Theory

Modeling of Biochemical Reaction Systems

Assumptions:

- The reaction systems are spatially homogeneous at every moment of time evolution.
- The underlying reaction rates are described by the mass action law.

Model Description:

- Variables: Entities that change in time governed by chemical reactions. In the example below, [A], [B], [C].
- Parameters: Entities that do not change or change independently of the chemical reactions, can be perturbed by external intervention: k(t).
- Time evolution:

$$A + B \rightarrow C \qquad \frac{d[C]}{dt} = k(t)[A][B]$$

Deterministic or Stochastic Models

When the number of molecules is large (> ~1000 per cell):

- Concentrations fluctuate in a continuous manner.
- Concentration fluctuations are negligible.
- Time evolution is described deterministically.

E.g.,

$$A + B \rightarrow C \qquad \frac{d[C]}{dt} = k[A][B]$$

When the number of molecules is small (< ~1000 per cell):

- Concentrations fluctuate in a discrete manner.
- Concentrations fluctuate significantly.

  The number of *LacI* tetrameric repressor protein in E.coli ~ 10 molecules.
  If one *LacI* repressor binds to a promoter region, the number of free *LacI* repressors = 9.
  10% change in its concentration and number!

- Time evolution is described stochastically.

Numerical Simulation Algorithms

Deterministic Models

- Ordinary differential equations
- Standard libraries e.g., CVODE, etc.

Stochastic Models

- The master equations
- Gillespie's stochastic simulation algorithm and its variants.

Basic Terms

Stoichiometric Amount The stoichiometric amount is defined as the number of molecules of a particular reactant or product partaking in a reaction. Stoichiometric amounts will always be positive numbers.

Stoichiometric coefficient The stoichiometry coefficient refers to the relative amount of substance that is consumed and/or produced by a reaction.

Rate of Change The rate of change in concentration or amount of a specified molecular species.

Reaction Rate The rate of reaction, r, is defined to be the slope of the concentration-time plot for a species divided by the stoichiometric coefficient of that species.

Mass-action Ratio The ratio of the products to the reactants *in vivo* is called the mass-action ratio, denoted by the symbol Γ.

Disequilibrium Constant The ratio of the mass-action ratio to the equilibrium constant is called the disequilibrium ratio and denoted by the symbol, ρ.

Elasticity The elasticity describes how sensitive a reaction rate is to changes in reactant, product and effector concentrations, that is the degree to which changes are transmitted from the immediate environment of a reaction to the reaction rate.

Kinetic order See elasticity

Stoichiometric Amounts

Page 3 in book

The **stoichiometric amount** is the number of molecules of a particular reactant or product taking part in a reaction.

List the stoichiometric amounts in the following reaction:

$$2A + B \rightarrow A + C$$

On the reactant side the stoichiometric amount for A is two and for B is one. On the product side, the stoichiometric amount for A is one and for C one.

Stoichiometric Coefficients

Page 6/7 in book

The **stoichiometric coefficient**, c_i, for a molecular species A_i, is the difference between the stoichiometric amount of the species on the product side and the stoichiometric amount of the same species on the reactant side, that is:

$$c_i = \text{Stoichiometric Amount of Product, } A_i - \text{Stoichiometric Amount of Reactant, } A_i$$

Write down the stoichiometric coefficients for the following reactions:

a) $A + A \rightarrow A + B$

The stoichiometric amount of A on the reactant side is 2 and on the product side, 1. Therefore the stoichiometric coefficient for A is $1 - 2 = -1$. The stoichiometric amount of B on the product side is 1 and on the reactant side, 0, therefore the stoichiometric coefficient for B is $1 - 0 = 1$.

b) $A \rightarrow B + \frac{1}{2}A$

The stoichiometric amount of A on the reactant side is 1 and on the product side $\frac{1}{2}$, therefore the stoichiometric coefficient for A is $1/2 - 1 = -1/2$. The stoichiometric amount of B on the reactant side is 0 and on the product side, 1, therefore the stoichiometric coefficient for B is $1 - 0 = 1$.

Rate of Change

Page 5 in book

The rate of change can be defined as the rate of change in concentration or amount (depending on units) of a designated species. If S is the species then the rate of change is given by:

$$\text{Rate} = \frac{\Delta S}{\Delta t} \qquad \text{Rate} = \frac{dS}{dt}$$

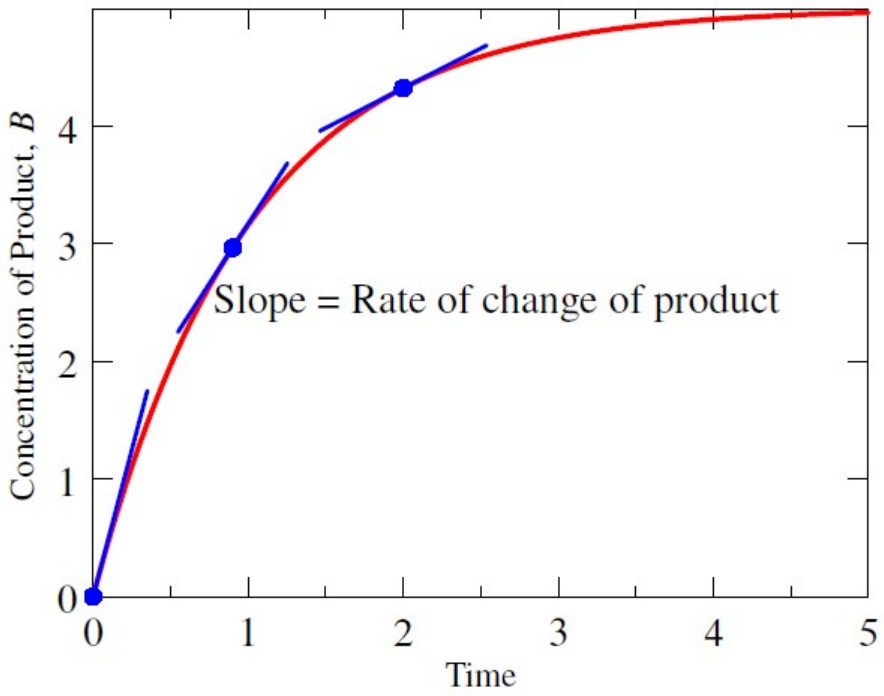


Figure 1.1: Progress curve for a simple irreversible reaction, $A \rightarrow B$.

Reaction Rate (v)

Page 8/9 in book

For a reaction of the form

$$n_1 A + n_2 B + \ldots \longrightarrow m_1 P + m_2 Q + \ldots$$

where we assume that each species only occurs on one side of the reaction, and where $n_1, n_2, \ldots$ and $m_1, m_2, \ldots$ represent the stoichiometric amounts, the reaction rate is given by:

$$\text{Rate} = v \equiv -\frac{1}{n_1}\frac{dA}{dt} = -\frac{1}{n_2}\frac{dB}{dt} \ldots = \frac{1}{m_1}\frac{dP}{dt} = \frac{1}{m_2}\frac{dQ}{dt} \ldots \qquad (1.1)$$

A reaction rate is the rate of change normalized with respect to the stoichiometric coefficient.

Reaction Rate: Rate Laws

Mass-action

$$v = k \prod_i S_i^{n_i}$$

Michaelis-Menten

$$v = \frac{Vm\ S}{K_d + S}$$

Reversible Michaelis-Menten

$$v = \frac{V_f / K_S (S - P/K_{eq})}{1 + S/K_S + P/K_P}$$

Hill Equation

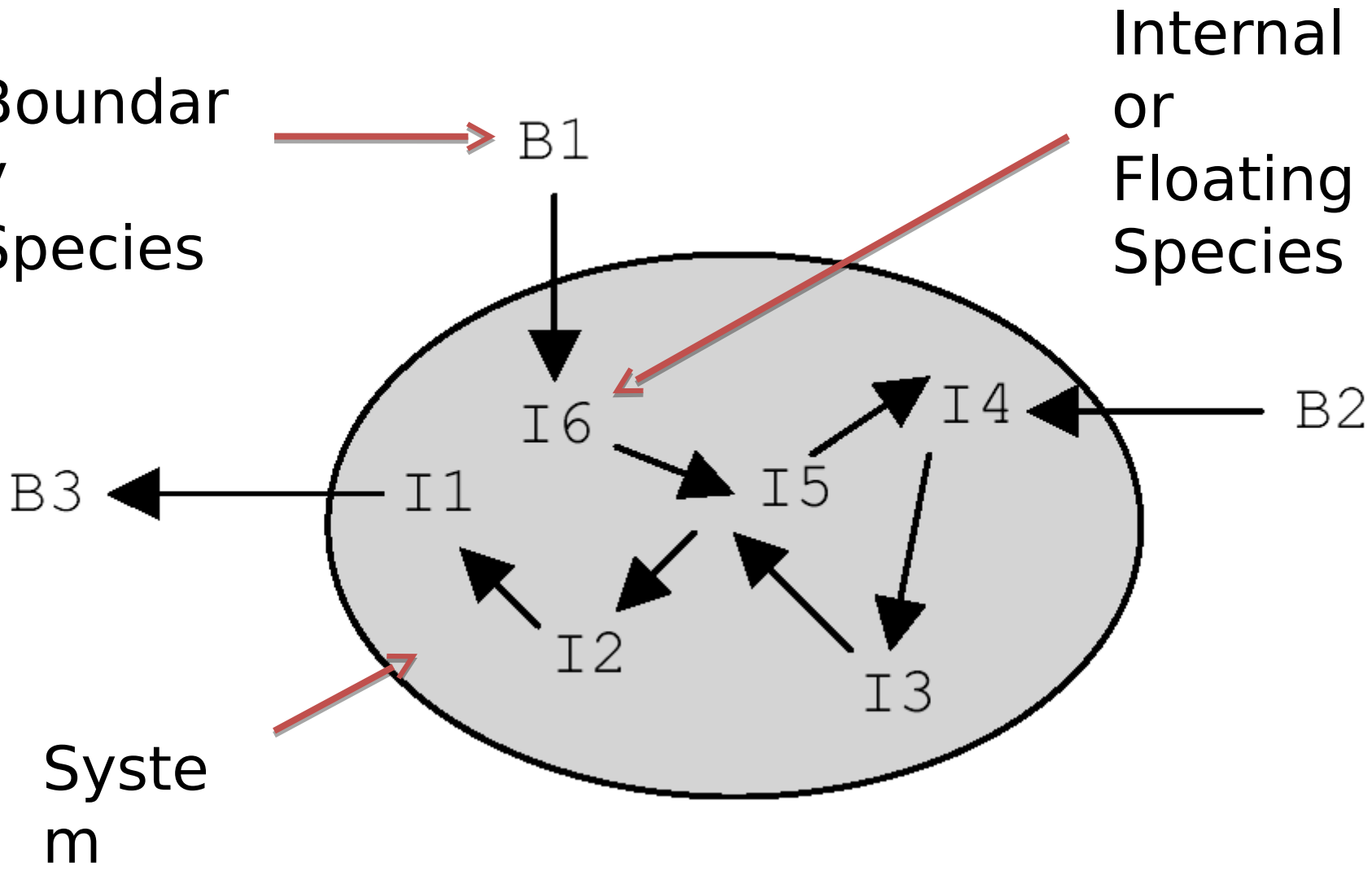
$$v = \frac{Vm\ S^n}{K_d + S^n}$$

Cooperatively + Allosteric Equation

$$Y = \frac{\alpha\,(1+\alpha)^{n-1}}{(1+\alpha)^n + L(1+\beta)^n}$$

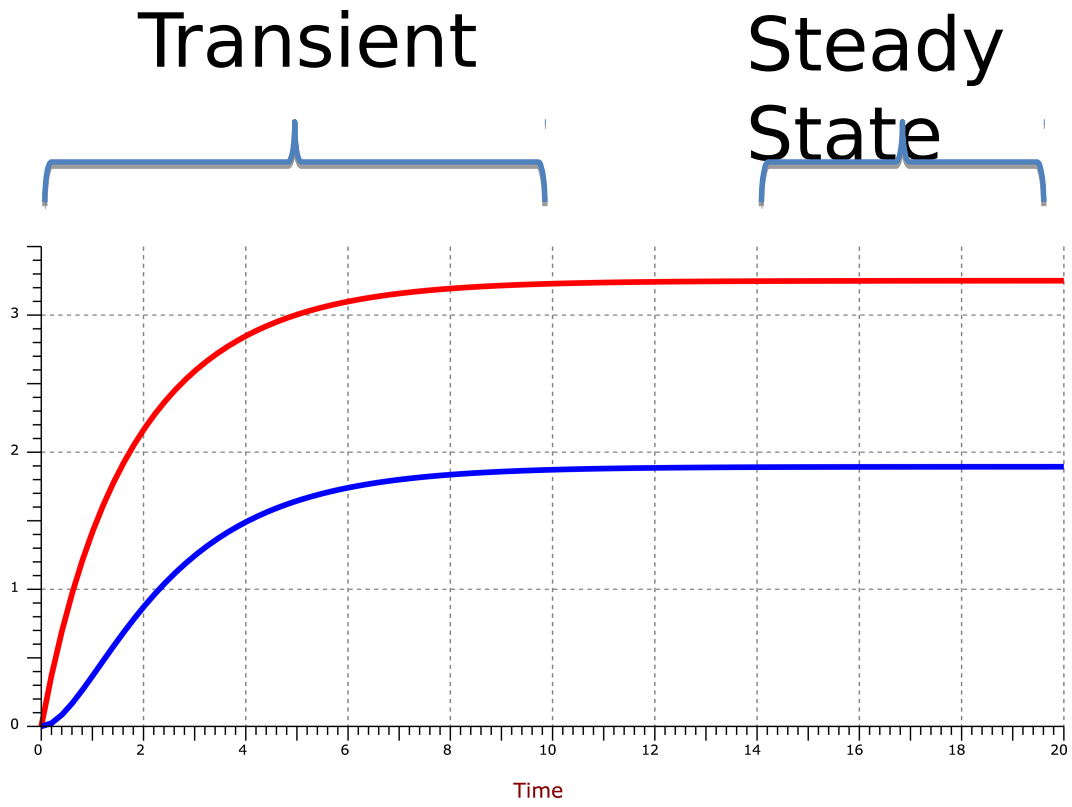
See “Enzyme Kinetics for Systems Biology” for Details

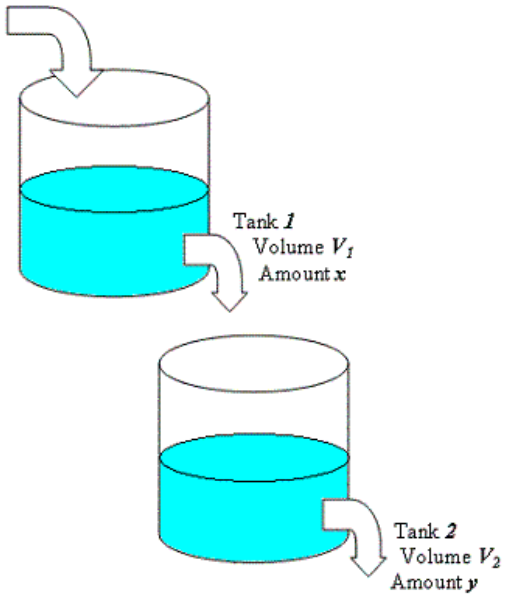
Boundary and Floating Species



A Boundary Species is under the direct control of the modeler

Transients and Steady State





Hands On Exercises

Telluriu
m

Closed System

uild a model of a closed system: Xo -> S1 -> S2 -> X1

```
-> S1   v = k1*Xo - k2*S1
-> S2   v = k3*S1 - k3*S2
-> X1   v = k5*S2 - k6*X1

= 4;   X1 = 0;
= 1.2; k2 = 0.45;
= 0.56; k4 = 0.2;
= 0.89; k6 = 0;
```

uestions:

Carry out a simulation and plot the time course for the syst
t = 0 to t = 50.
Once the system settles down what is the net flux
through the pathway?

Coffee Break

System Quantities

1. Variables:
   State Variables, Dynamical Variables, Floating Species

In principle only indirectly under the control of the Experimentalist. Determined by the system.

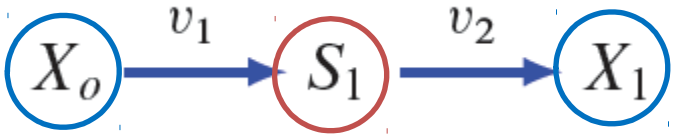
2. Parameters:
   Kinetic Constants, Boundary Species (fixed)

In principle under the direct control of the experimentalist

Steady State

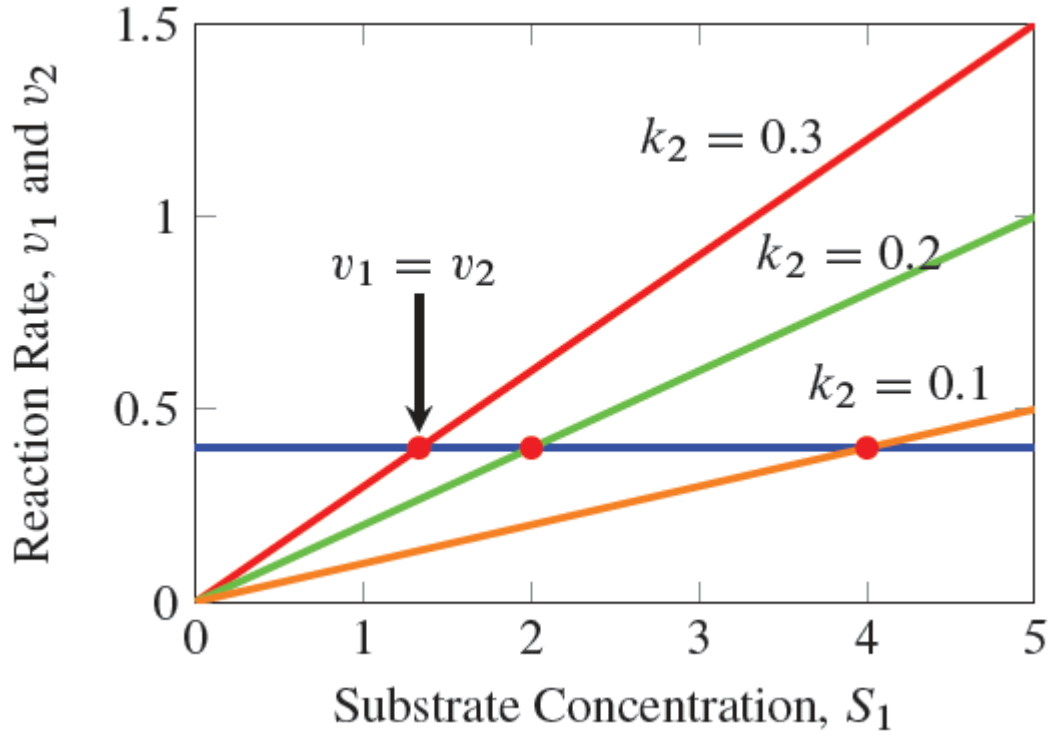
The steady state is one of the most important states to consider in a dynamical model. In the literature it is also sometimes referred to as the stationary solution or state, singular points, fixed points, or even equilibrium. We will

The steady state is the primary reference point from which to consider a model's behavior. At steady state, the concentrations of all molecular species are constant and there is a net flow of mass through the network.



$$v_1 = k_1 X_o$$
$$v_2 = k_2 S_1$$

Steady State



Steady State

$$\frac{dS_1}{dt} = k_1 X_o - k_2 S_1$$

$$dS_1/dt = 0,$$

$$S_1 = \frac{k_1 X_o}{k_2}$$

$$J = k_2 \frac{k_1 X_o}{k_2} = k_1 X_o$$

Open System

rn the close system you build into an open system by fixing
 and X1.

uestions:

1. Carry out a simulation and plot the time course for the syst
   t = 0 to t = 50.
2. Once the system settles down what is the net flux
   through the pathway?

Open System, Steady State

`r.steadystate();`

`This method returns a single number.`

This number indicates how close the solution is to the steady state.

Numbers < 1E-5 usually indicate it has found a steady state.

Confirm using **`print r.dv()`** <- prints rates of change

Useful Model Variables

```
r.dv()  <- returns the rates of change vector dx/dt

r.sv()  <- returns vector of current floating species

           concentrations

r.fs()  <- returns list of floating species
             names (same order as sv)
```

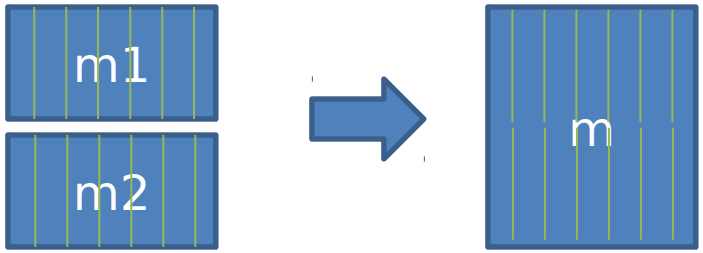
Useful Model Variables

```
r.pv() <- returns vector of all current parameter values

r.ps() <- returns list of kinetic parameter names

r.bs() <- returns list of boundary species names
```

Applying Perturbations in Tellurium



vstack ((m1, m2)) -> m
(augment by row)

```
import tellurium as te
import numpy

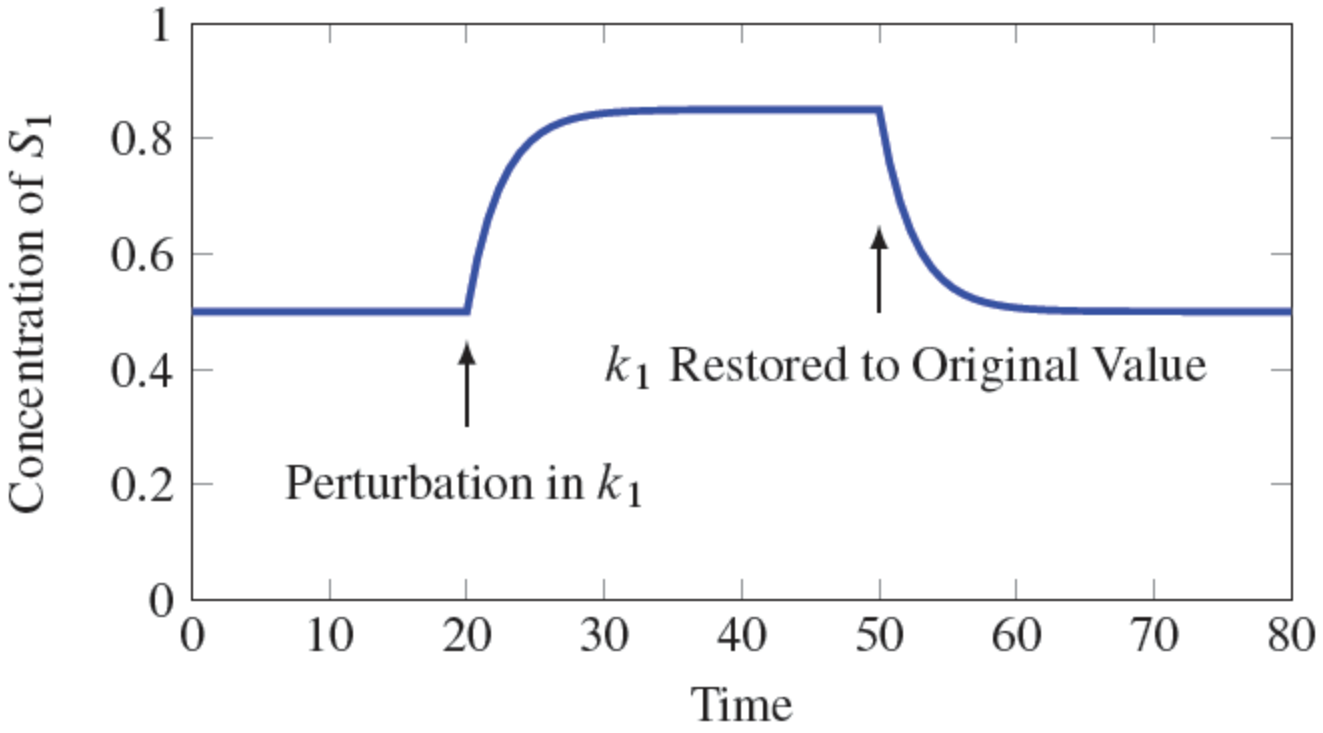
r = te.loada (```
    # Model Definition
    v1: $Xo -> S1;  k1*Xo;
    v2: S1 -> $w;   k2*S1;

    # Initialize constants
    k1 = 1; k2 = 1; S1 = 15; Xo = 1;
```)

# Time course simulation
m1 = r.simulate (0, 15, 100, ["Time","S1"]);
r.model.k1 = r.model.k1 * 6;
m2 = r.simulate (15, 40, 100, ["Time","S1"]);
r.model.k1 = r.model.k1 / 6;
m3 = r.simulate (40, 60, 100, ["Time">,"S1"]);

m = numpy.vstack ((m1, m2, m3)); # Merge data
r.plot (m)
```
```

Perturbations to Parameters



Perturbations to Variables

```
import tellurium as te
import numpy

r = te.loada ('''
   $Xo -> S1; k1*Xo;
   S1 -> $X1; k2*S1;

   k1 = 0.2; k2 = 0.4; Xo = 1; S1 = 0.5;
''')

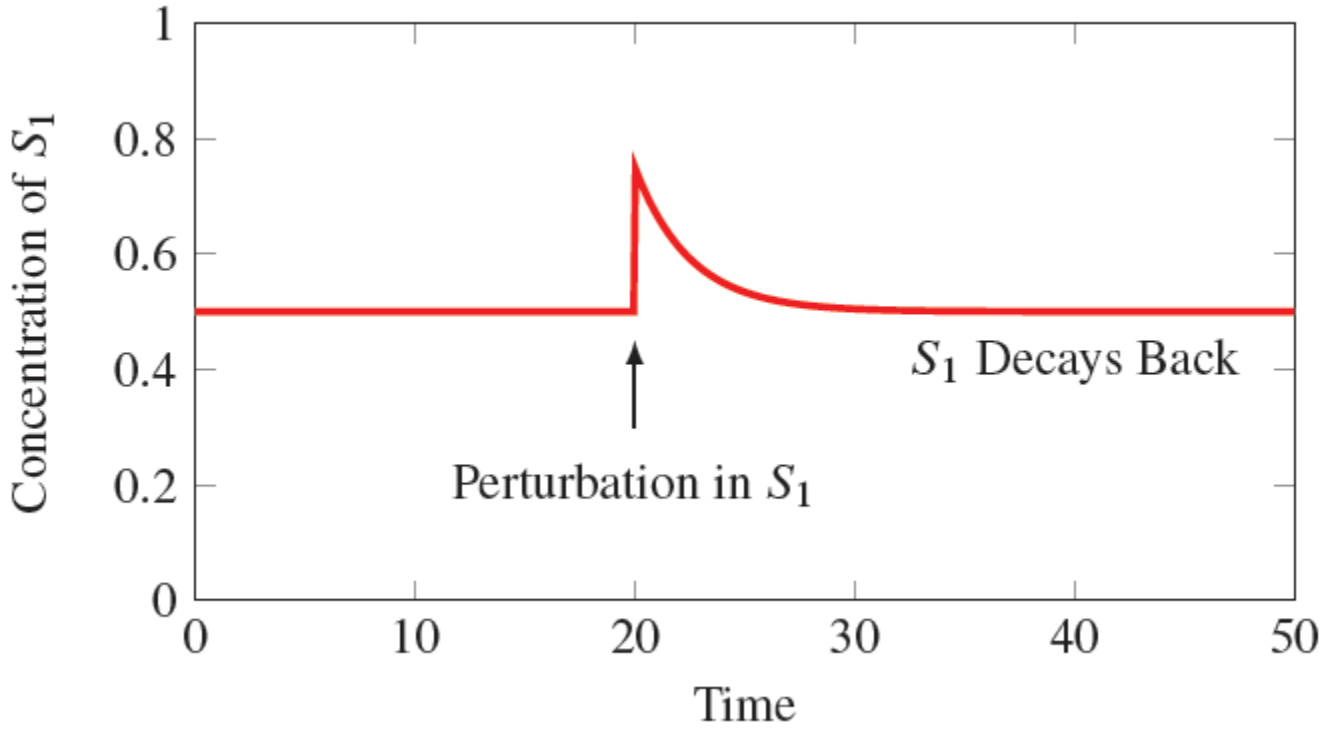
# Simulate the first part up to 20 time units
m1 = r.simulate (0, 20, 100, ["time", "S1"]);

# Perturb the concentration of S1 by 0.35 units
r.model.S1 = r.model.S1 + 0.35;

# Continue simulating from last end point
m2 = r.simulate (20, 50, 100, ["time", "S1"]);

# Merge and plot the two halves of the simulation
r.plot (numpy.vstack ((m1, m2)));
```

Perturbations to Variables



More on Plotting

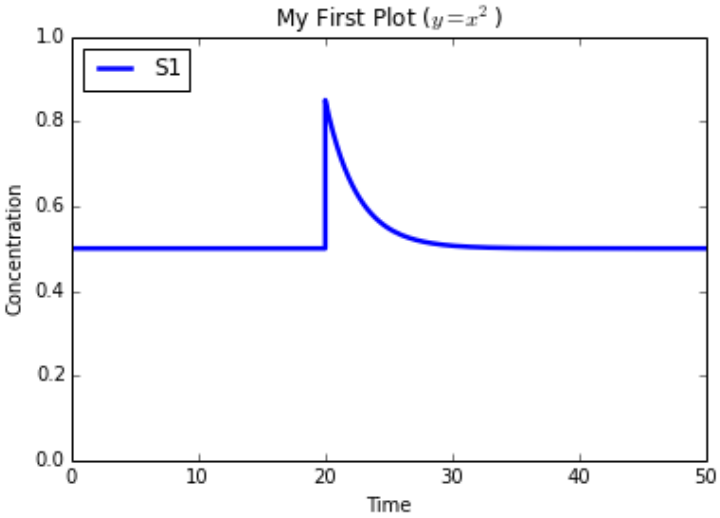
```
import tellurium as te
import numpy
import matplotlib.pyplot as plt

r = te.loada ('''
   $Xo -> S1; k1*Xo;
   S1 -> $X1; k2*S1;

   k1 = 0.2; k2 = 0.4; Xo = 1; S1 = 0.5;
''')

# Simulate the first part up to 20 time units
m1 = r.simulate (0, 20, 100, ["time", "S1"]);
r.model.S1 = r.model.S1 + 0.35;
m2 = r.simulate (20, 50, 100, ["time", "S1"]);

plt.ylim ((0,1))
plt.xlabel ('Time')
plt.ylabel ('Concentration')
plt.title ('My First Plot ($y = x^2$)')
r.plot (numpy.vstack ((m1, m2)));
```



Three Important Plot Commands

```
r.plot (result)  # Plots a legend

te.plotArray (result) # No legend

te.setHold (True) # Overlay plots
```

Plotting Overlay Example

```
import tellurium as te
import numpy
import matplotlib.pyplot as plt

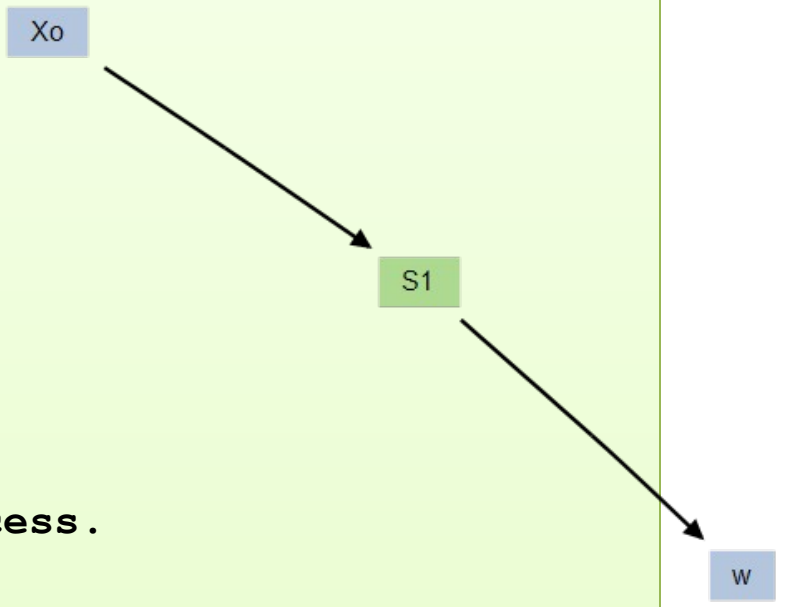

# model Definition
r = te.loada ('''
     v1: $Xo -> S1;  k1*Xo;
     v2: S1 -> $w;   k2*S1;

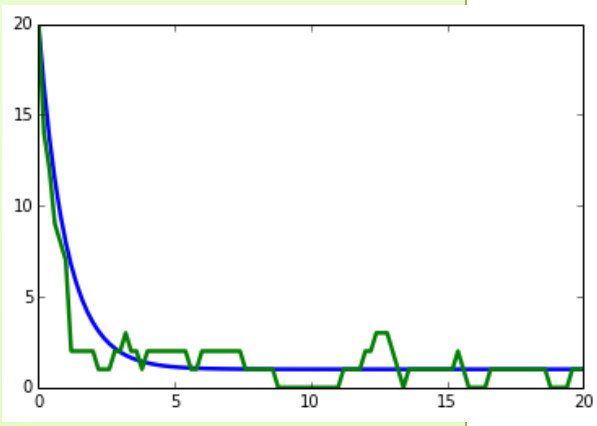
     //initialize.  Deterministic process.
     k1 = 1; k2 = 1; S1 = 20; Xo = 1;
''')

m1 = r.simulate (0,20,100);

# Stochastic process
r.resetToOrigin()
m2 = r.gillespie (0, 20, 100, ['time', 'S1'])

# plot all the results together
te.setHold (True)
te.plotArray (m1)
te.plotArray (m2)
```





Specifying Events

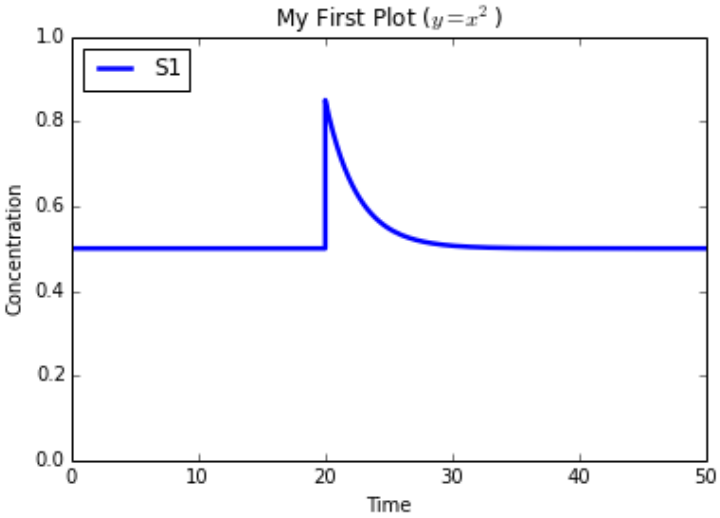
```
import tellurium as te
import numpy
import matplotlib.pyplot as plt
import roadrunner

roadrunner.Config.setValue (roadrunner.Config.LOADSBMLOPTIONS_CONSERVED_MOIETIES, False)
r = te.loada ('''
   $Xo -> S1; k1*Xo;
   S1 -> $X1; k2*S1;

   k1 = 0.2; k2 = 0.4; Xo = 1; S1 = 0.5;
   at (t > 20): S1 = S1 + 0.35
''')

# Simulate the first part up to 20 time units
m = r.simulate (0, 20, 100, ["time", "S1"]);

plt.ylim ((0,1))
plt.xlabel ('Time')
plt.ylabel ('Concentration')
plt.title ('My First Plot ($y = x^2$)')
r.plot (numpy.vstack ((m1, m2)));
```



Why the disturbance is stable

$$dS_1/dt = k_1 X_o - k_2 S_1$$

If the system is at steady state, let us make a small perturbation to the steady state concentration of S_1, δS_1 and ask what is the rate of change of $S_1 + \delta S_1$ as a result of this perturbation, that is what is $d(S_1 + \delta S_1)/dt$? The new rate of change equation is rewritten as follows:

$$\frac{d(S_1 + \delta S_1)}{dt} = k_1 X_o - k_2 (S_1 + \delta S_1)$$

If we insert the solution for S_1 (equation 5.1) into the above equation we are left with:

$$\frac{d\delta S_1}{dt} = -k_2 \delta S_1$$

In other words the rate of change of the disturbance itself, δS_1 is negative, that is, the system attempts to reduce the disturbance

Solving ODEs

What if I only have a set of ODES?

dy/dt = -k*y

```
r = te.loada (‘’’
        y’ = -k*y;  # Note the apostrophe

        y = 1; k = 0.2;
‘’’)
```

Solving ODEs

When you run simulate **make sure** you specify the ode variables!

```
r = te.loada ('''
        y' = -k*y;  # Note the apostrophe

        y = 1; k = 0.2;
''')

result = r.simulate (0, 10, 50,['time', 'y'])
r.plot (result)
```

Simulate the Chaotic Lorenz System

```
Simulate the Lorenz System.

dx/dt = sigma*(y - x)
dy/dt = x*(rho - z) - y
dz/dt = x*y - beta*z



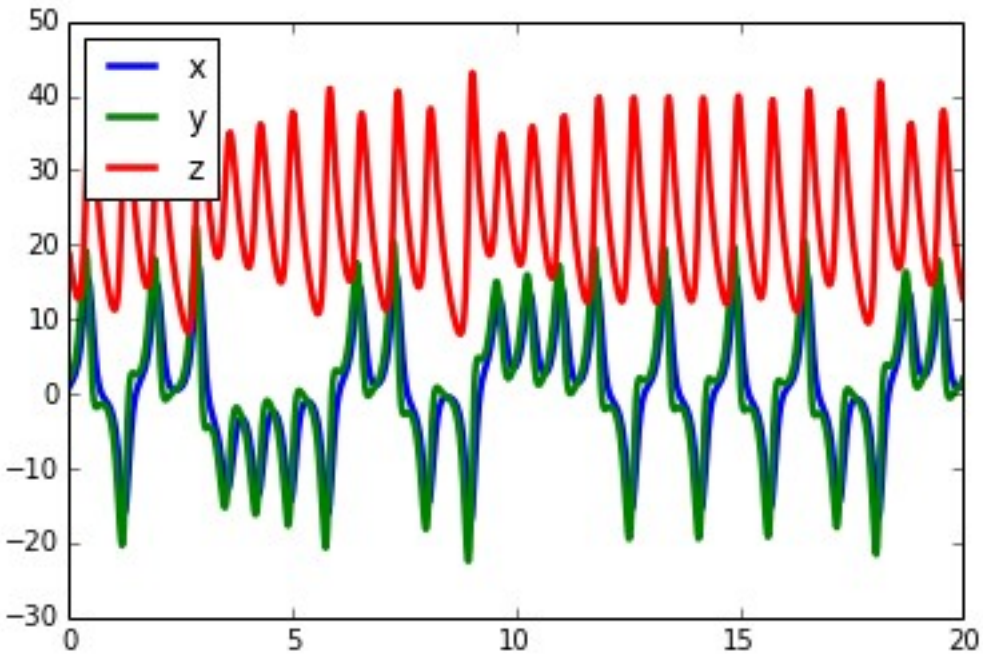
x = 0.96259;  y = 2.07272;  z = 18.65888;

sigma = 10;  rho = 28; beta = 2.67;

Simulate t=0 to t=20

http://en.wikipedia.org/wiki/Lorenz_system
```

Solving ODEs




```
import tellurium as te

r = te.loada ('''
      x' = sigma*(y - x);
      y' = x*(rho - z) - y;
      z' = x*y - beta*z;

      x = 0.96259;  y = 2.07272;  z = 18.65888;

      sigma = 10;  rho = 28; beta = 2.67;
''')

result = r.simulate (0, 20, 1000, ['time', 'x', 'y', 'z'])
r.plot (result)
```

How Can I Exchange Models?

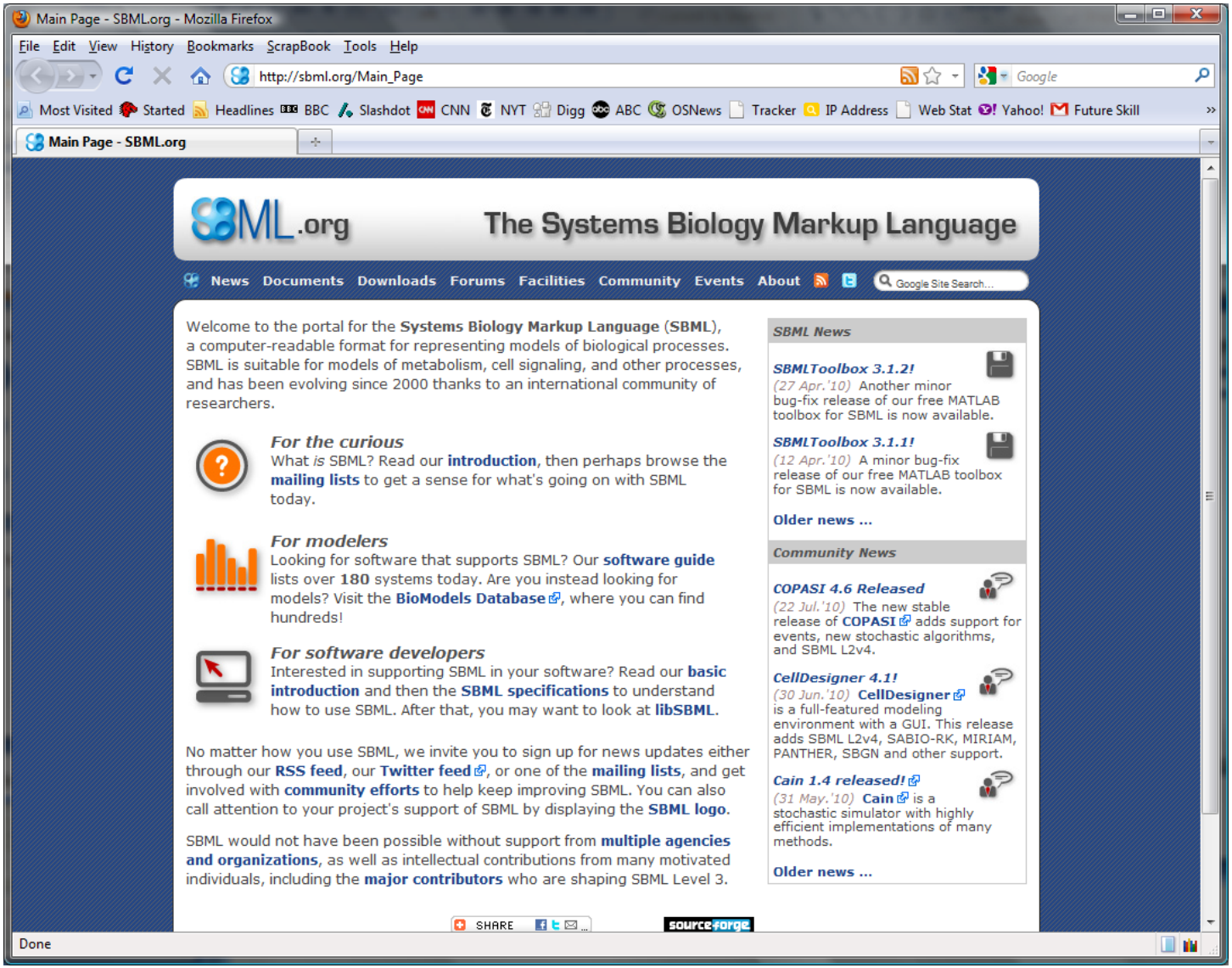
L (Systems Biology Markup Language): de facto standard for senting cellular networks. A large number (>200) of tools support SBML.

ML: Stores models in mathematical form, therefore is quite general, but gical information is lost. Not possible to reconstruct network. Less than a full of tools support CellML

l: A proposed standard for visually representing cellular networks. No stent format has yet been devised which limits its use in software.

ab: Proprietary math based scripting language

SBML


Main Page - SBML.org - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://sbml.org/Main_Page
Most Visited Started Headlines BBC Slashdot CNN NYT Digg ABC OSNews Tracker IP Address Web Stat Yahoo! Future Skill
Main Page - SBML.org
SBML.org
The Systems Biology Markup Language
News Documents Downloads Forums Facilities Community Events About
Welcome to the portal for the Systems Biology Markup Language (SBML), a computer-readable format for representing models of biological processes. SBML is suitable for models of metabolism, cell signaling, and other processes, and has been evolving since 2000 thanks to an international community of researchers.
For the curious
What is SBML? Read our introduction, then perhaps browse the mailing lists to get a sense for what's going on with SBML today.
For modelers
Looking for software that supports SBML? Our software guide lists over 180 systems today. Are you instead looking for models? Visit the BioModels Database, where you can find hundreds!
For software developers
Interested in supporting SBML in your software? Read our basic introduction and then the SBML specifications to understand how to use SBML. After that, you may want to look at libSBML.
No matter how you use SBML, we invite you to sign up for news updates either through our RSS feed, our Twitter feed, or one of the mailing lists, and get involved with community efforts to help keep improving SBML. You can also call attention to your project's support of SBML by displaying the SBML logo.
SBML would not have been possible without support from multiple agencies and organizations, as well as intellectual contributions from many motivated individuals, including the major contributors who are shaping SBML Level 3.
SBML News
SBMLToolbox 3.1.2!
(27 Apr.'10) Another minor bug-fix release of our free MATLAB toolbox for SBML is now available.
SBMLToolbox 3.1.1!
(12 Apr.'10) A minor bug-fix release of our free MATLAB toolbox for SBML is now available.
Older news ...
Community News
COPASI 4.6 Released
(22 Jul.'10) The new stable release of COPASI adds support for events, new stochastic algorithms, and SBML L2v4.
CellDesigner 4.1!
(30 Jun.'10) CellDesigner is a full-featured modeling environment with a GUI. This release adds SBML L2v4, SABIO-RK, MIRIAM, PANTHER, SBGN and other support.
Cain 1.4 released!
(31 May.'10) Cain is a stochastic simulator with highly efficient implementations of many methods.
Older news ...
SHARE
sourceforge
Done

Systems Biology Markup Language

Originally developed in 2000 to allow users to exchange models between the small number of simulators that existed at that time.

Since then it has become the de facto standard for model exchange in systems biology

SBML represents models using XML by describing:

1. Compartment
2. Molecular Species
3. Chemical and Enzymatic Reactions (including gene regulatory)
4. Parameters
5. Kinetic Rate Laws
6. Additional Mathematical Equations when necessary

Systems Biology Markup Language

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created by XMLPrettyPrinter on 7/30/2012 -->
<sbml xmlns = "http://www.sbml.org/sbml/level2" level = "2" version = "1">
  <model id = "cell">
    <listOfCompartments>
      <compartment id = "compartment" size = "1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id = "S1" boundaryCondition = "true" initialConcentration = "1" compartment =
"compartment"/>
      <species id = "S3" boundaryCondition = "true" initialConcentration = "0" compartment =
"compartment"/>
      <species id = "S2" boundaryCondition = "false" initialConcentration = "1.33"
compartment = "compartment"/>
    </listOfSpecies>
    <listOfParameters>
      <parameter id = "k1" value = "3.4"/>
      <parameter id = "k2" value = "2.3"/>
    </listOfParameters>
    <listOfReactions>
      <reaction id = "J1" reversible = "false">
        <listOfReactants>
          <speciesReference species = "S1" stoichiometry = "1"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species = "S2" stoichiometry = "1"/>
        </listOfProducts>
```

Systems Biology Markup Language

```
<kineticLaw>
        <math xmlns = "http://www.w3.org/1998/Math/MathML">
          <apply>
            <times/>
            <ci>
                k1
            </ci>
            <ci>
                S1
            </ci>
          </apply>
        </math>
      </kineticLaw>
    </reaction>
    <reaction id = "J2" reversible = "false">
      <listOfReactants>
        <speciesReference species = "S2" stoichiometry = "1"/>
      </listOfReactants>
      <listOfProducts>
        <speciesReference species = "S3" stoichiometry = "1"/>
      </listOfProducts>
      <kineticLaw>
        <math xmlns = "http://www.w3.org/1998/Math/MathML">
          <apply>
            <times/>
            <ci>
                k2
```

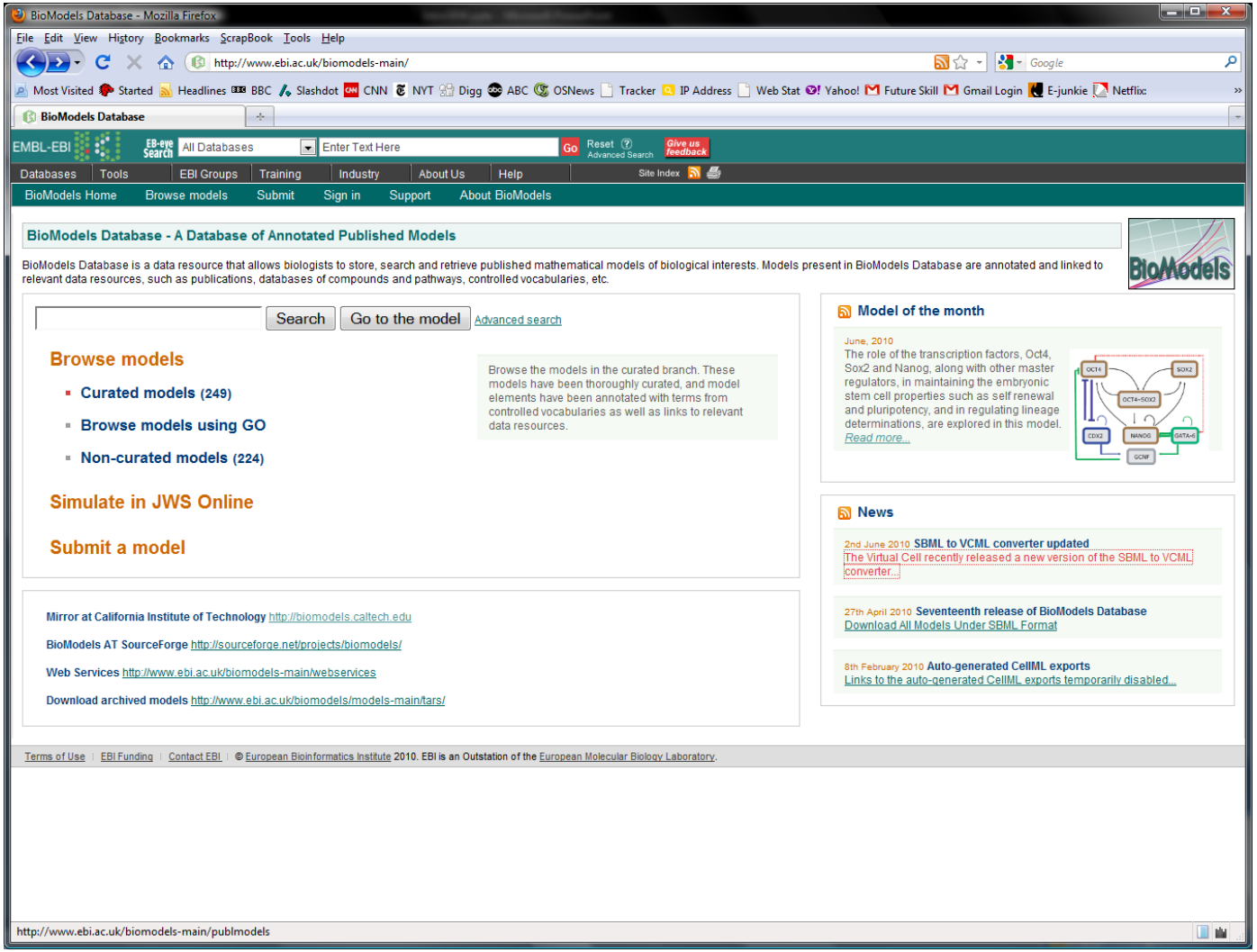
Model Repositories

421 Curated models as of July 2012

433 Non-curated Models.

Biomodels.net

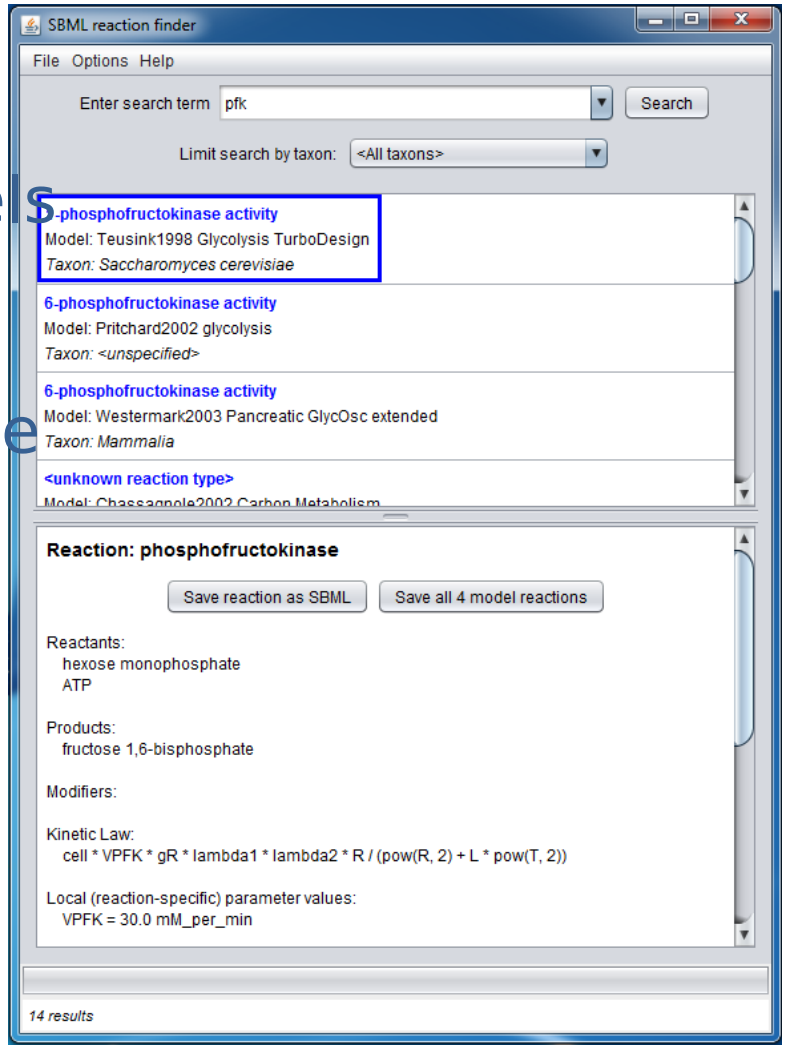
At the EBI near Cambridge, UK



Parts Repository: Max Neals

This tools decomposes all biomodels into their constituent parts.

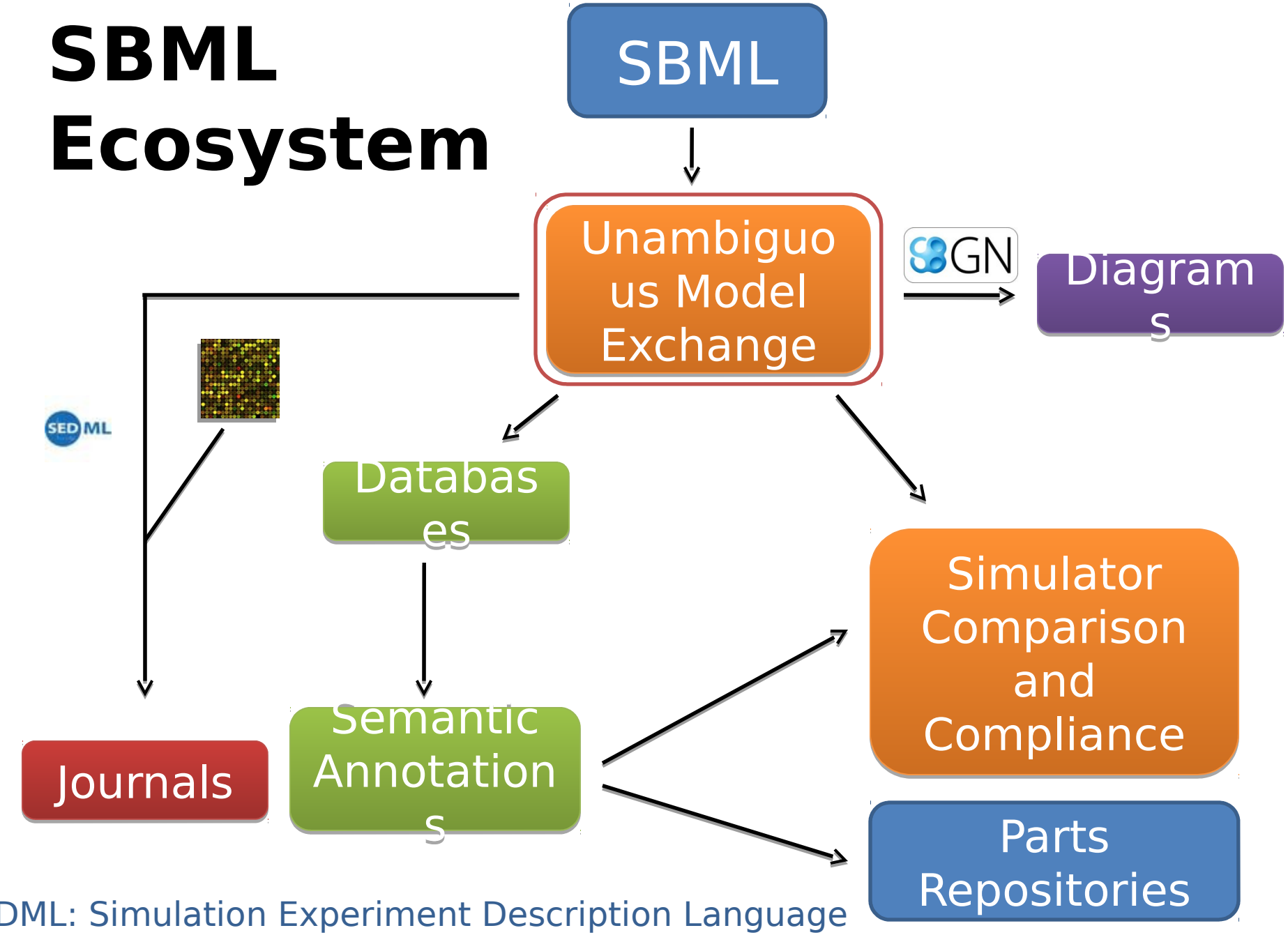
For example, search for pfk to locate all pfk parts in the biomodels database.



See the following web site for details:

http://sites.google.com/site/semanticsofbiologicalprocesses/projects/sbmlrxnfinder

SBML Ecosystem



DML: Simulation Experiment Description Language
GN : Systems Biology Graphical Notation

Exporting/Importing Models

Importing:

1. **Antimony (using loada)**

2. **SBML (using roadrunner.RoadRunner (sbml model)**

Exporting:

1. **r.getAntimony()**
2. **r.getSBML()**
3. **r.getMatlab()**

Exercise

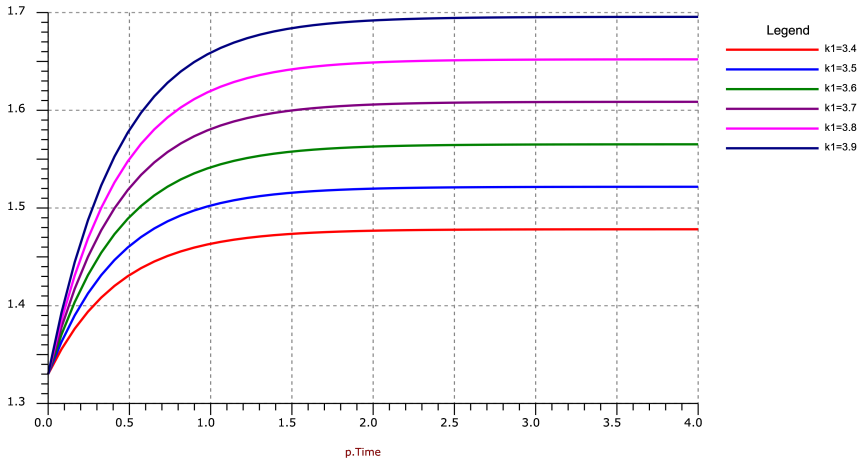
Build a simple model and export the SBML and Matlab

Parameter Scan

```
# Parameter Scan
import tellurium as te
import numpy

r = te.loada ('''
    J1: $X0 -> S1; k1*X0;
    J2: S1 -> $X1; k2*S1;

    X0 = 1.0; S1 = 0.0; X1 = 0.0;
    k1 = 0.4; k2 = 2.3;
''')
```




```
m = r.simulate (0, 4, 100, ["Time", "S1"])
for i in range (0,4):
    r.model.k1 = r.model.k1 + 0.1
    r.reset()
    m = numpy.hstack ((m, r.simulate (0, 4, 100, ['S1'])))

#m[:,1] *= 5
te.plotArray (m)
```