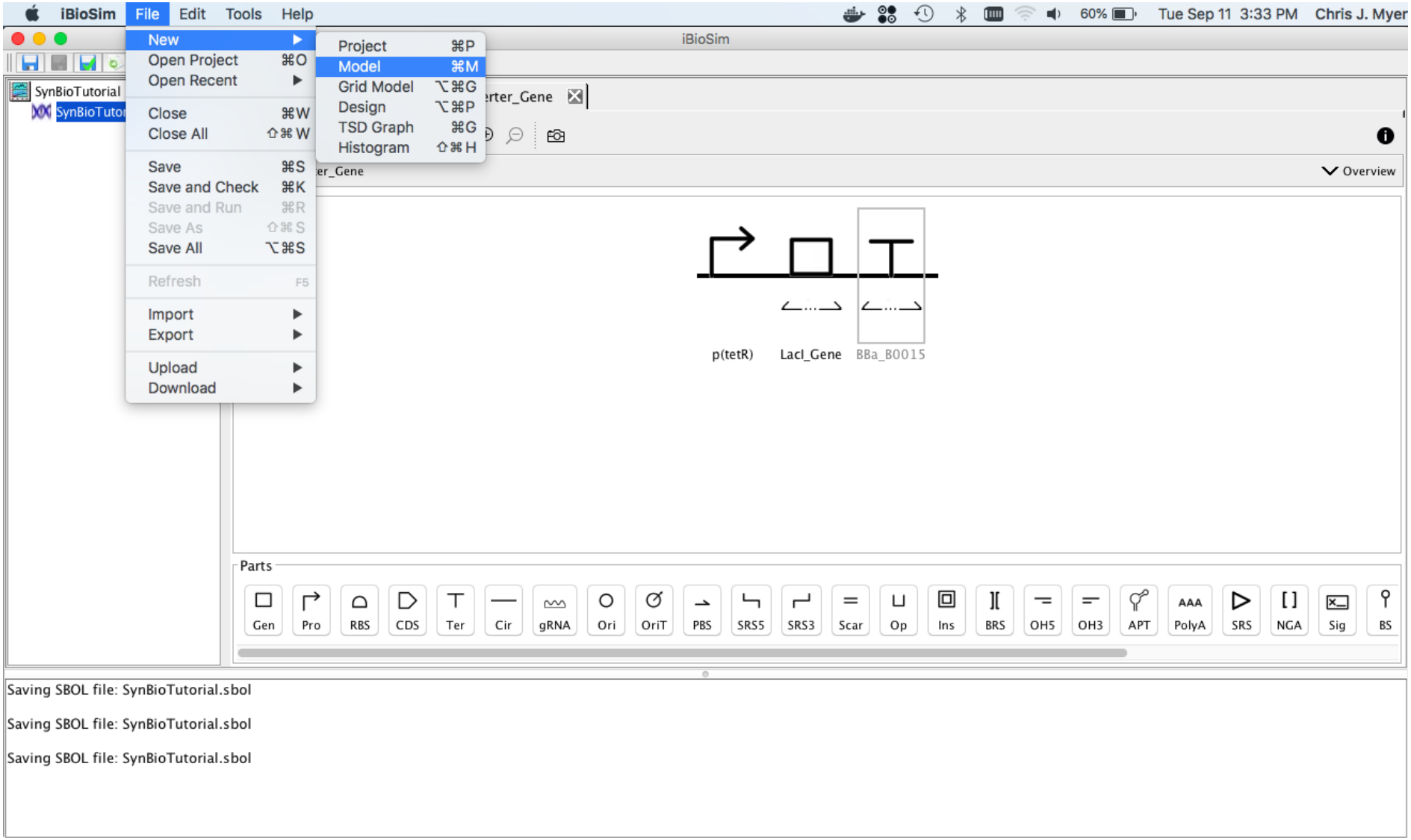
9 January 2017 Models on neurodegeneration: review article

Published in CPT:PSP Mechanistic models on neurodegenerative disease processes

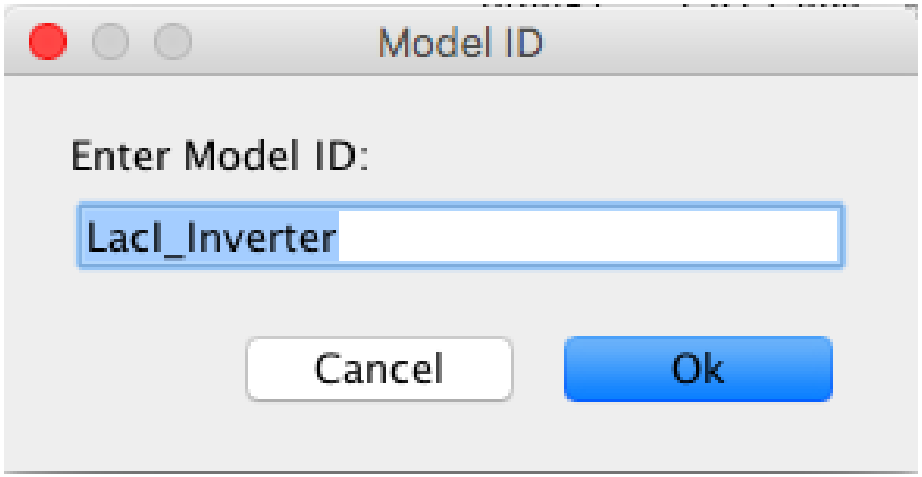
Importing from the BioModels Database



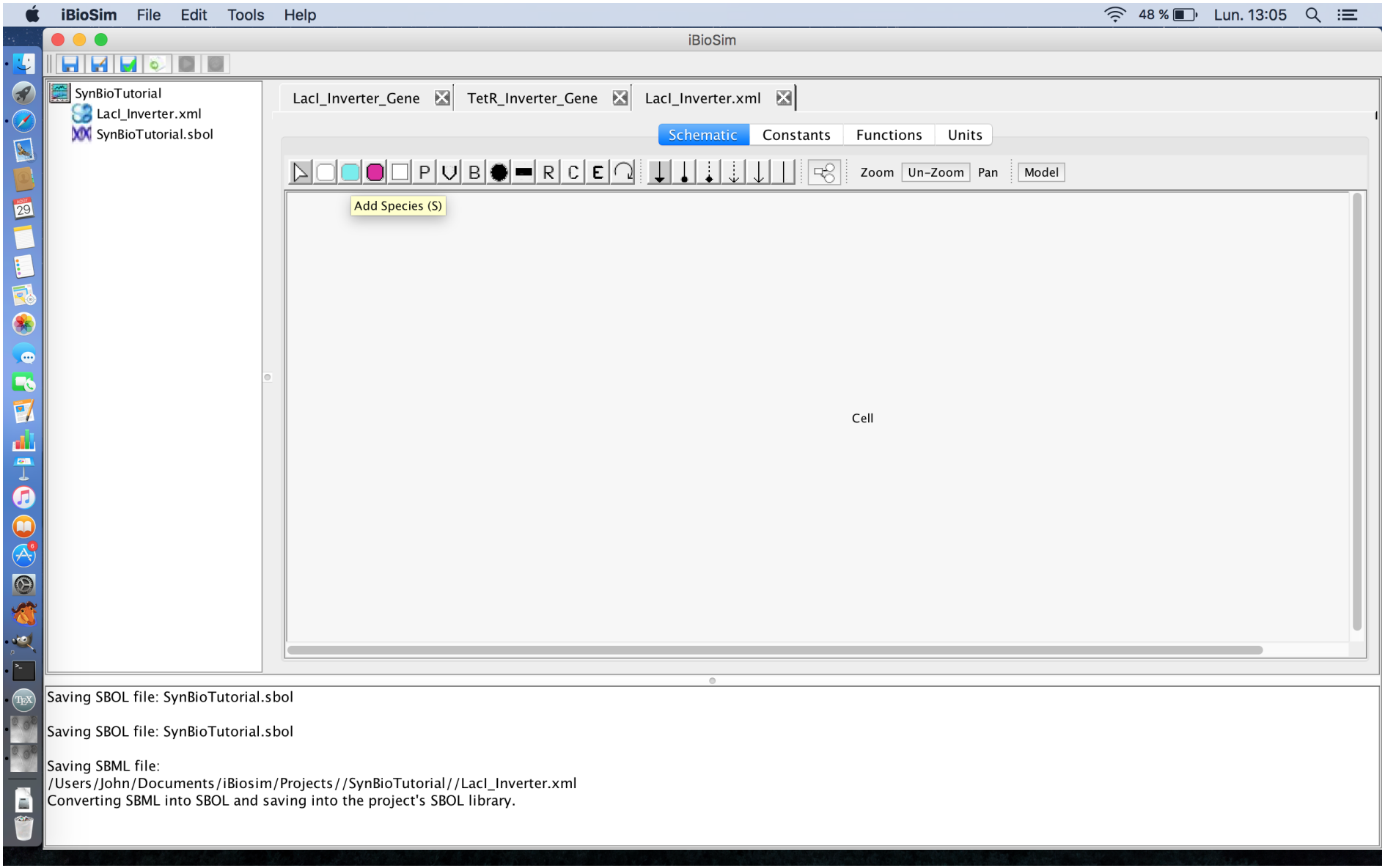
Creating Models



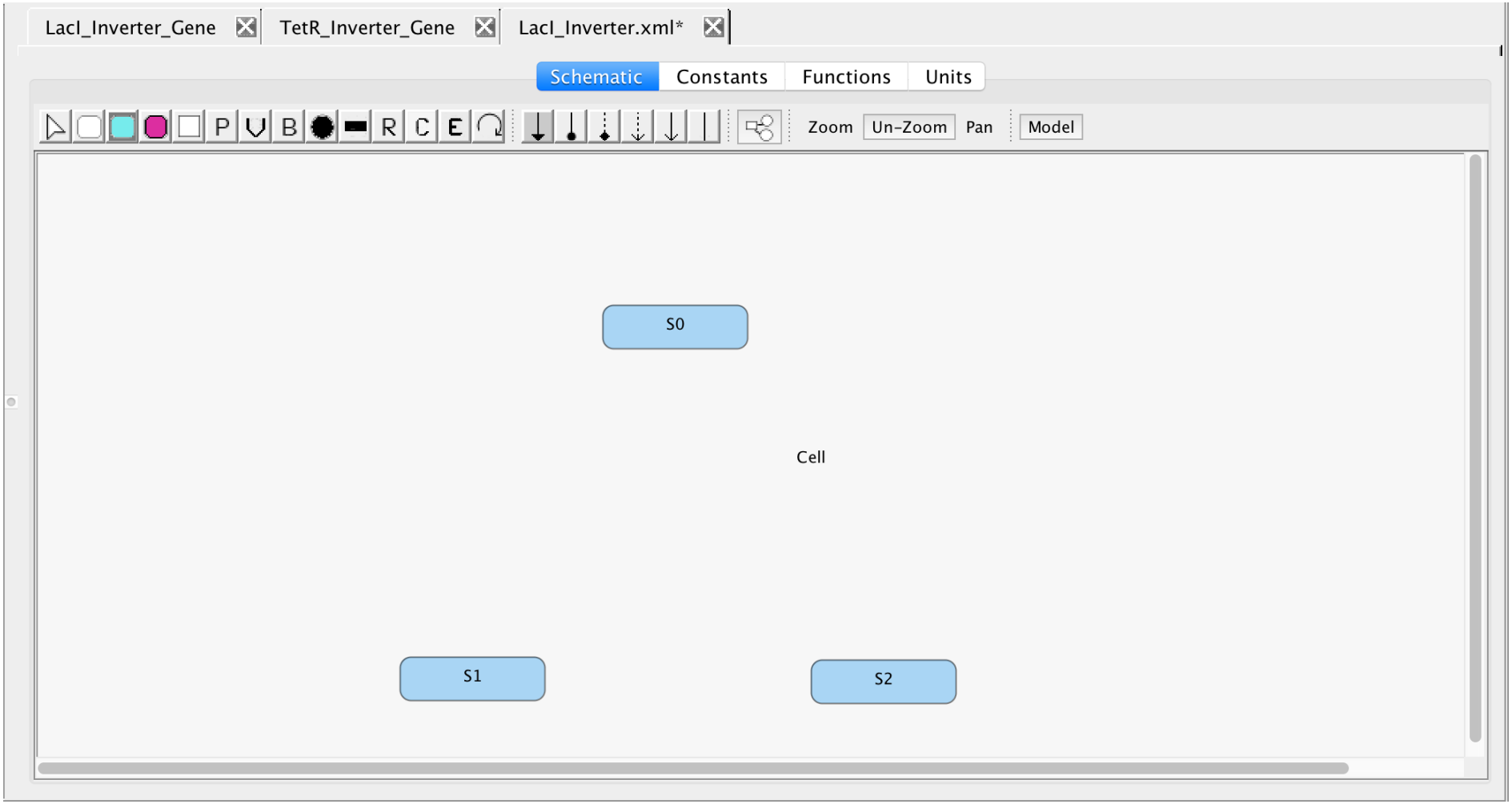
Model ID



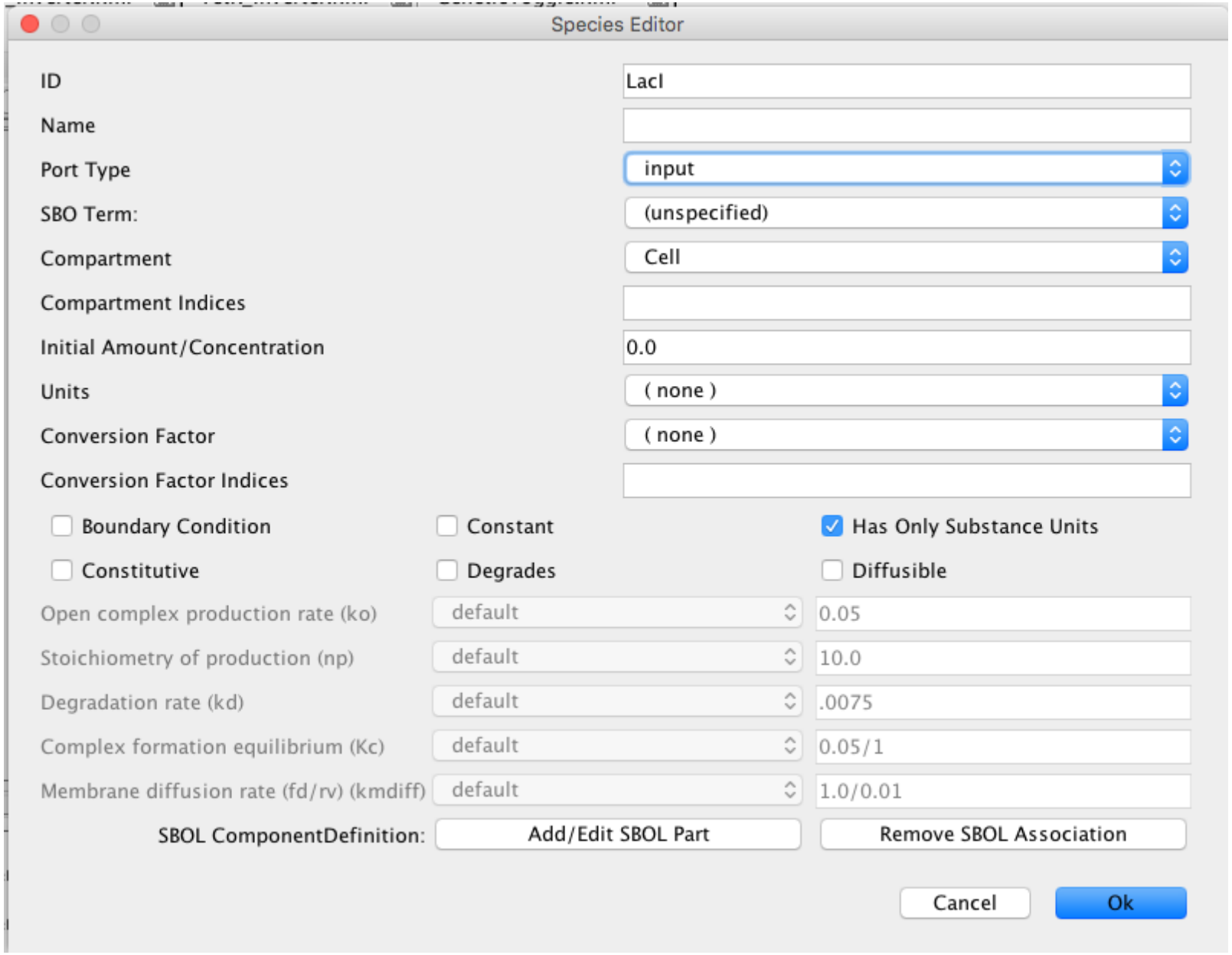
Model Editor



Species


LacI_Inverter_Gene
TetR_Inverter_Gene
LacI_Inverter.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
S0
Cell
S1
S2

Species


Species Editor
ID
LacI
Name
Port Type
input
SBO Term:
(unspecified)
Compartment
Cell
Compartment Indices
Initial Amount/Concentration
0.0
Units
(none)
Conversion Factor
(none)
Conversion Factor Indices
Boundary Condition
Constant
Has Only Substance Units
Constitutive
Degrades
Diffusible
Open complex production rate (ko)
default
0.05
Stoichiometry of production (np)
default
10.0
Degradation rate (kd)
default
.0075
Complex formation equilibrium (Kc)
default
0.05/1
Membrane diffusion rate (fd/rv) (kmdiff)
default
1.0/0.01
SBOL ComponentDefinition:
Add/Edit SBOL Part
Remove SBOL Association
Cancel
Ok

Species

- *Species* are the molecules (usually proteins) that are produced either genetically or by *chemical reactions*.
- A species has the following fields:
  - ID: composed of only alphanumeric characters and underscores (may include optional array dimensions enclosed in square brackets).
  - Name: an arbitrary string description of the species (optional).
  - Port Type: indicates how this species can be connected in hierarchical models.
    - *input*: produced outside this model.
    - *internal*: produced inside this model, not used in other models.
    - *output*: produced inside this model, can be used in other models.
  - SBO Term: term from the *physical entity representation* branch of the *system biology ontology* (`http://www.ebi.ac.uk/sbo/main/`).
  - Compartment: location of the species (default=Cell).
  - Compartment Indicies: used to reference compartment when an array.

Species (cont)

- Initial amount/concentration (ns): initial value of the amount or concentration (denoted [number]) of the species.
- Units: the units for the amount/concentration (default=none).
- Conversion Factor: constant global parameter used to convert species' units into *extent units* (i.e., units of change due to reactions).
- Conversion Factor Indicies: used to reference parameter when an array.

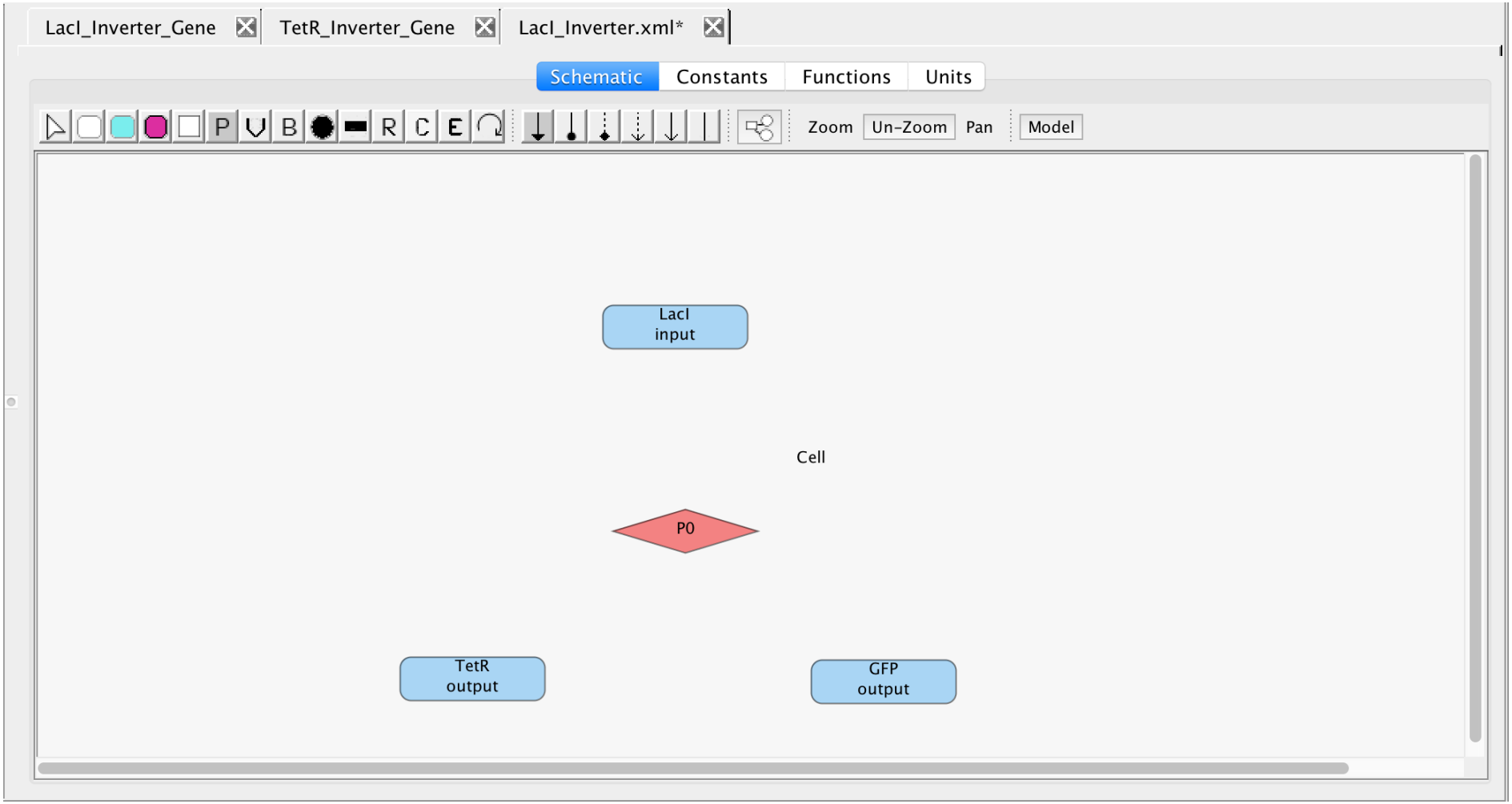
Species (cont)

- Boundary Condition: amount/concentration cannot be changed by reactions (default=false).
- Constant: amount/concentration is constant (default=false).
- Has Only Substance Units: use amounts in mathematical equations (default=true).
- Constitutive: creates a default production reaction.
- Degrades: creates a default degradation reaction.
- Diffusible: species can diffuse in multi-compartment and grid models.

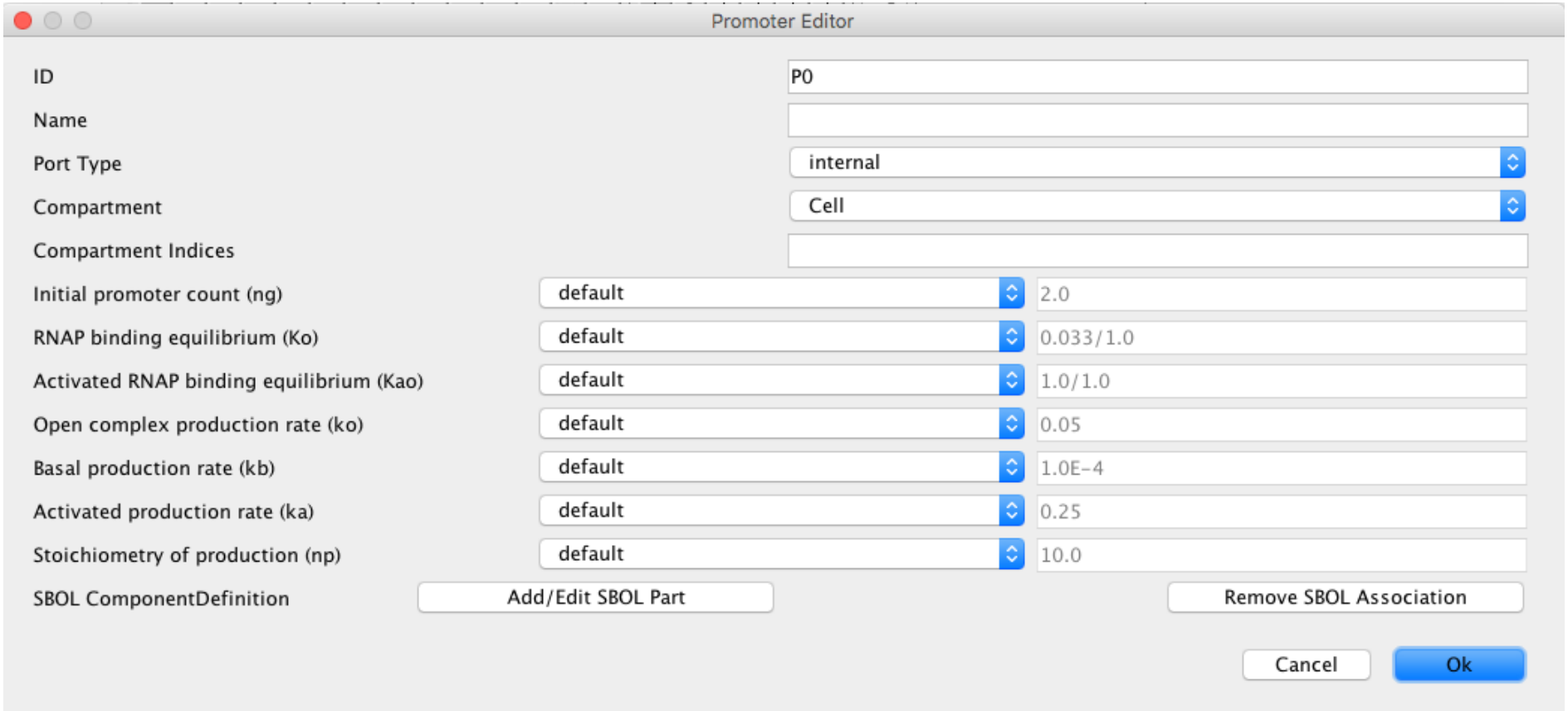
Species (cont)

- Open complex production rate (ko) for constitutive reaction.
- Stoichiometry of production (np) for constitutive reaction.
- Degradation rate (kd) for degradation reaction.
- Complex formation equilibrium (Kc): can be specified as a forward and reverse rate constant using the $\langle$ forward rate $\rangle / \langle$ reverse rate $\rangle$ form.
- Membrane diffusion rate (fd/rv) (kmdiff)
- Associate SBOL button used to assign SBOL to this species.

Promoters


LacI_Inverter_Gene
TetR_Inverter_Gene
LacI_Inverter.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
LacI
input
Cell
P0
TetR
output
GFP
output

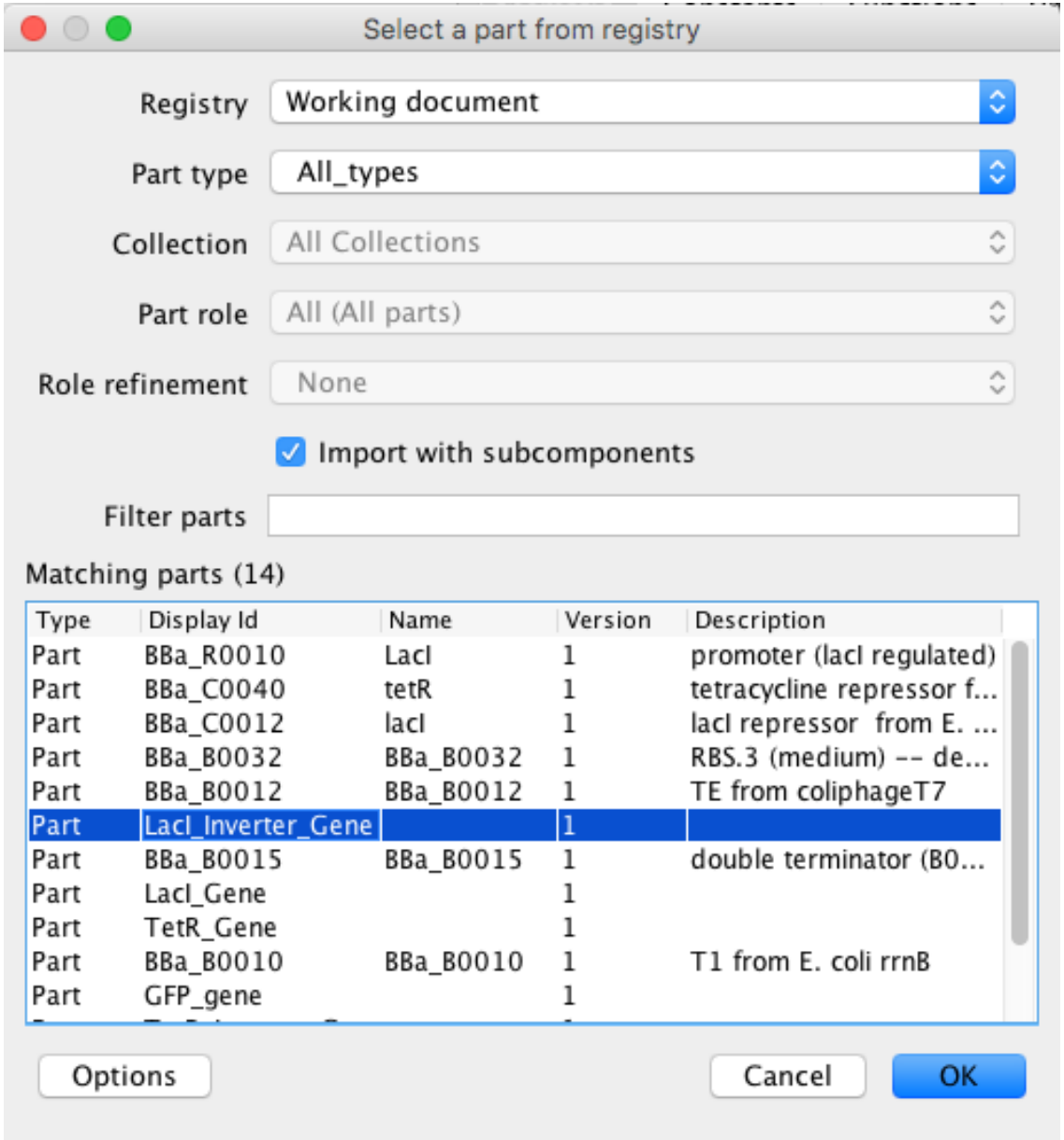
Promoters


Promoter Editor
ID
P0
Name
Port Type
internal
Compartment
Cell
Compartment Indices
Initial promoter count (ng)
default
2.0
RNAP binding equilibrium (Ko)
default
0.033/1.0
Activated RNAP binding equilibrium (Kao)
default
1.0/1.0
Open complex production rate (ko)
default
0.05
Basal production rate (kb)
default
1.0E-4
Activated production rate (ka)
default
0.25
Stoichiometry of production (np)
default
10.0
SBOL ComponentDefinition
Add/Edit SBOL Part
Remove SBOL Association
Cancel
Ok

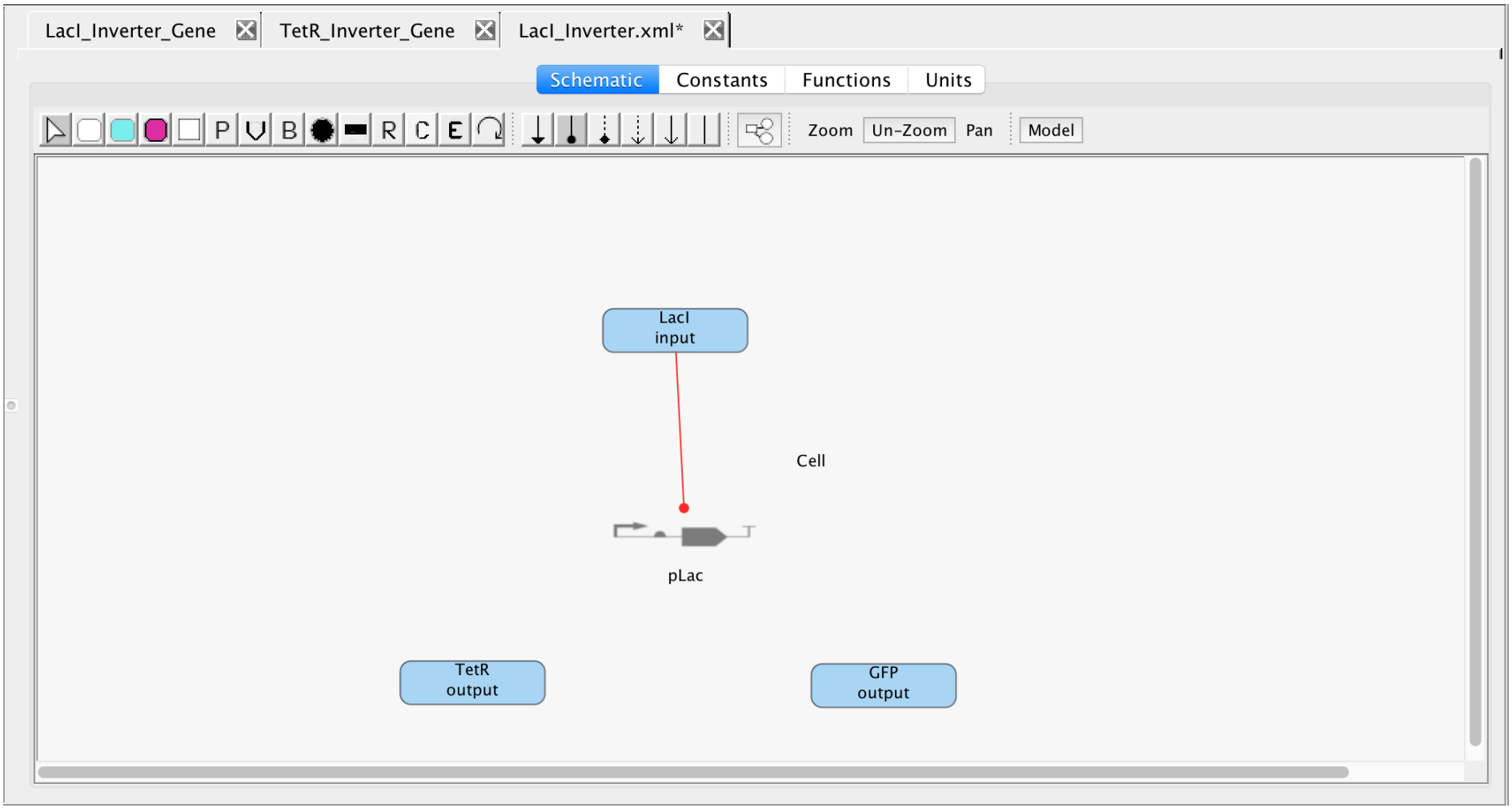
Promoters

- *Promoters* are special species which represent the region of the DNA from which transcription is initiated.
- A promoter has the following fields:
  - ID: composed only of alphanumeric characters and underscores.
  - Name: an arbitrary string description (optional).
  - Port Type: input, internal, output.
  - Compartment: location of the species (default=Cell).
  - Compartment Indicies: used to reference compartment when an array.
  - Initial promoter count (ng)
  - RNAP binding equilibrium (Ko)
  - Activated RNAP binding equilibrium (Kao)
  - Open complex production rate (ko)
  - Basal production rate (kb)
  - Activated production rate (ka)
  - Stoichiometry of production (np): average number of transcripts per mRNA.
  - Associate SBOL button used to assign SBOL to this species.

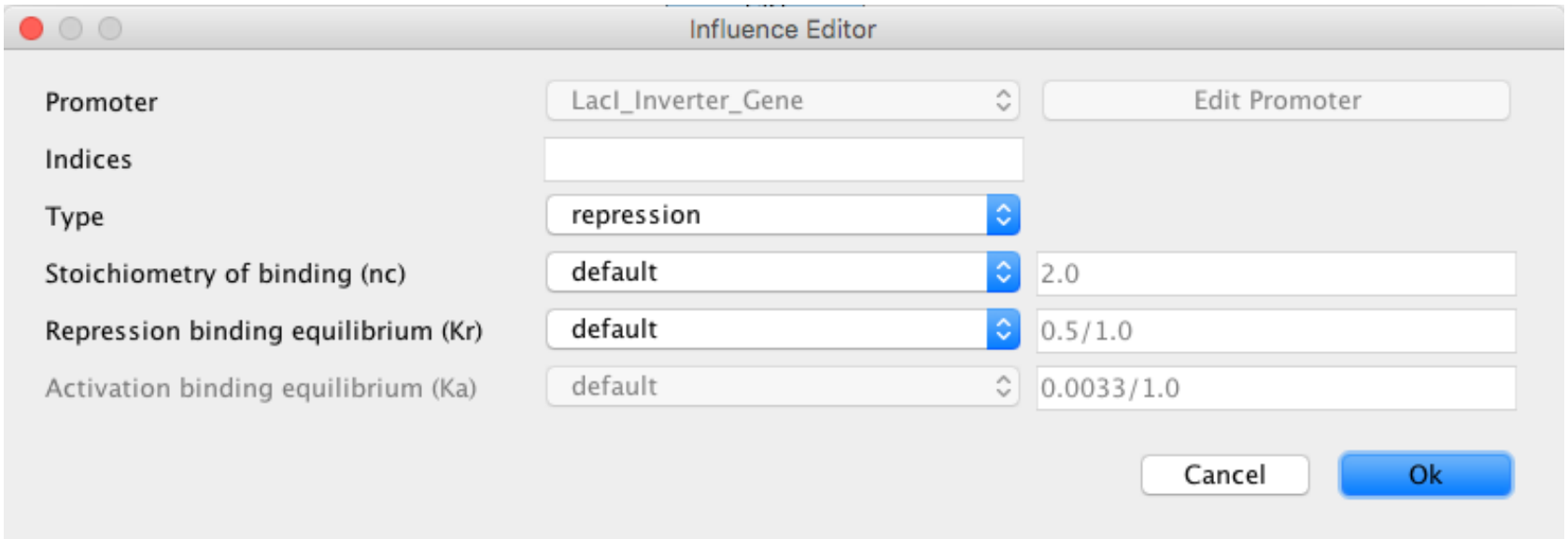
Associate SBOL



Repression


LacI_Inverter_Gene
TetR_Inverter_Gene
LacI_Inverter.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
LacI
input
Cell
pLac
TetR
output
GFP
output

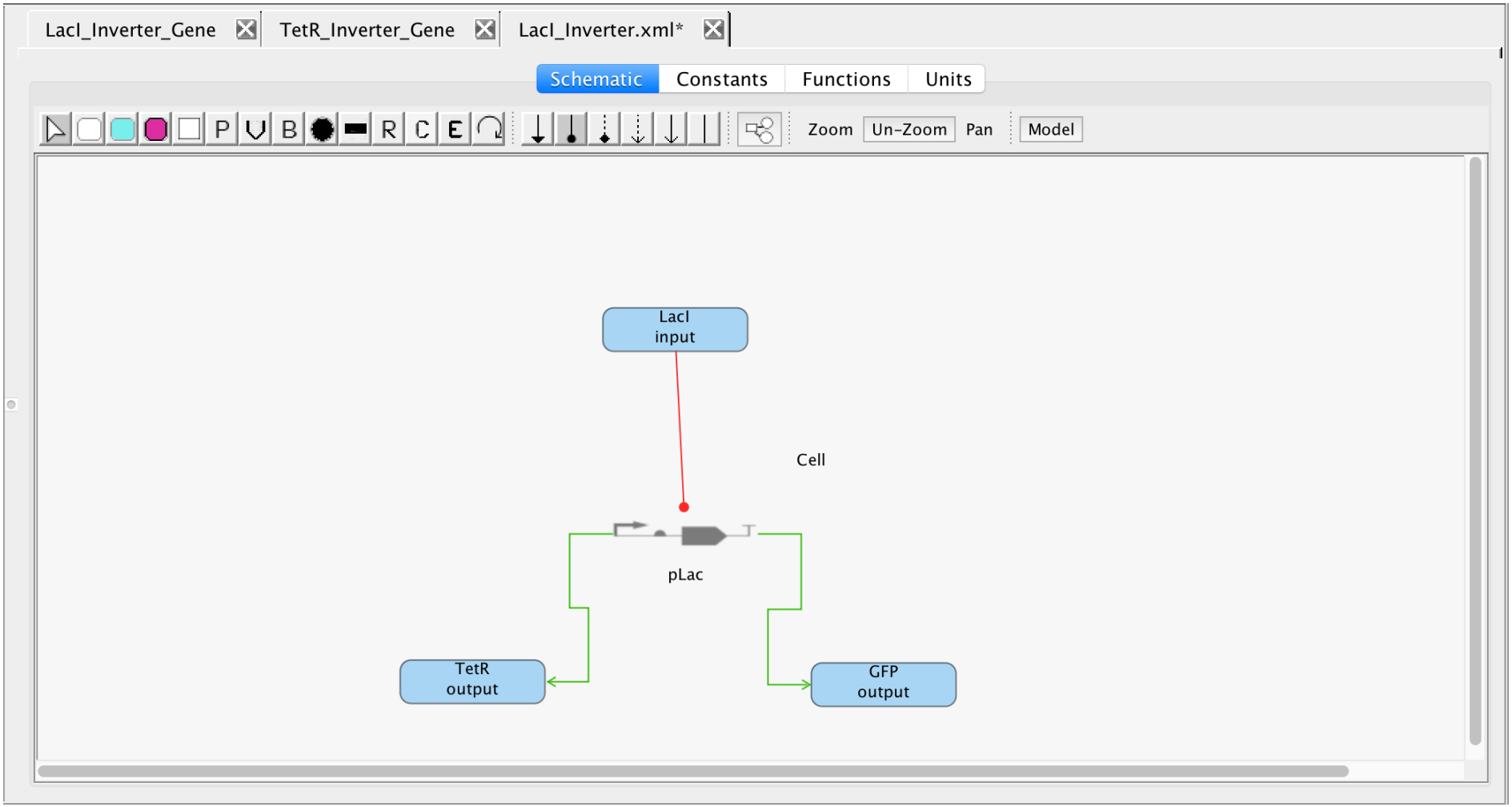
Influences


Influence Editor
Promoter
LacI_Inverter_Gene
Edit Promoter
Indices
Type
repression
Stoichiometry of binding (nc)
default
2.0
Repression binding equilibrium (Kr)
default
0.5/1.0
Activation binding equilibrium (Ka)
default
0.0033/1.0
Cancel
Ok

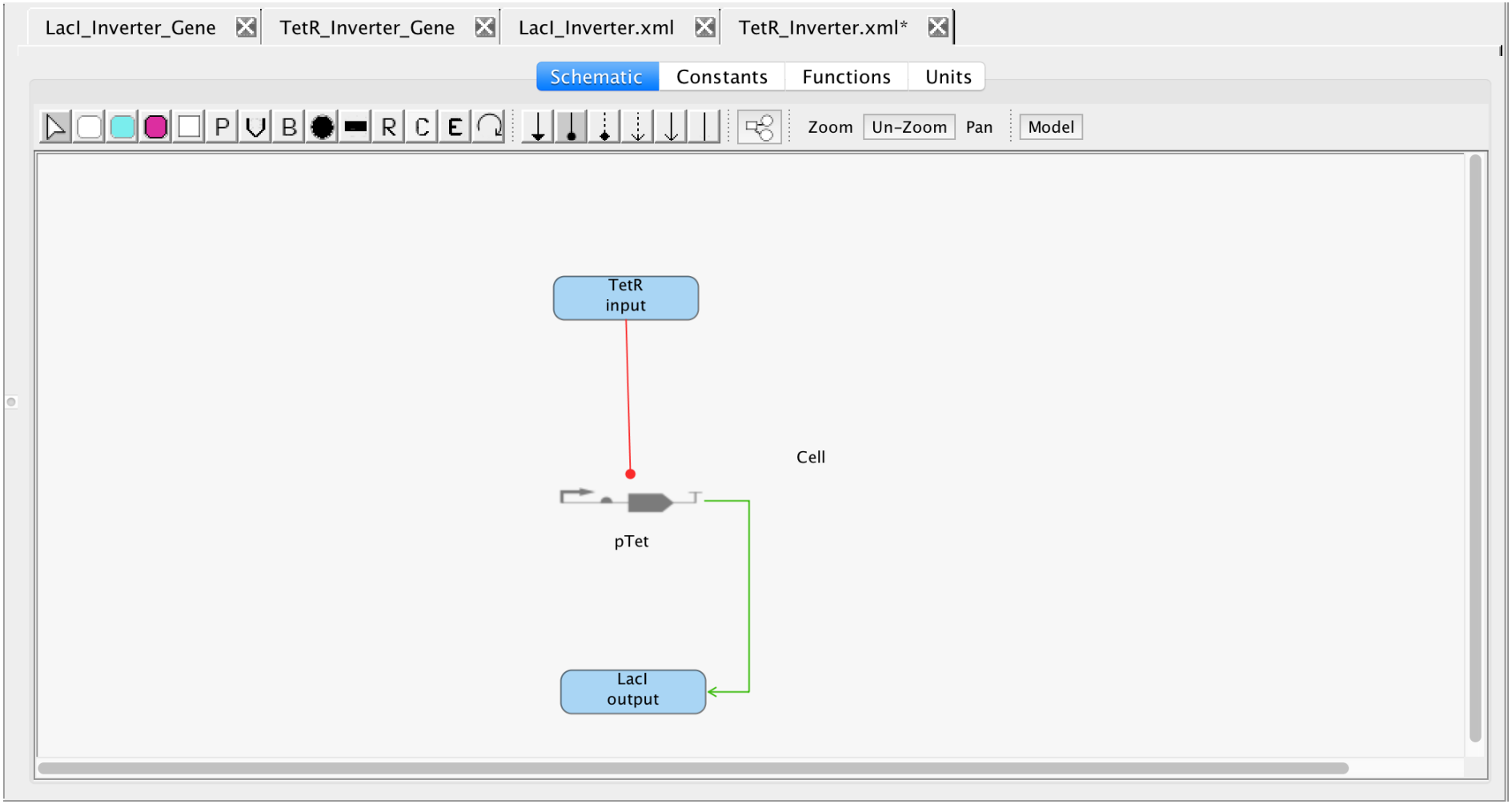
Influences

- *Influences* describe relationships between species.
- Influences have the following fields:
  - Promoter: location where influence is exerted.
  - Type: repression, activation, no influence, complex.
  - Stoichiometry of binding (nc)
  - Repression binding equilibrium (Kr)
  - Activation binding equilibrium (Ka)

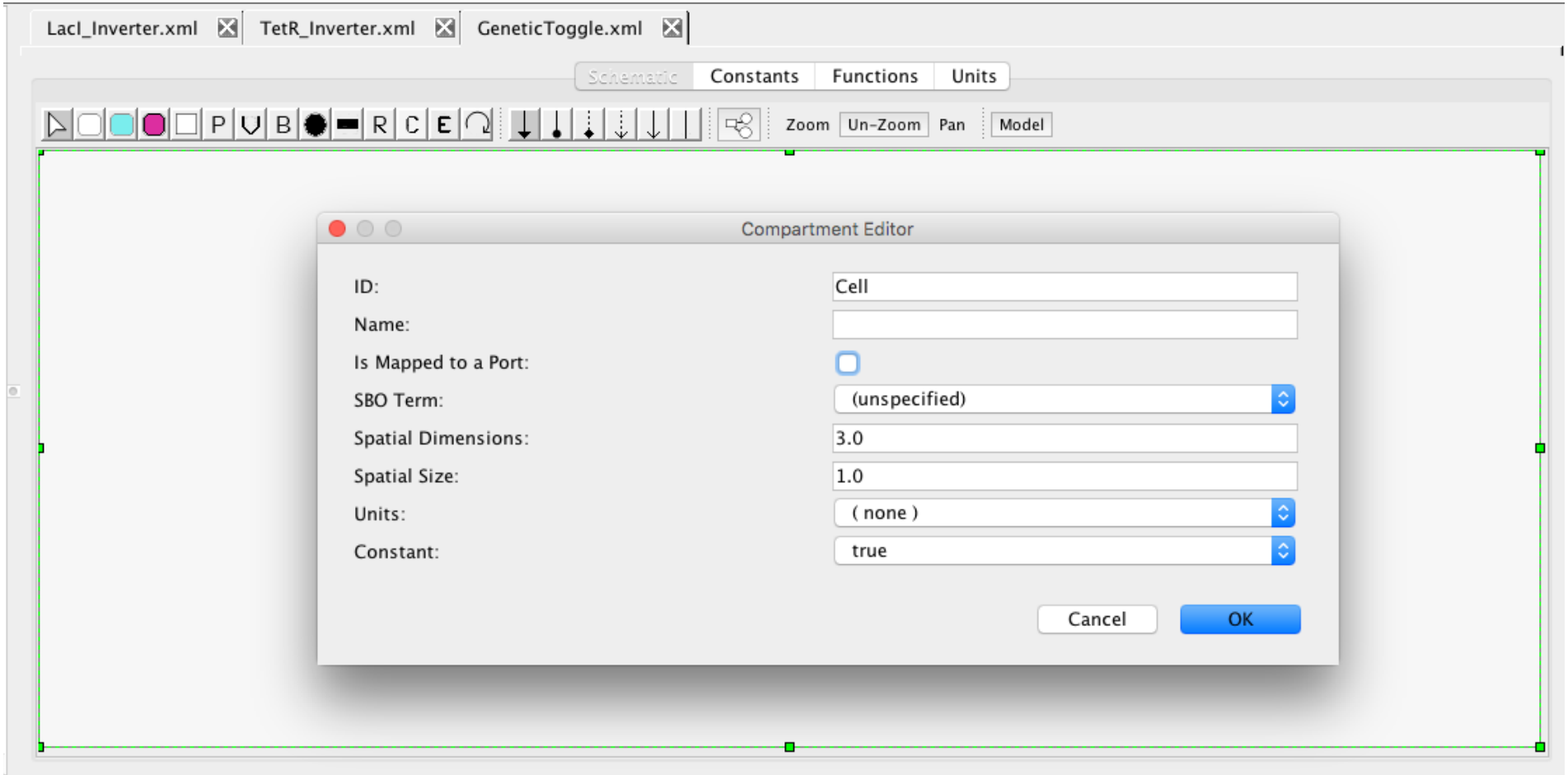
Production


LacI_Inverter_Gene
TetR_Inverter_Gene
LacI_Inverter.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
LacI
input
Cell
pLac
TetR
output
GFP
output

TetR Inverter



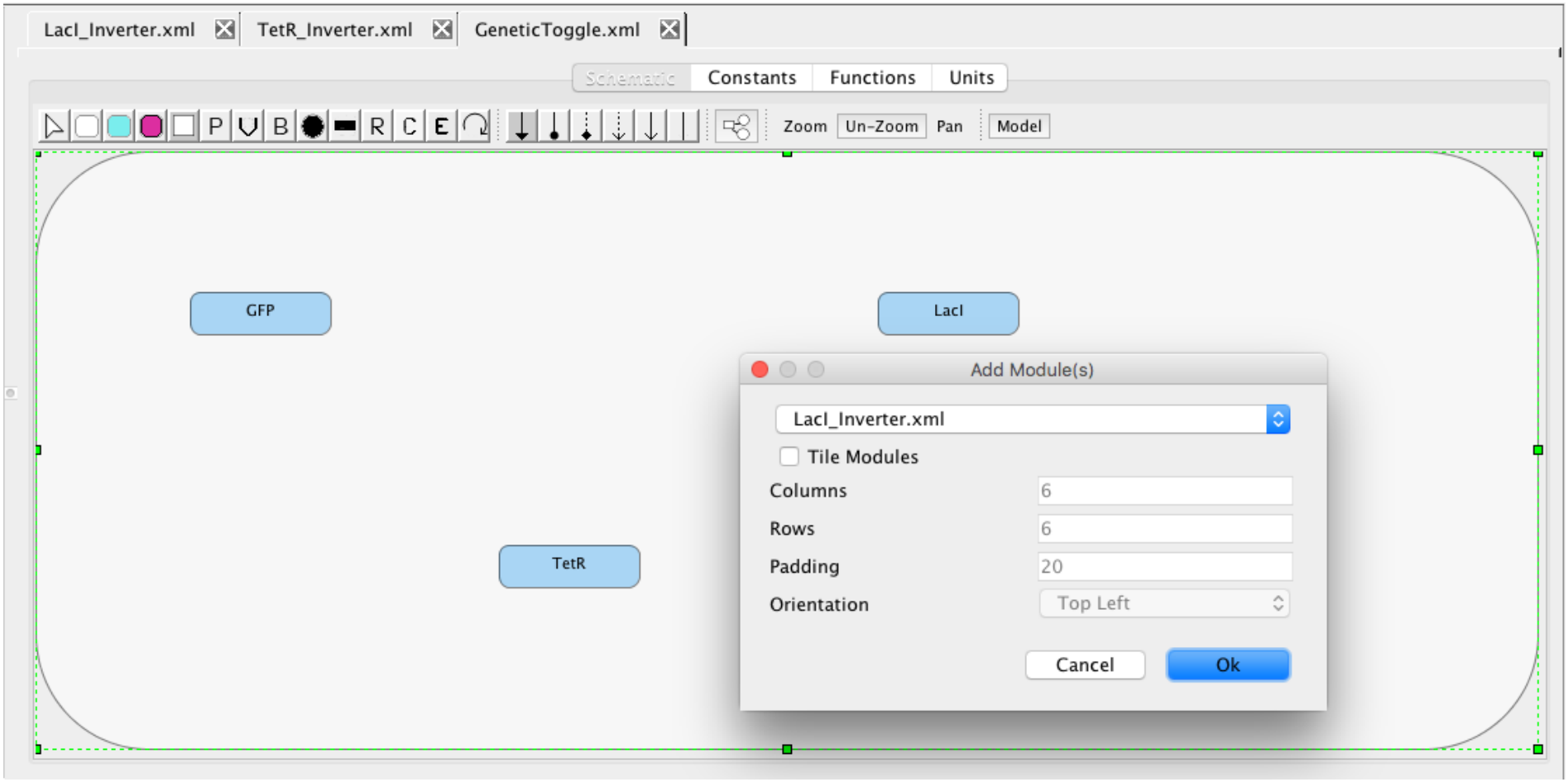
Compartments



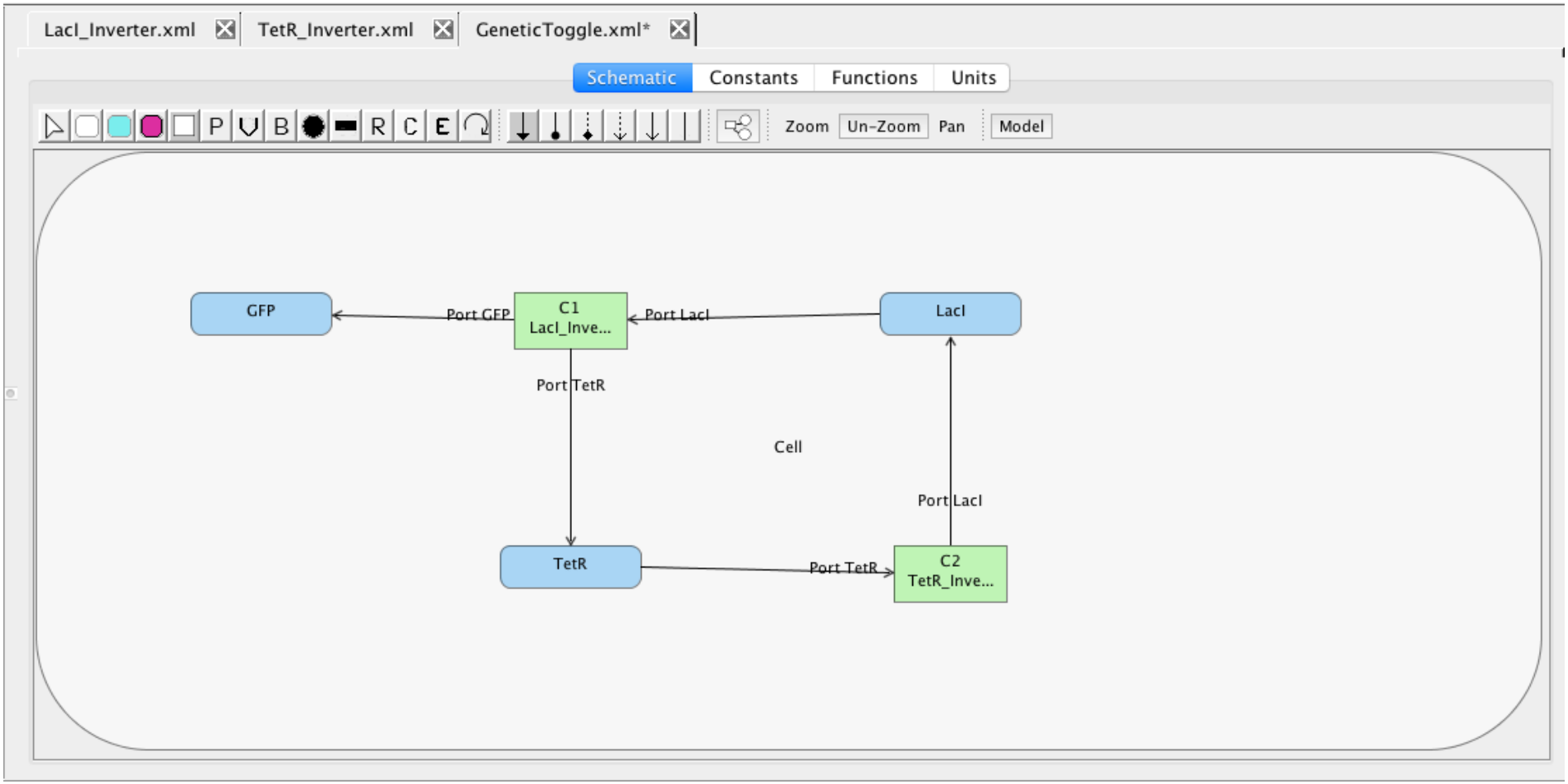
Compartments

- *Compartments* are used to specify locations where species are found.
- A new model includes a compartment named “Cell”.
- A compartment to which species or reactions have been assigned cannot be removed.
- A compartment has the following fields:
  - ID: composed of only alphanumeric characters and underscores (may include optional array dimensions enclosed in square brackets).
  - Name: an arbitrary string description (optional).
  - Is Mapped to a Port: indicates if this compartment can be replaced or deleted in hierarchical models (default=false).
  - Spatial Dimensions: number of spatial dimensions (default=3).
  - Spatial Size: initial size of the compartment (default=1.0).
  - Units: the units for the size (default=none).
  - Constant: indicates if the size is constant (default=true).

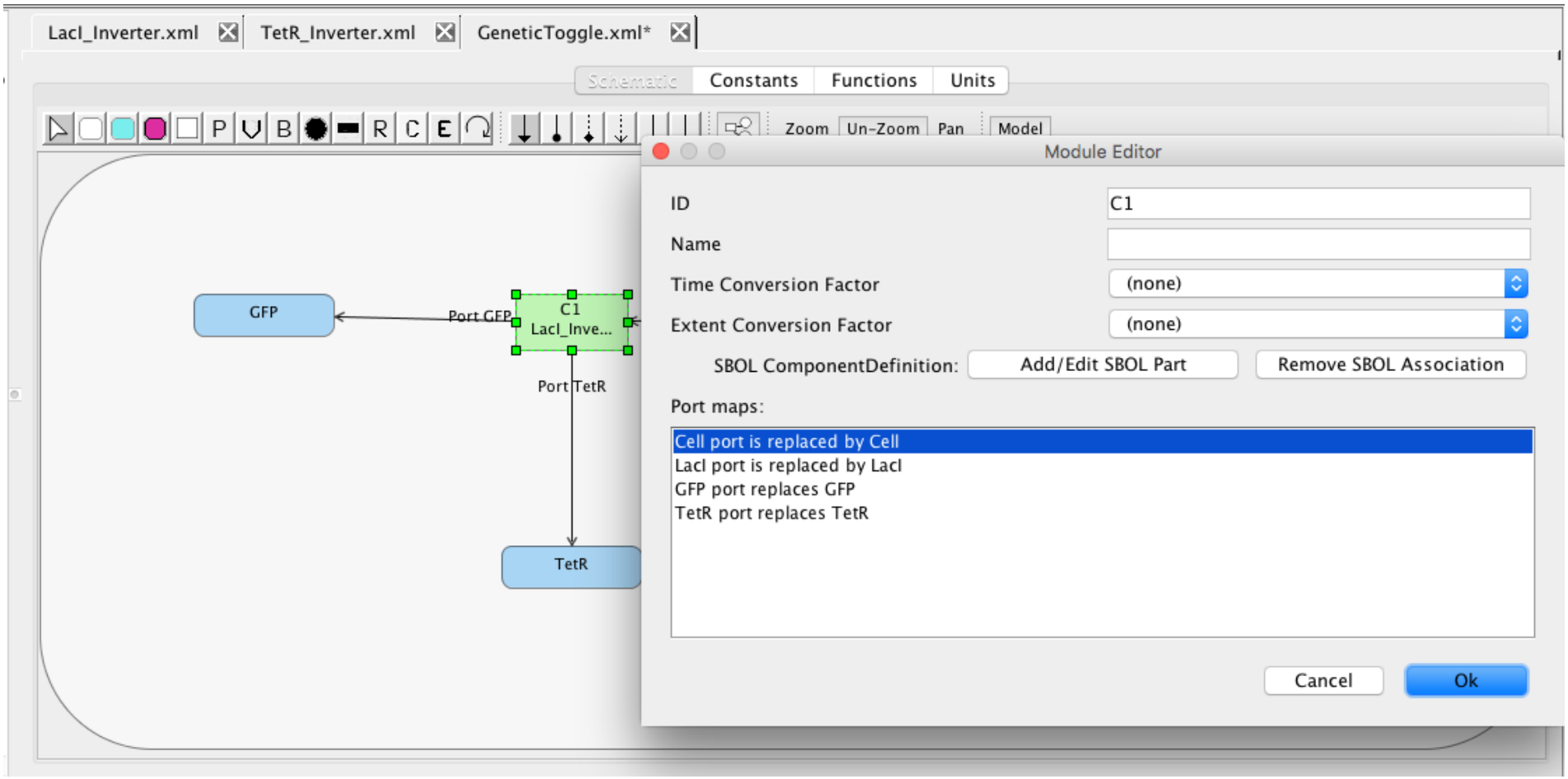
Adding Modules (SubModels)



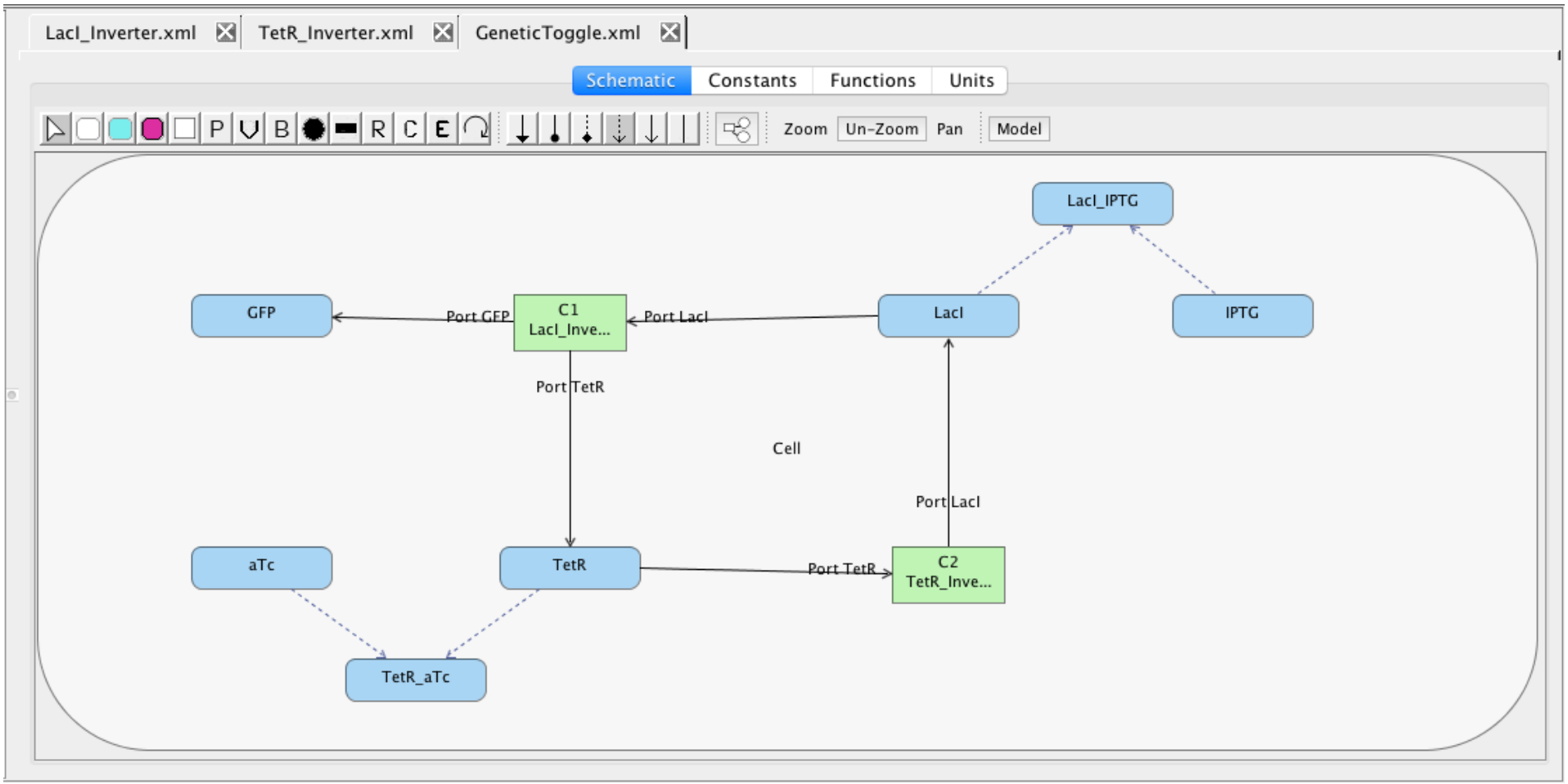
Adding Connections to Ports



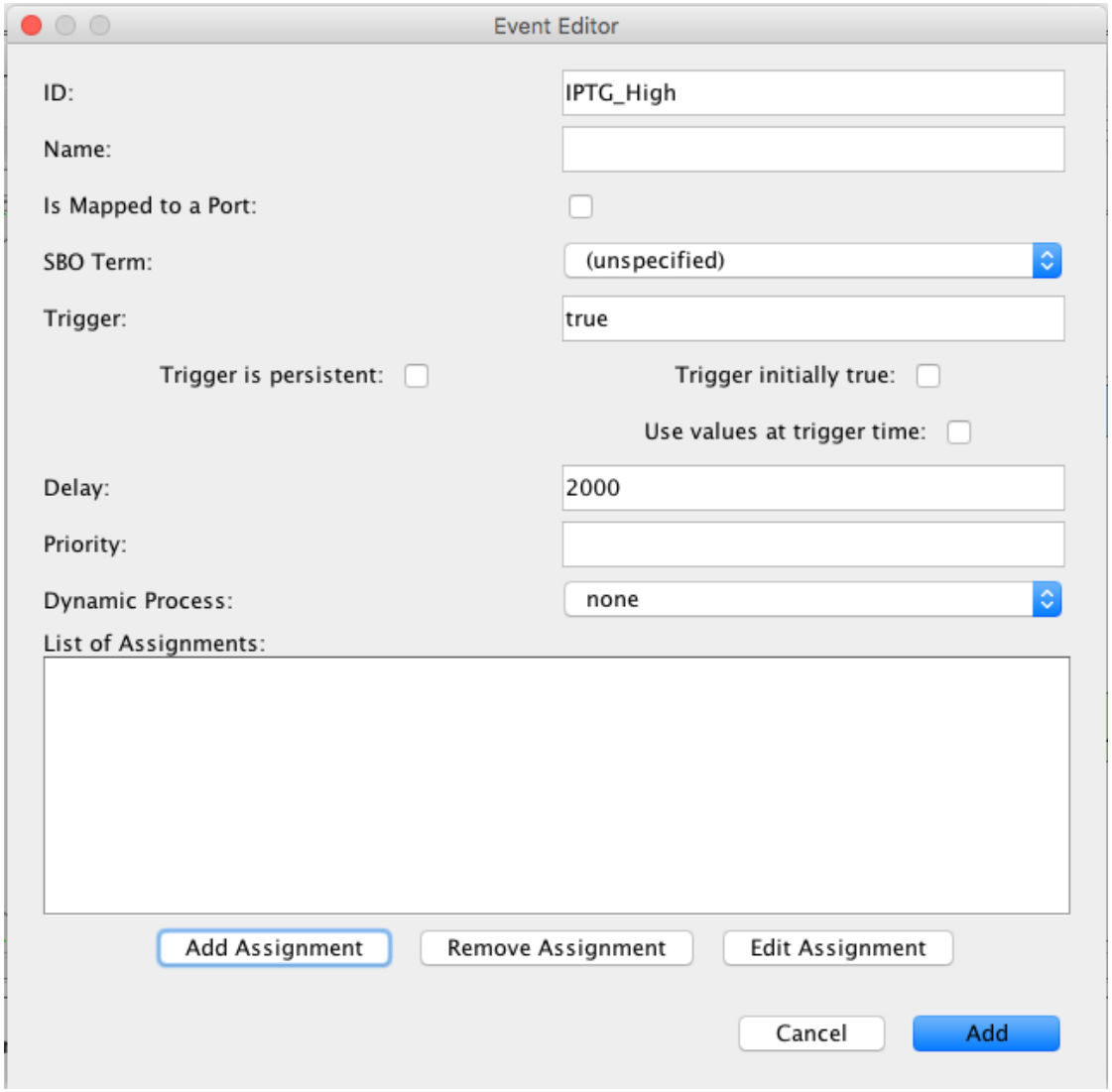
Editing Modules



Complex Formation



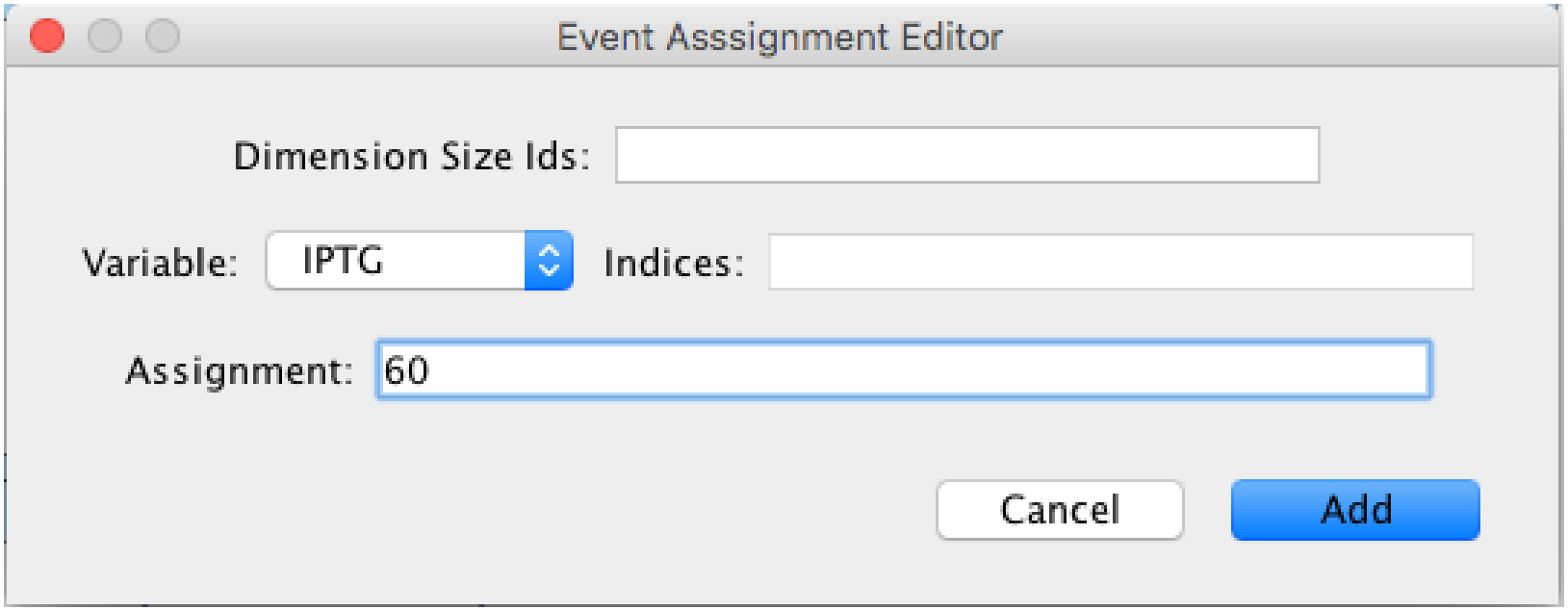
Events


Event Editor
ID:
IPTG_High
Name:
Is Mapped to a Port:
SBO Term:
(unspecified)
Trigger:
true
Trigger is persistent:
Trigger initially true:
Use values at trigger time:
Delay:
2000
Priority:
Dynamic Process:
none
List of Assignments:
Add Assignment
Remove Assignment
Edit Assignment
Cancel
Add

Events

- Events are used to specify discrete changes of compartment sizes, species amounts or concentrations, and parameter values.
  - ID: composed only of alphanumeric characters and underscores.
  - Name: an arbitrary string description (optional).
  - Is Mapped to a Port: can be replaced or deleted.
  - SBO Term: from the *occurring entity representation* branch of SBO.
  - Trigger: when changes from false to true indicates that event is triggered.
  - Trigger is persistent: indicates the behavior when a trigger expression becomes false before an event is executed (default=false).
  - Trigger is initially true: value of trigger before time 0.
  - Use values at trigger time: indicates if values for the event assignments should be calculated at trigger time or execution time (default=false).
  - Delay: formula evaluated when an event is triggered, and the result is the amount of time before the event is to be executed.
  - Priority: formula which sets the priority for this event when multiple events are scheduled to be executed simultaneously.
  - Dynamic Process: used for division, death, and movement.
  - List of Event Assignments: state change due to event.

Events


Event Asssignment Editor
Dimension Size Ids:
Variable:
IPTG
Indices:
Assignment:
60
Cancel
Add

Event Semantics

- Update the event queue:
  - Add events whose trigger expression has changed from false to true. (at $t = 0$ consider initialValue)
  - Evaluate newly triggered events' delay expressions and schedule them for the current time plus the delay.
  - If UseValuesAtTriggerTime is true, evaluate their event assignments.
  - Remove events with non-persistent trigger expressions that become false.
- Select next event to execute:
  - Compute priority of all events that can be executed at the current time, and execute the event with the highest priority.
  - If UseValuesAtTriggerTIme is false, event assignments are evaluated at execute time.
- Repeat this process until no additional events are executed.

Event Example

- Current state: $t = 0$ and $a = b = c = 0$.
- Events:

ID	e1	e2	e3	e4	e5
Trigger	true	a==1	a==1	a==2	a==2
InitialValue	false	true	true	true	true
Persistent	true	true	false	true	true
Delay	10	10	20	20	20
Priority	0	0	0	1	2
AtTriggerTime	true	true	false	true	true
Assignments	$a := 1$	$a := 2$	$b := a$	$c := 1$	$c := 2$

Event Example

- Current state: $t = 10$ and $a = b = c = 0$.
- Events:

ID	e1	e2	e3	e4	e5
Trigger	true	a==1	a==1	a==2	a==2
InitialValue	false	true	true	true	true
Persistent	true	true	false	true	true
Delay	10	10	20	20	20
Priority	0	0	0	1	2
AtTriggerTime	true	true	false	true	true
Assignments	$a := 1$	$a := 2$	$b := a$	$c := 1$	$c := 2$

- Event queue:

ID	e1
Time	10
Assignments	$a := 1$

Event Example

- Current state: $t = 10$, $a = 1$, and $b = c = 0$.
- Events:

ID	e1	e2	e3	e4	e5
Trigger	true	a==1	a==1	a==2	a==2
InitialValue	false	true	true	true	true
Persistent	true	true	false	true	true
Delay	10	10	20	20	20
Priority	0	0	0	1	2
AtTriggerTime	true	true	false	true	true
Assignments	$a := 1$	$a := 2$	$b := a$	$c := 1$	$c := 2$

- Event queue:

ID	e2	e3
Time	20	30
Assignments	$a := 2$	$b := a$

Event Example

- Current state: $t = 20$, $a = 2$, and $b = c = 0$.
- Events:

ID	e1	e2	e3	e4	e5
Trigger	true	a==1	a==1	a==2	a==2
InitialValue	false	true	true	true	true
Persistent	true	true	false	true	true
Delay	10	10	20	20	20
Priority	0	0	0	1	2
AtTriggerTime	true	true	false	true	true
Assignments	$a := 1$	$a := 2$	$b := a$	$c := 1$	$c := 2$

- Event queue:

ID	e4	e5
Time	40	40
Assignments	$c := 1$	$c = 2$

Event Example

- Current state: $t = 40$, $a = 2$, $b = 0$, and $c = 2$.
- Events:

ID	e1	e2	e3	e4	e5
Trigger	true	a==1	a==1	a==2	a==2
InitialValue	false	true	true	true	true
Persistent	true	true	false	true	true
Delay	10	10	20	20	20
Priority	0	0	0	1	2
AtTriggerTime	true	true	false	true	true
Assignments	$a := 1$	$a := 2$	$b := a$	$c := 1$	$c := 2$

- Event queue:

ID	e4
Time	40
Assignments	$c = 1$

Event Example

- Current state: $t = 40$, $a = 2$, $b = 0$, and $c = 1$.
- Events:

ID	e1	e2	e3	e4	e5
Trigger	true	a==1	a==1	a==2	a==2
InitialValue	false	true	true	true	true
Persistent	true	true	false	true	true
Delay	10	10	20	20	20
Priority	0	0	0	1	2
AtTriggerTime	true	true	false	true	true
Assignments	$a := 1$	$a := 2$	$b := a$	$c := 1$	$c := 2$

Unit Definitions

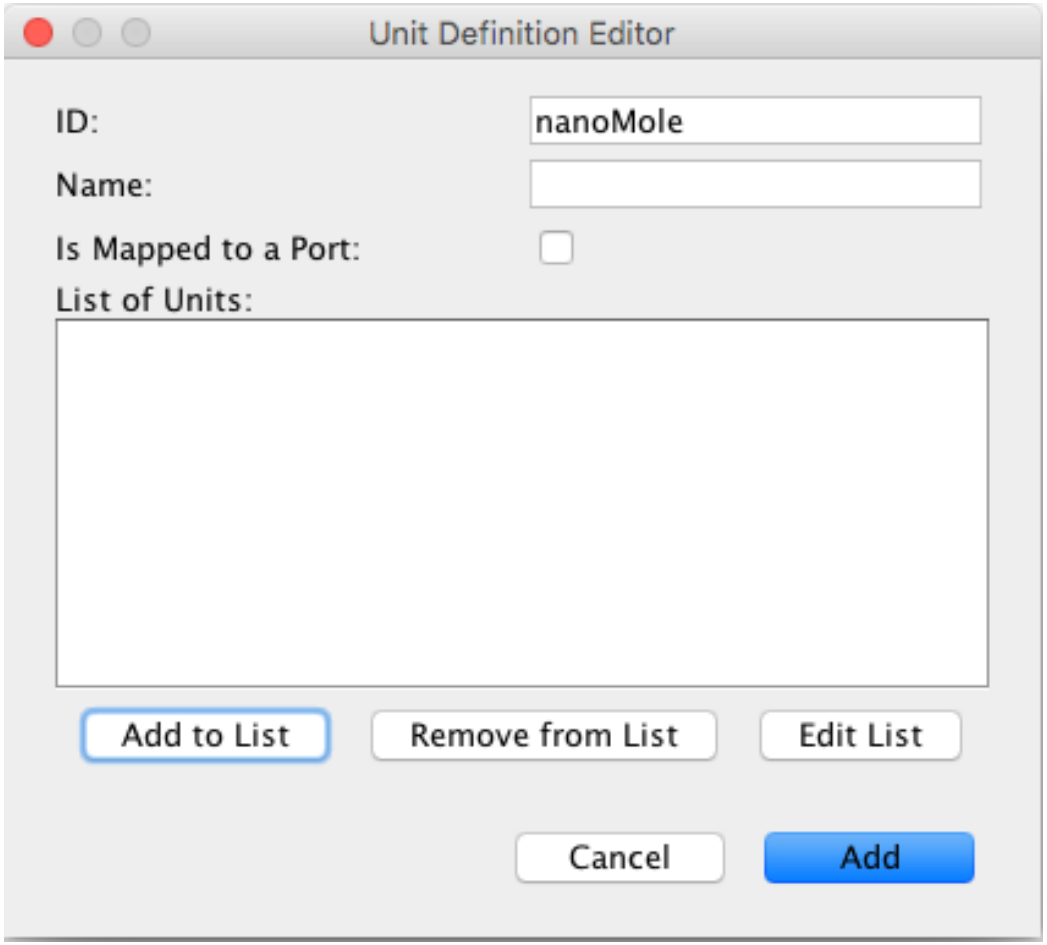
- Unit definitions are used to construct user-defined units which are derived from the set of base units.

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	

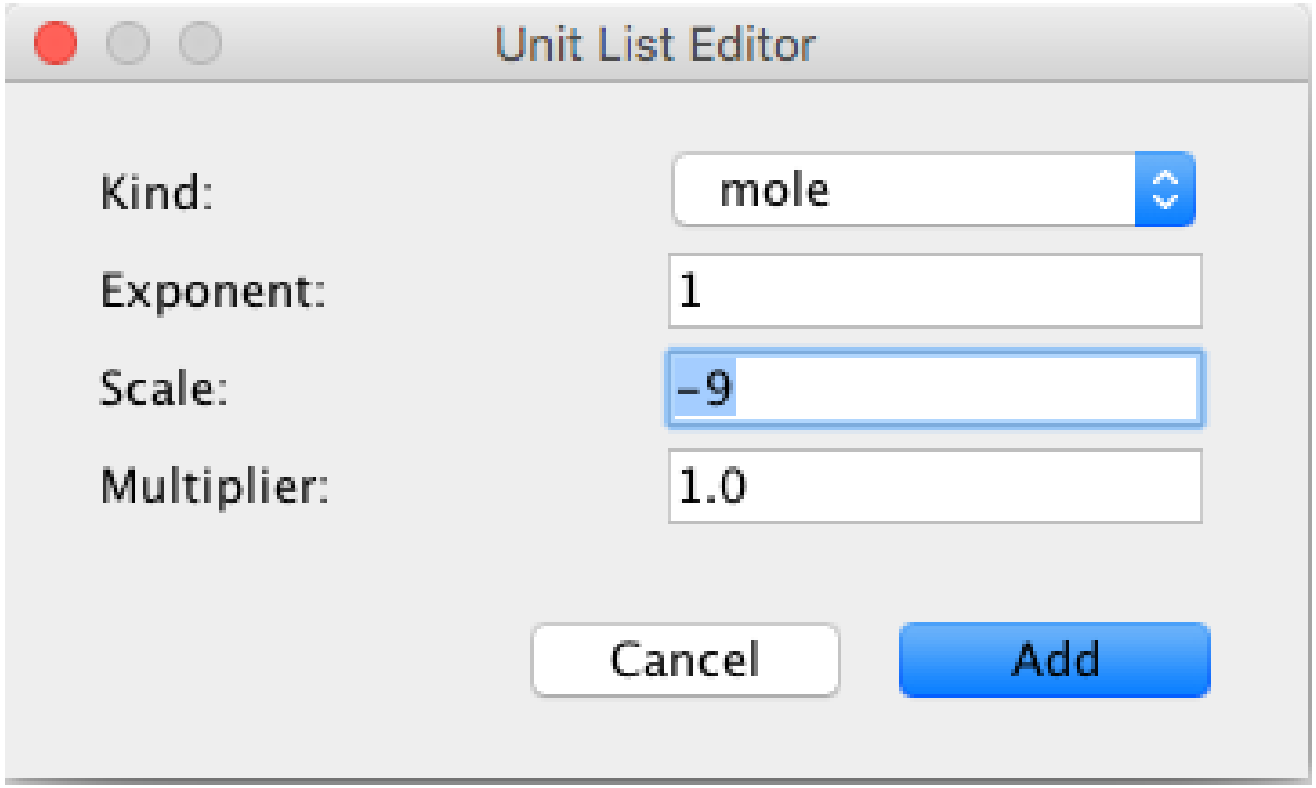
- A unit definition includes an ID, an optional name, and a list of units.
- Each unit is composed of a kind, exponent, scale, and multiplier.
- Exponent and multiplier are real numbers, and scale is an integer.

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

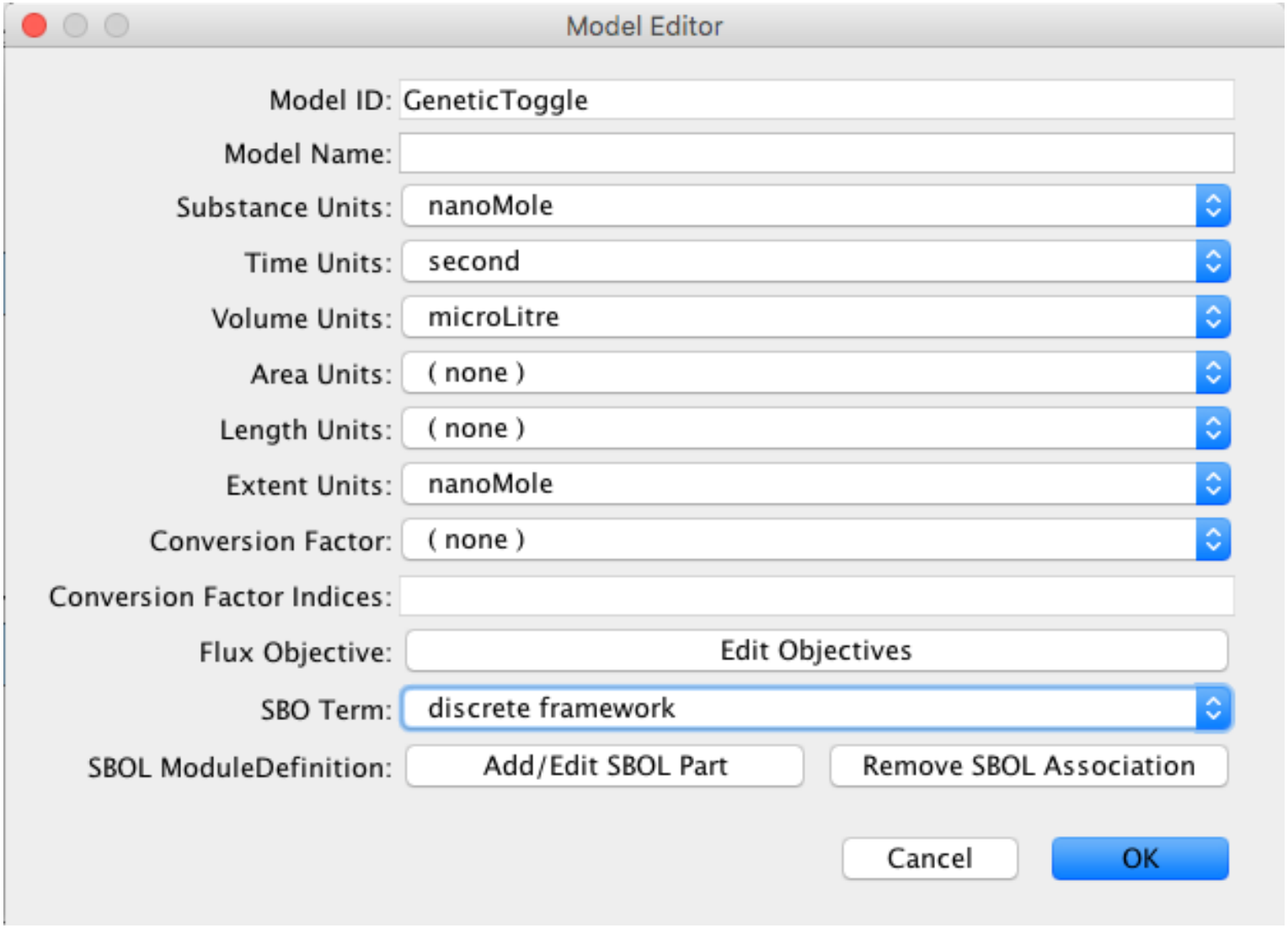
Unit Definitions



Unit Definitions



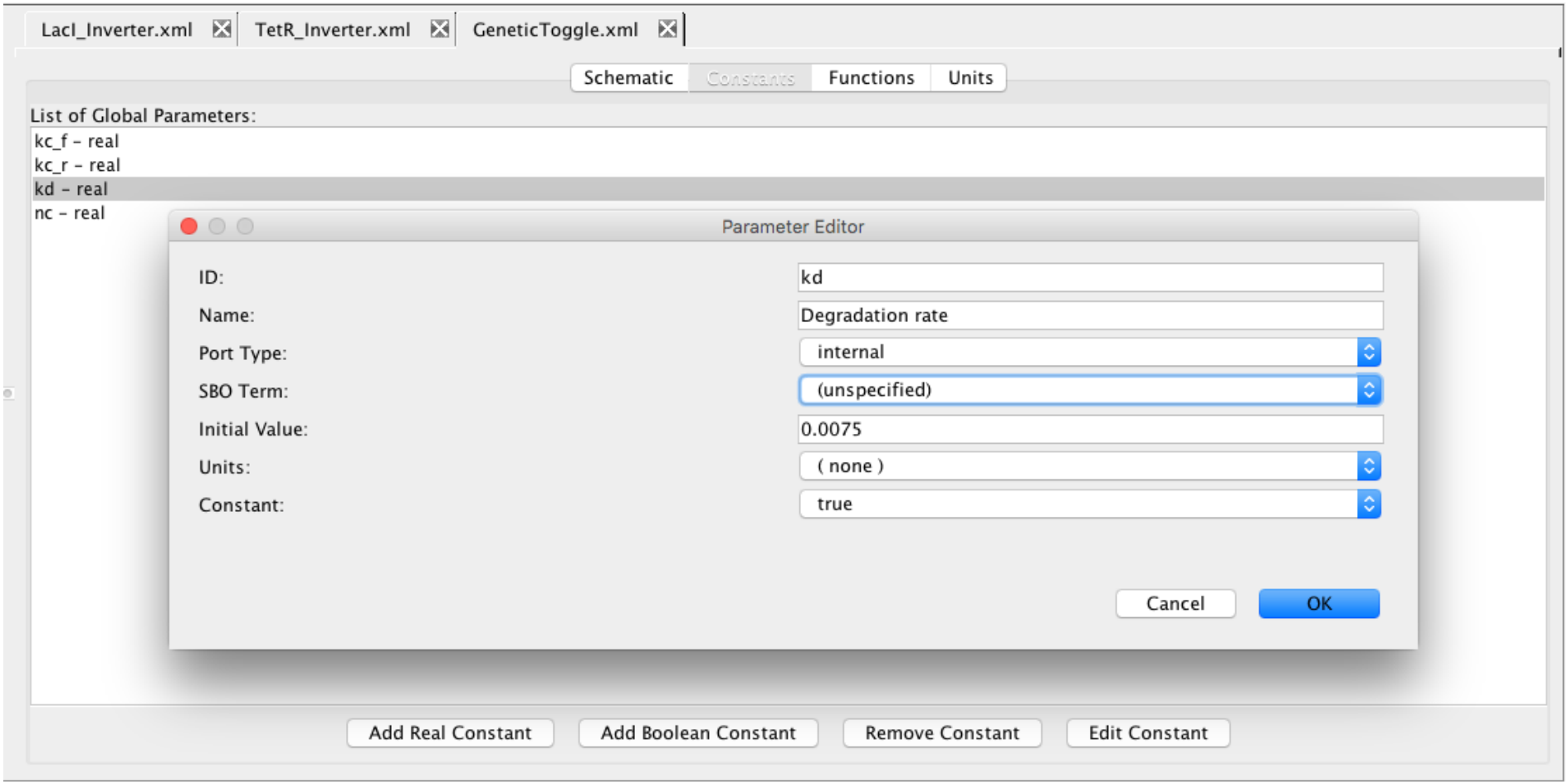
Model Editor



Model Editor

- ID: composed only of alphanumeric characters and underscores.
- Name: an arbitrary string description (optional).
- *Model units* are the default units for a model.
  - *Substance units* are the default units for species.
  - *Time units* are the default units for time.
  - *Volume units* are the default units for 3-dimensional compartments.
  - *Area units* are the default units for 2-dimensional compartments.
  - *Length units* are the default units for 1-dimensional compartments.
  - *Extent units* are the default units for amount changed due to reactions. Kinetic laws have default units of extent units / time units.
- *Conversion factor* is constant global parameter used as the default conversion factor to convert species units into extent units.
- *Conversion factor indices* used to reference a parameter in an array.
- *Flux Objective* used to set flux balance objectives.
- SBO Term: from the *modelling framework* branch of SBO.
- *SBOL Module Definition* used to associate SBOL with this model.

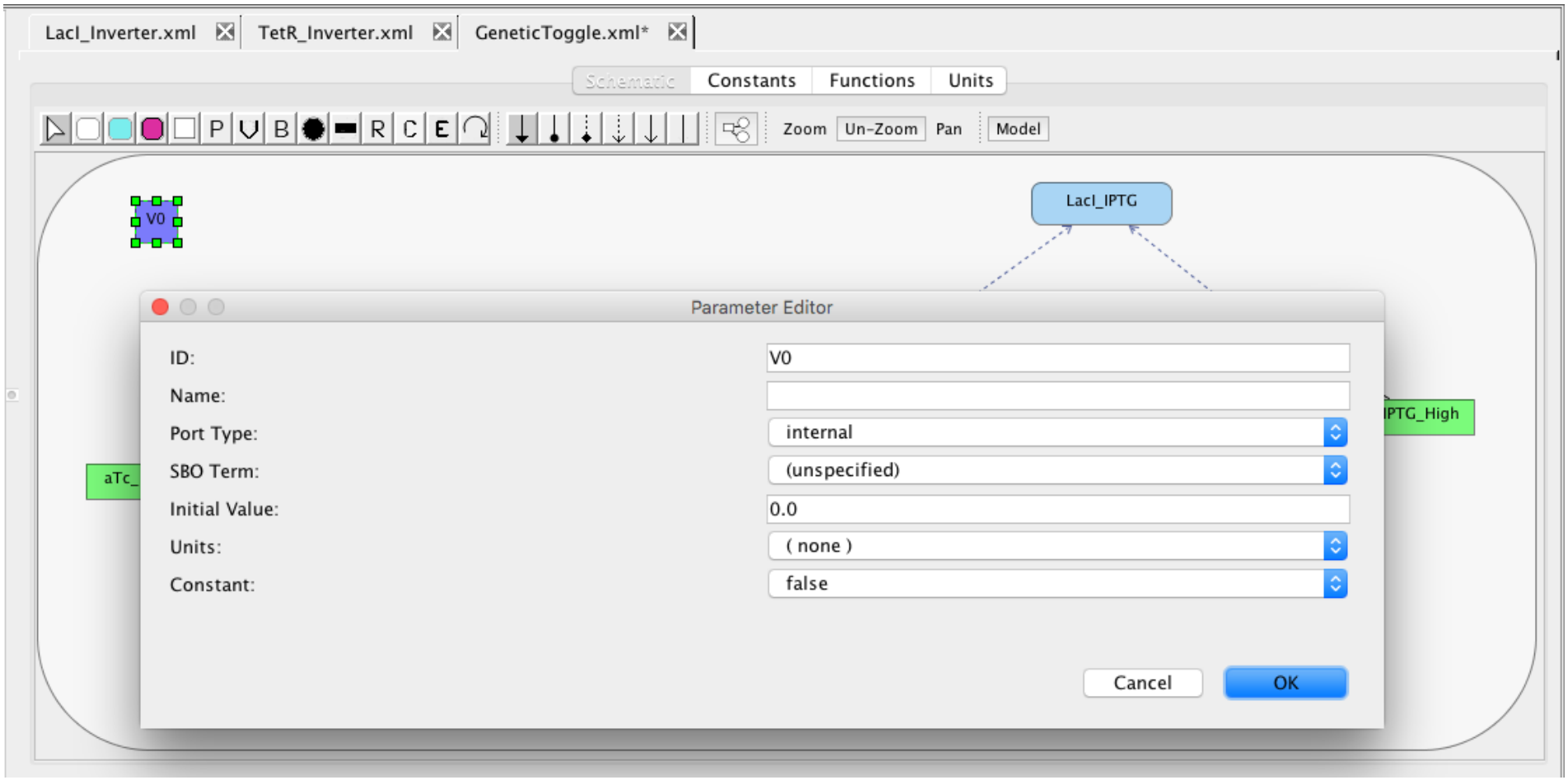
Constants


LacI_Inverter.xml
TetR_Inverter.xml
GeneticToggle.xml
Schematic
Constants
Functions
Units
List of Global Parameters:
kc_f - real
kc_r - real
kd - real
nc - real
Parameter Editor
ID:
kd
Name:
Degradation rate
Port Type:
internal
SBO Term:
(unspecified)
Initial Value:
0.0075
Units:
(none)
Constant:
true
Cancel
OK
Add Real Constant
Add Boolean Constant
Remove Constant
Edit Constant

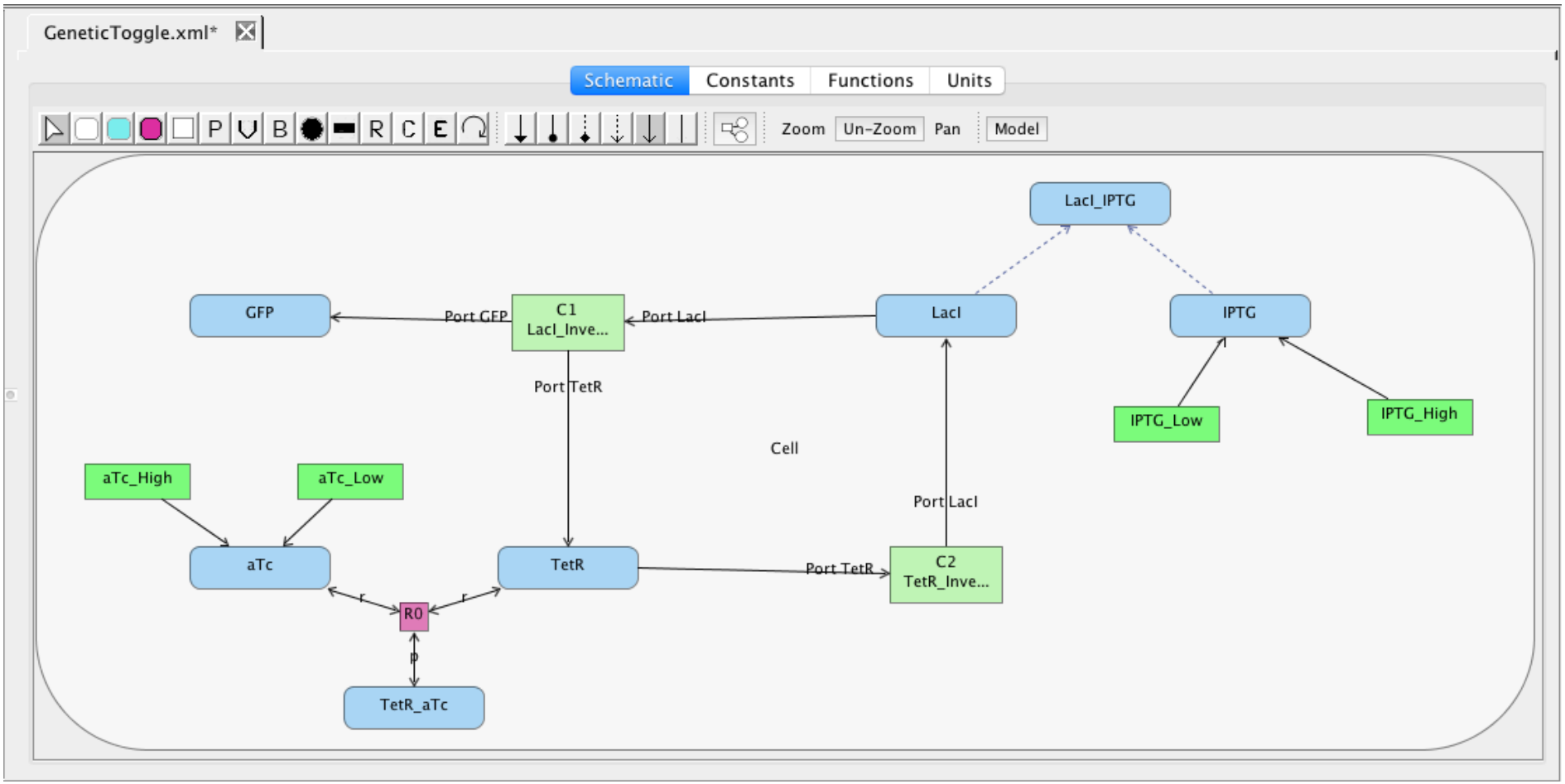
Parameters

- Global parameters are constants or variables that can be used in math formulas, and include the following fields:
  - ID: composed only of alphanumeric characters and underscores.
  - Name: an arbitrary string description (optional).
  - Port Type: input, internal, output.
  - SBO Term: from the *systems description paramter* branch of SBO.
  - Initial Value: initial value for the parameter.
  - Units: the units for the parameter value (default=none).
  - Constant: indicates if the parameter value is constant (default=true).

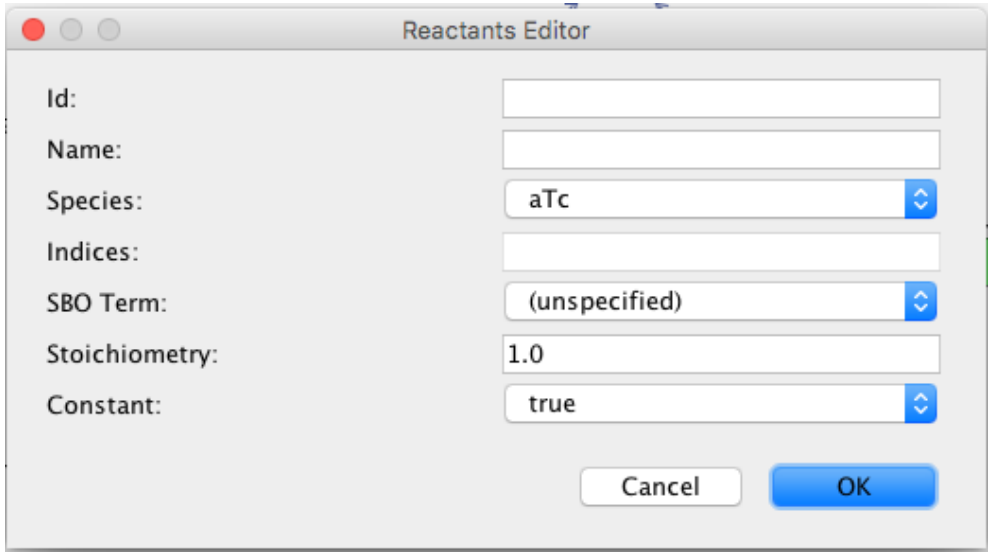
Variables

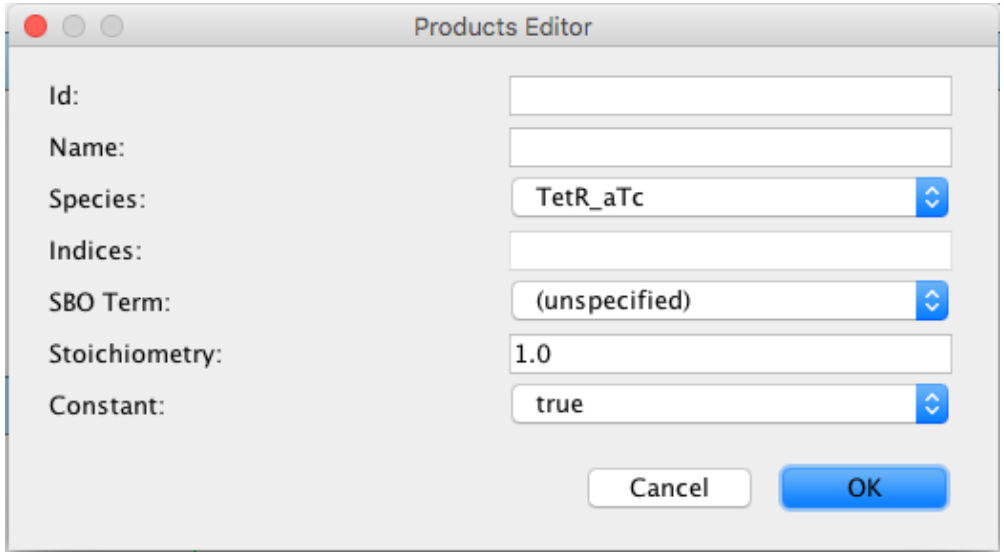

LacI_Inverter.xml
TetR_Inverter.xml
GeneticToggle.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
V0
LacI_IPTG
Parameter Editor
ID:
V0
Name:
Port Type:
internal
SBO Term:
(unspecified)
Initial Value:
0.0
Units:
(none)
Constant:
false
Cancel
OK

Reactions


GeneticToggle.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
LacI_IPTG
GFP
Port GFP
C1
LacI_Inve...
Port LacI
LacI
IPTG
Port TetR
IPTG_Low
IPTG_High
Cell
aTc_High
aTc_Low
Port LacI
aTc
TetR
Port TetR
C2
TetR_Inve...
r
r
R0
p
TetR_aTc

Reactant/Product Editors



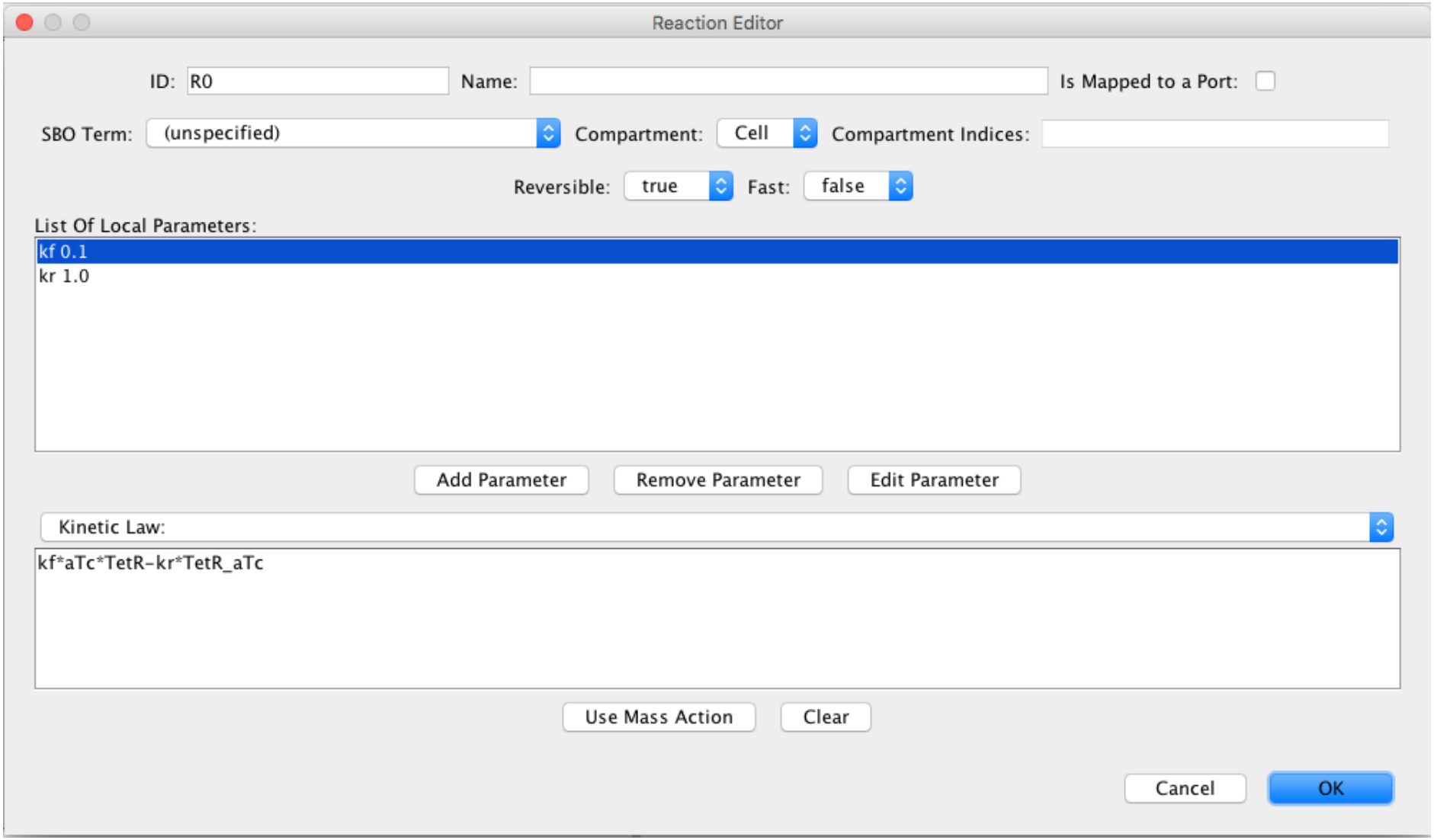


Reactants/Products

- ID: do not confuse with species ID, used in math formulas for stoichiometry (optional).
- Name: an arbitrary string description of the species (optional).
- Species: the ID of the reactant/product species.
- Indices: reference species within an array.
- SBO Term: from the *participant role* branch of SBO.
- Stoichiometry: the number of molecules consumed by the reaction (may be overridden when the reactant/product ID is used in an assignment).
- Constant: indicates whether the stoichiometry can change dynamically.

NOTE: There is also a modifier editor without stoichiometry/constant fields.

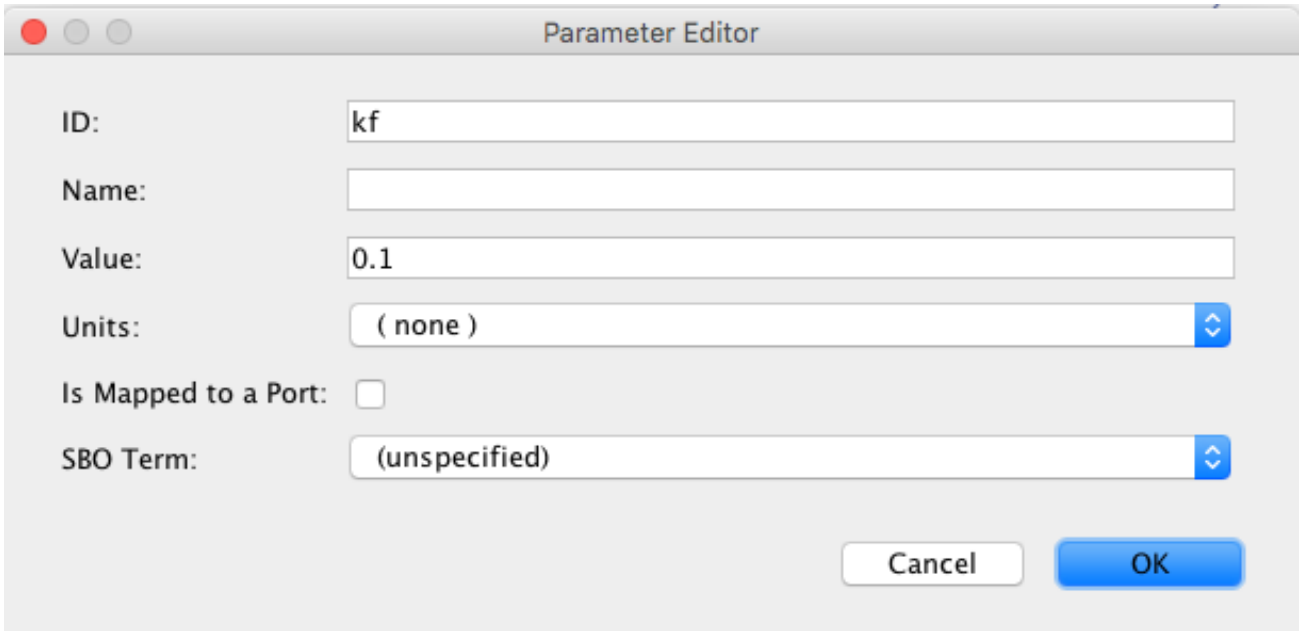
Reaction Editor



Reactions

- Reactions are used to create or destroy molecular species.
- A reaction is composed of the following:
  - ID: composed only of alphanumeric characters and underscores.
  - Name: an arbitrary string description (optional).
  - Is Mapped to a Port: indicates if this reaction can be replaced or deleted in hierarchical models (default=false).
  - SBO Term: from the *occurring entity representation* branch of SBO.
  - Compartment: the location where this reaction takes place.
  - Compartment Indicies: used to reference compartment when an array.
  - Reversible: indicates if the reaction is reversible (default=false).
  - Fast: indicates if the reaction reaches equilibrium rapidly (default=false). Limited support in analysis.
  - List of Local Parameters: symbolic values that can be used in the *kinetic law* for this reaction (default *kf* and *kr* are provided).
  - Kinetic Law: math formula describing the rate or probability for this reaction.

Local Parameter Editor

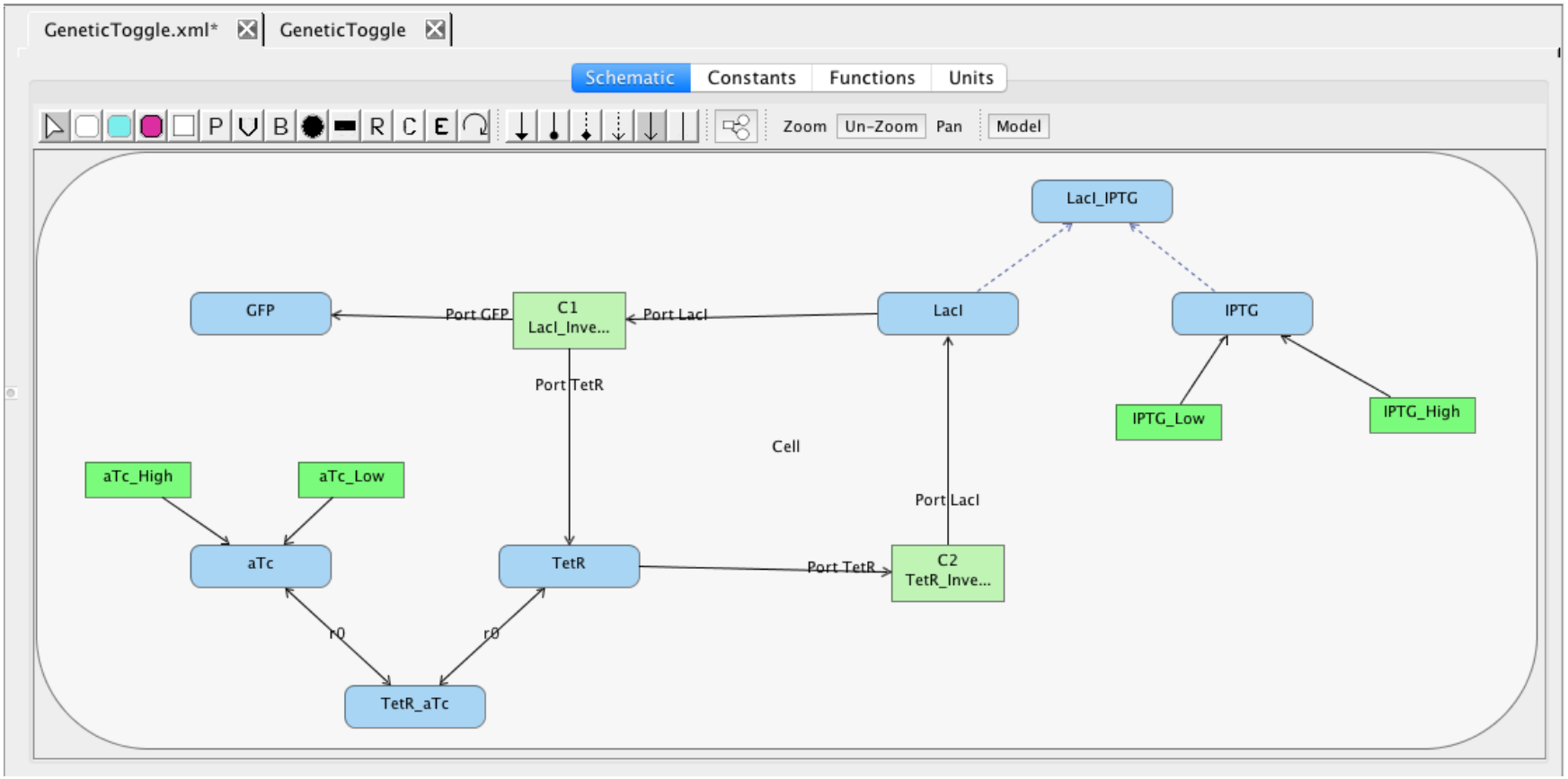


Local Parameters

- Local parameters are constants that can be used in kinetic laws, and include the following fields:
  - ID: composed only of alphanumeric characters and underscores.
  - Name: an arbitrary string description (optional).
  - Initial Value: initial value for the parameter.
  - Units: the units for the parameter value (default=none).
  - Is Mapped to a Port: indicates if this local parameter can be replaced or deleted in hierarchical models (default=false).
  - SBO Term: from the *systems description paramter* branch of SBO.

Implicit Reactions

- 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.



SBML Mathematical Formulas

- Variables (compartments, species, parameters, reaction IDs, and species reference IDs).
- Real numbers followed optionally by its units.
- Vector constant ({ }) which can be nested.
- Array selector(s) ([ ]) can occur after array variables or vector constants.
- Built-in constants: exponential, pi, Infinity, NaN, true, and false.
- Special variable time or t, which returns the current simulation time.
- Mathematical operators including add (+), subtract (-), multiply (*), divide (/), power (^ or pow(x,y)), and root.
- A function defined in the list of function definitions.
- Logical functions: and (&&), or (||), xor, not (!).
- Relational functions: eq (==), neq (!=), geq (>=), gt (>), leq (<=), and lt (<).

SBML Mathematical Formulas (cont)

- 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.
- 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 Special Functions

- 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., $a - floor(a/b) * b$)).

iBioSim Random Functions

- `iBioSim` adds support for random functions.
- Continuous random functions:
  - uniform(a,b)
  - normal(m,s)
  - exponential(mu)
  - gamma(a,b)
  - lognormal(z,s)
  - chisq(nu)
  - laplace(a)
  - cauchy(a)
  - rayleigh(s)
- Discrete random functions:
  - poisson(mu)
  - binomial(p,n)
  - bernoulli(p)

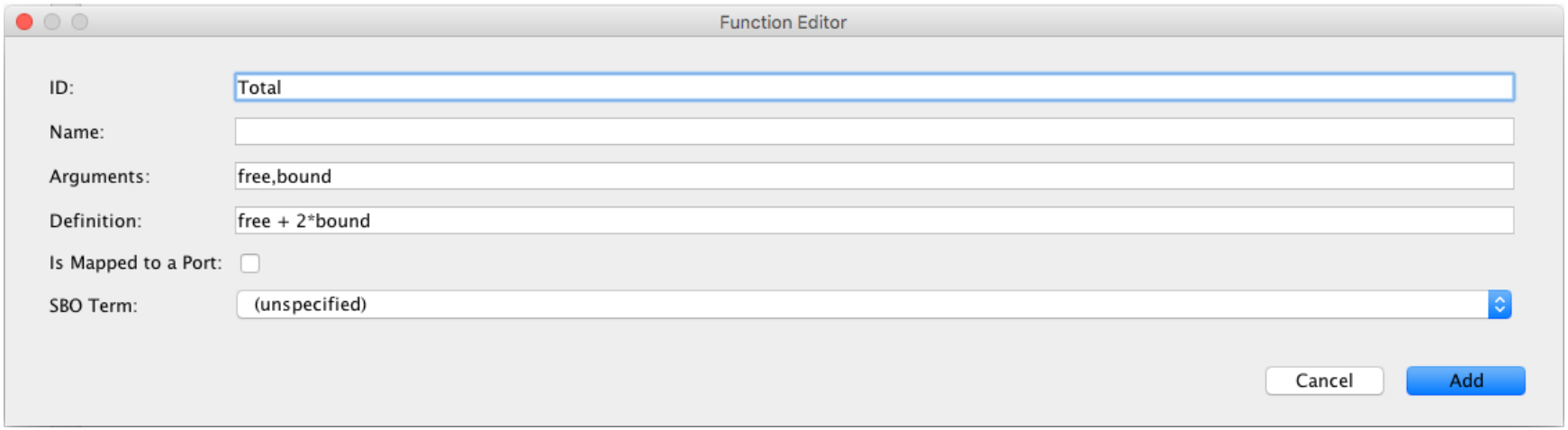
Continuous Stochastic Logic (CSL)

- $PG(t,x)$ - 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.
- $U(t,x,y)$ - 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., $G(t,x) \wedge F(t,y)$).

Function Definitions

- Function definitions are used to create user defined functions that can then be used in SBML math formulas.
- Function definitions include an ID, an optional name field, a comma-separated list of arguments, and its definition.
- Can only use variable names which are arguments to the function.
- While functions can call other functions, they cannot be recursive either directly or indirectly.

Function Definitions

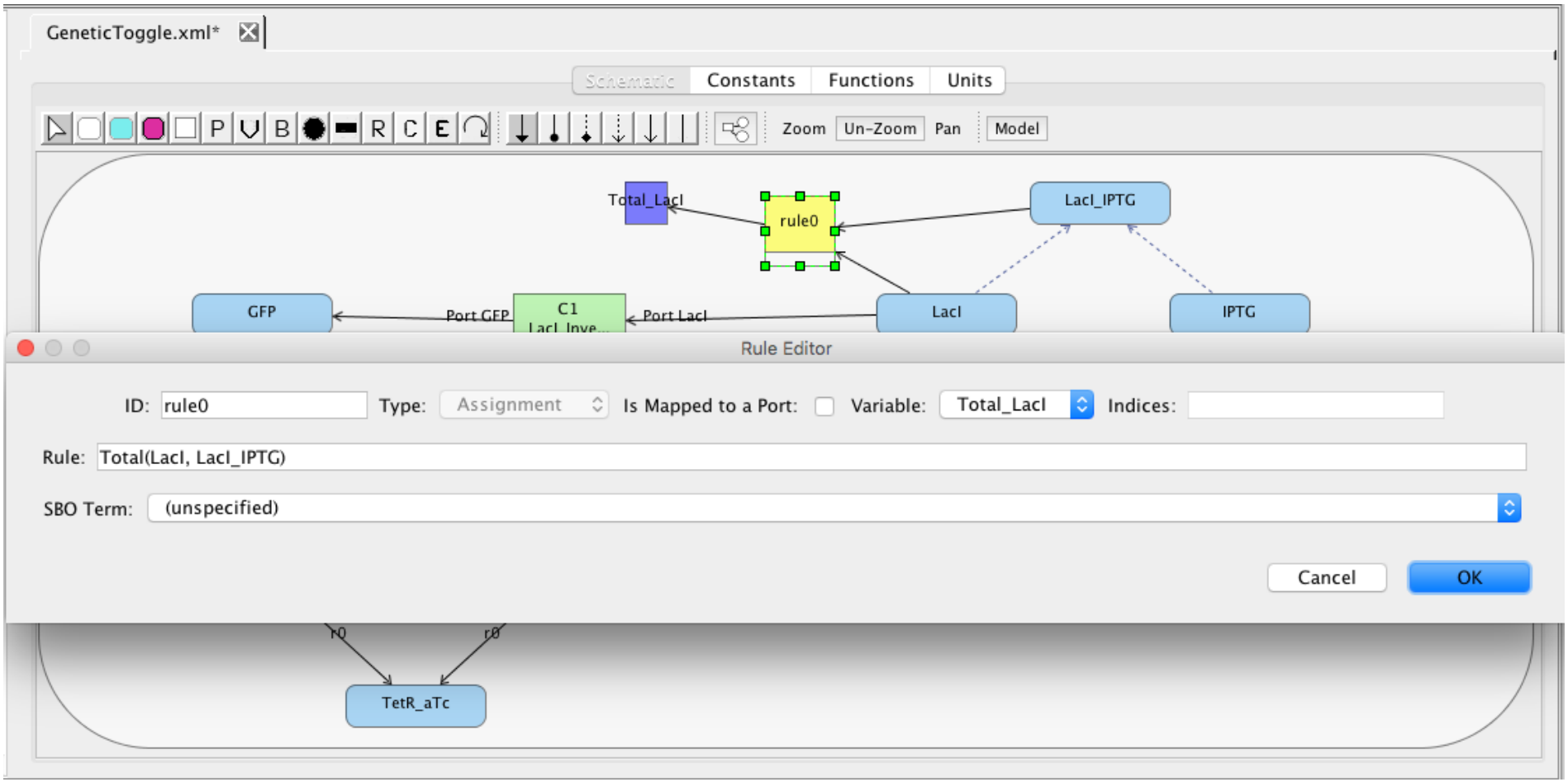


Rules

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 with a formula.
- Variable cannot be determined by both assignment rule and initial assignment.
- Rate rules specify the rate of change of a compartment size, species amount or concentration, or parameter with a formula.
- A variable cannot be determined by both an assignment and a rate rule.
- A species that is reactant or product of any reaction cannot be updated by either an assignment rule or rate rule.

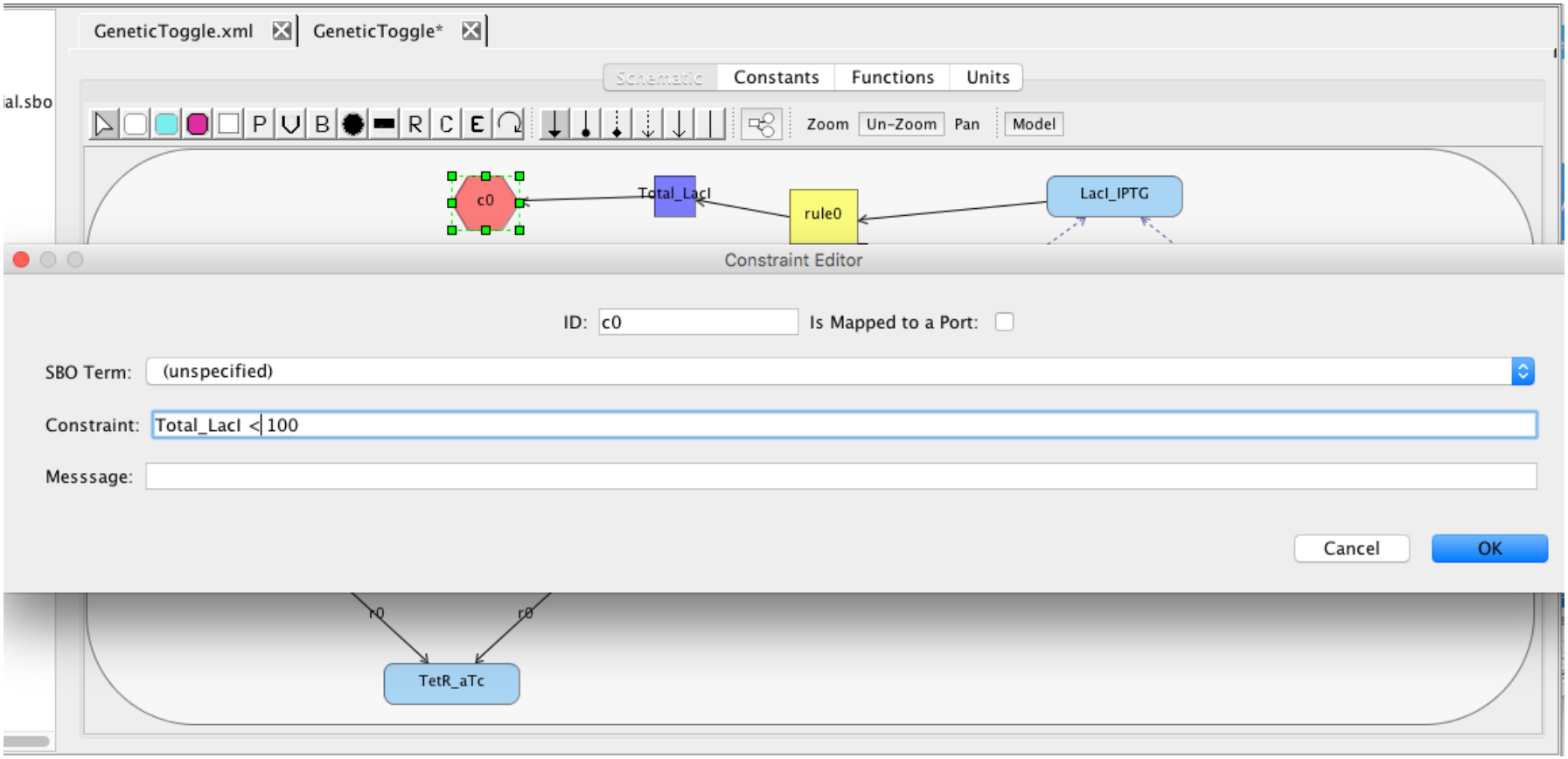
Rules


GeneticToggle.xml*
Schematic
Constants
Functions
Units
Zoom
Un-Zoom
Pan
Model
Total_LacI
rule0
LacI_IPTG
GFP
Port GFP
C1
LacI_Inve...
Port LacI
LacI
IPTG
Rule Editor
ID: rule0
Type: Assignment
Is Mapped to a Port:
Variable: Total_LacI
Indices:
Rule: Total(LacI, LacI_IPTG)
SBO Term: (unspecified)
Cancel
OK
r0
r0
TetR_aTc

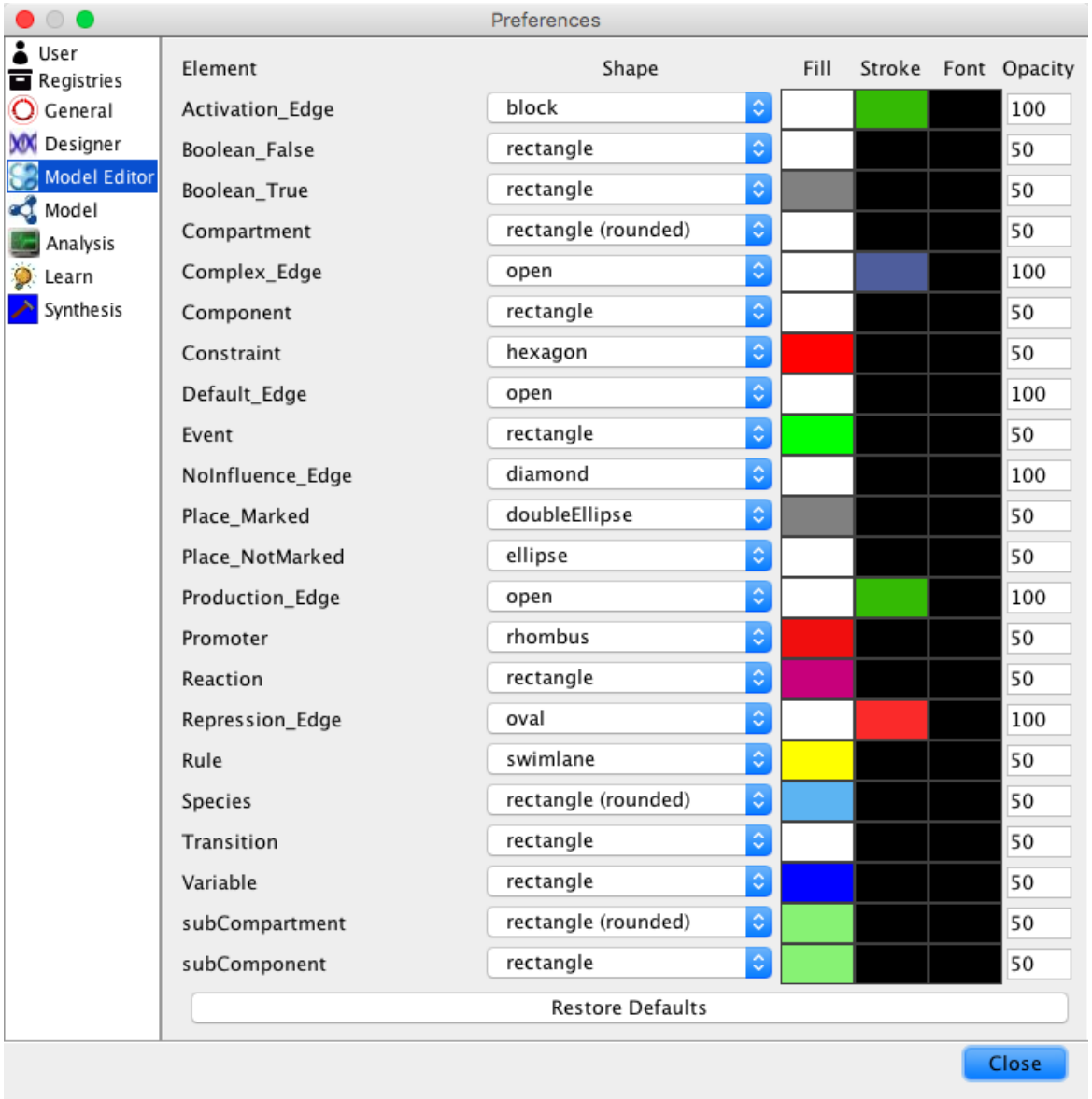
Constraints

- Constraints specify properties that should remain true.
- If a constraint becomes false, simulation terminates.
- Our analysis method can provide histograms that show the proportion of simulations that are terminated due to each possible constraint.
- Can also provide time courses showing when constraints become false.
- Composed of an ID which is used to identify it in graphs, a constraint given as a formula, and a message describing the constraint.
- A default id is automatically generated when a new constraint is created.

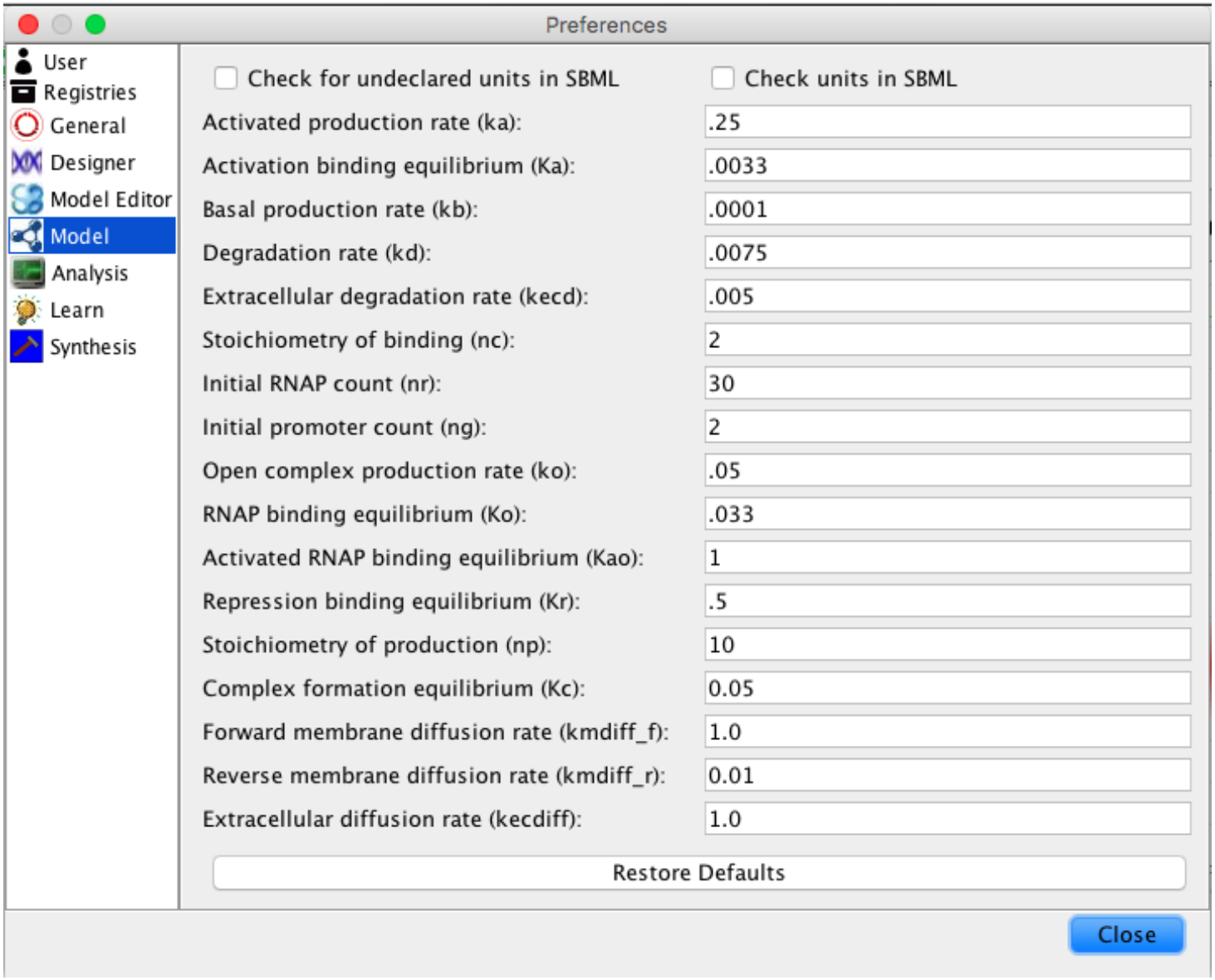
Constraints



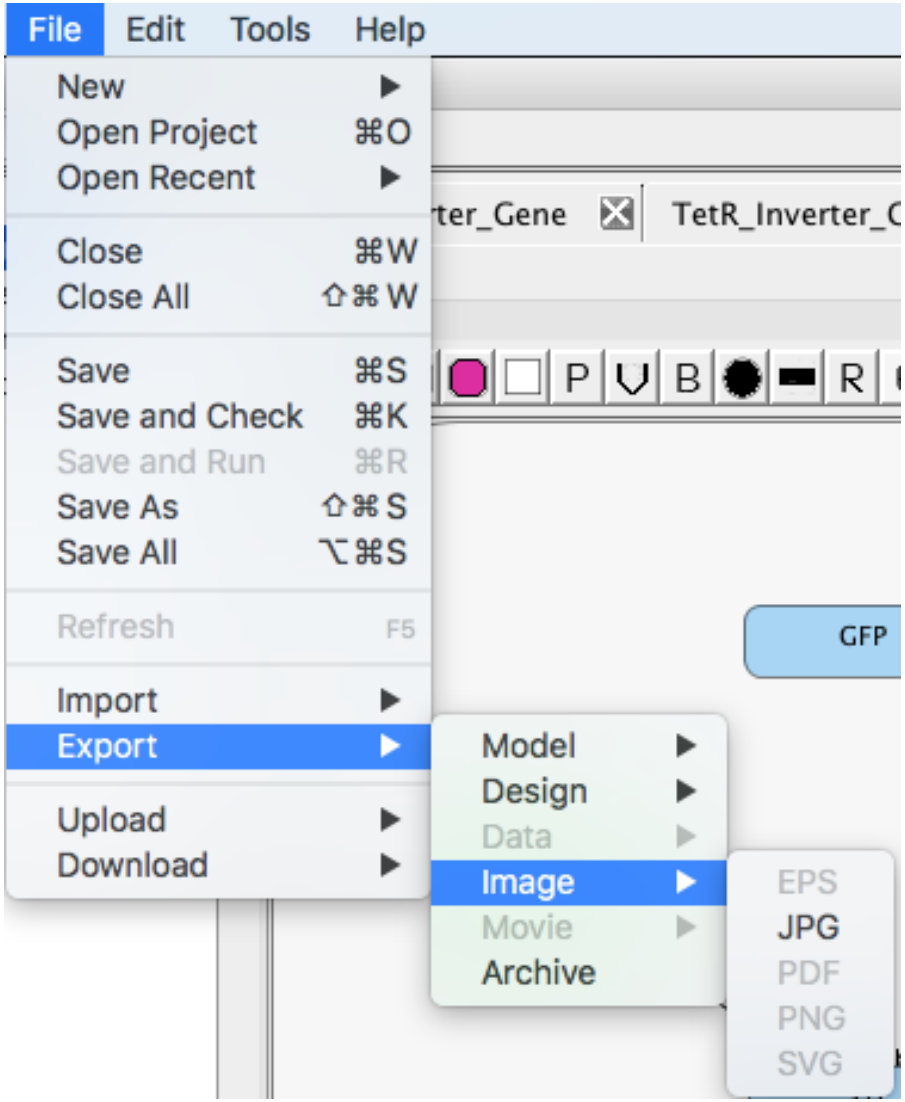
Model Editor Preferences



Model Preferences



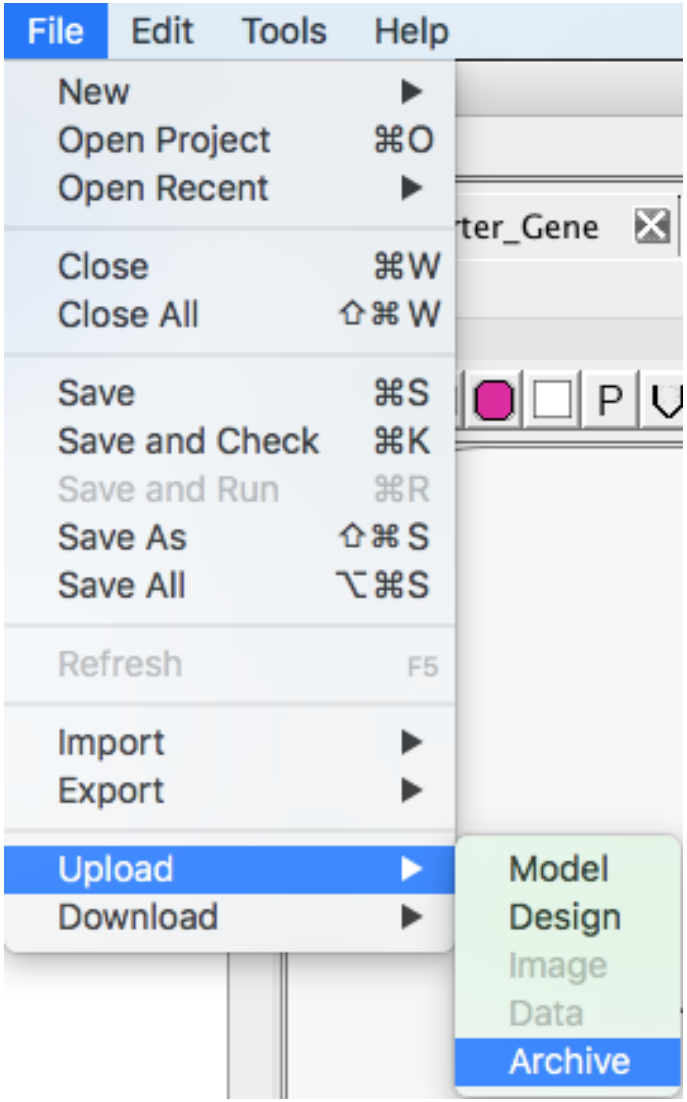
Exporting Models and Other Files



Exporting Models and Other Files

- You can export:
  - Both hierarchical and flattened SBML models.
  - SBOL in both versions 1 and 2.
  - Sequences in GenBank and FASTA formats.
  - Simulation data in TSD, CSV, or DAT formats.
  - Graphs and model images in EPS, JPG, PDF, PNG, and SVG formats.
  - Simulation movies in AVI or MP4.
  - COMBINE Archive files.

Uploading Models and Archives



Credits

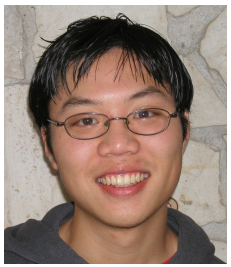
Nathan Barker

Scott Glass

Kevin Jones

Hiroyuki Kuwahara

Curtis Madsen



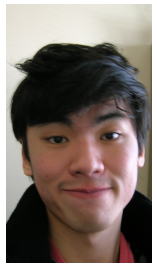
Nam Nguyen

Tramy Nguyen

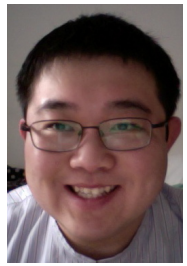
Tyler Patterson

Nicholas Roehner

Jason Stevens



Leandro Watanabe



Zhen Zhang

Supported by National Science Foundation Grants ECCS-0331270, CCF-07377655, CCF-0916042, CCF-1218095, DBI-1356041, and CCF-1748200.

Assignment #4

- Construct a genetic circuit model for the genetic toggle switch.
  - Construct a model of the LacI_Inverter with input species LacI and output species TetR, and attach your DNA design for this inverter to the promoter.
  - Construct a model of the TetR_Inverter with input species TetR and output species LacI and GFP, and attach your DNA design for this inverter to the promoter.
  - Construct a top-level model that connects the two inverters using the LacI, TetR, and GFP species.
  - Add to your top-level model the small molecules IPTG and aTc, as well as complex formation reactions with LacI and TetR, respectively.
  - Upload a COMBINE Archive for your project to `https://synbiohub.utah.edu`, and include a share link in your submission.
- Construct a genetic circuit model for your genetic circuit.
  - Upload a COMBINE Archive for your project to `https://synbiohub.utah.edu`, and include a share link in your submission.