Systems Biology with R

Bairong Shen

Center for Systems Biology

Soochow University, Suzhou,China

Outline

A. Basic R programming (demo)
B. Basic concepts for ODE and biological kinetics
C. Introduction to SBML and R packages for systems biology
D. Case study

A. Basic R programming

Demo 1

B. Basic concepts for ODE and biological kinetics

1. Euler's Method
2. Runge-Kutta 4^{th} Order Method
3. R package: odesolve
4. Exercise

1. Euler's Method

Leonhard Paul Euler (April 15, 1707 – September 18 1783) was a pioneering Swiss mathematician and physicist, who spent most of his life in Russia and Germany. He published more papers than any other mathematician of his time.

Euler's Method

$$\frac{dy}{dx} = f(x,y), y(0) = y_0$$

$$\text{Slope} = \frac{y_1 - y_0}{x_1 - x_0}$$

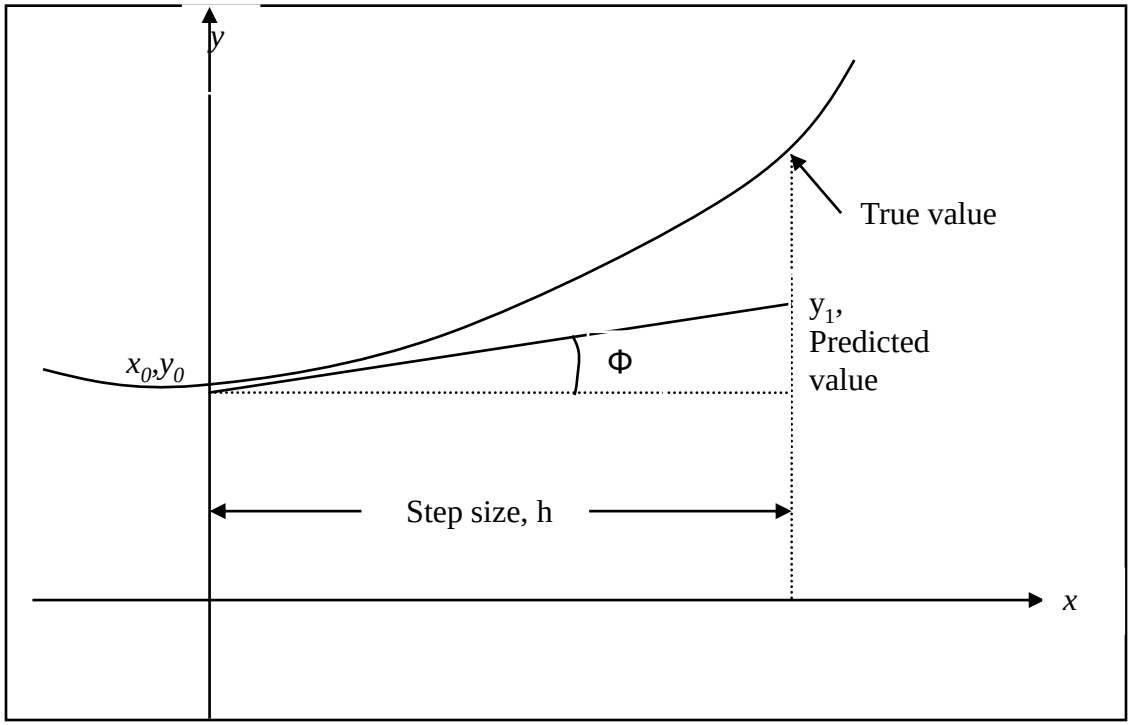
$$= f(x_0, y_0)$$

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0)$$

$$= y_0 + f(x_0, y_0)h$$

$$y_{i+1} = y_i + f(x_i, y_i)h$$

$$h = x_{i+1} - x_i$$



Graphical interpretation of the first step of Euler's method

How to write ODE

How does one write a first order differential equation in the form of

$$\frac{dy}{dx} = f(x, y)$$

Example

$\frac{dy}{dx} + 2y = 1.3e^{-x}, y(0) = 5$ is rewritten as $\frac{dy}{dx} = 1.3e^{-x} - 2y, y(0) = 5$

In this case

$$f(x, y) = 1.3e^{-x} - 2y$$

Errors in Euler's Method

Euler's method has large errors.
This can be illustrated using Taylor series.

$$y_{i+1} = y_i + \left.\frac{dy}{dx}\right|_{x_i,y_i}(x_{i+1}-x_i) + \left.\frac{1}{2!}\frac{d^2y}{dx^2}\right|_{x_i,y_i}(x_{i+1}-x_i)^2 + \left.\frac{1}{3!}\frac{d^3y}{dx^3}\right|_{x_i,y_i}(x_{i+1}-x_i)^3 + \ldots$$

$$y_{i+1} = y_i + f(x_i,y_i)(x_{i+1}-x_i) + \frac{1}{2!}f'(x_i,y_i)(x_{i+1}-x_i)^2 + \frac{1}{3!}f''(x_i,y_i)(x_{i+1}-x_i)^3 + \ldots$$

As you can see the first two terms of the Taylor series

$y_{i+1} = y_i + f(x_i, y_i)h$ are the Euler's method.

The true error in the approximation is given by

$$E_t = \frac{f'(x_i,y_i)}{2!}h^2 + \frac{f''(x_i,y_i)}{3!}h^3 + \ldots$$

$$E_t \propto h^2$$

2. Runge-Kutta 4th Order Method

MARTIN WILHELM KUTTA
1867-1944

$$\frac{dy}{dx} = f(x, y), y(0) = y_0$$

The 4th order method is given by

$$y_{i+1} = y_i + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)h$$

$$k_1 = f\left(x_i, y_i\right)$$

$$k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right)$$

$$k_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right)$$

$$k_4 = f\left(x_i + h, y_i + k_3 h\right)$$

Precision of the 4th order R-K Method (Theorem)

Assume that $y = y(t)$ is the solution to the I.V.P. $y' = f(t, y)$ with $y(t_0) = y_0$. If $y(t) \in C^5[t_0, b]$ and $\{(t_k, y_k)\}_{k=0}^{m}$ is the sequence of approximations generated by the Runge-Kutta method of order 4, then at each step, the local truncation error is of the order $o(h^5)$, and the overall global truncation error e_k is of the order

$$|e_k| = |y(t_k) - y_k| = O(h^4), \text{ for } k = 1, 2, \ldots, m.$$

The error at the right end of the interval is called the final global error

$$E(y(b), h) = |y(b) - y_m| = O(h^4).$$

3. R package: odesolve

- **odesolve: Solvers for Ordinary Differential Equations**
- This package provides an interface for the ODE solver lsoda.
- ODEs are expressed as R functions or as compiled code.
  - lsoda Solve System of ODE (ordinary differential equation)s.

R function: lsode

$$\frac{dy}{dt} = f(t, y), y(0) = y_0$$

dx/dt = 8x from t = 1 to t = 8 with initial condition x(1) = 5

- lsoda(y, times, func, parms, rtol, atol, tcrit=NULL, jacfunc=NULL, verbose=FALSE, dllname=NULL, hmin=0, hmax=Inf)
  - y: the initial values for the ode system.
  - times: times at which explicit estimates for y are desired. The first value in times must be the initial time.
  - func: the user-supplied function that computes the values of the derivatives in the ode system (the *model definition*) at time t. **it must be called as: yprime = func(t, y, parms).**

R function: lsode

$$\frac{dy}{dt} = f(t, y), y(0) = y_0$$

dx/dt = 8x from t = 1 to t = 8 with initial condition x(1) = 5

- lsoda(y, times, func, parms, rtol, atol, tcrit=NULL, jacfunc=NULL, verbose=FALSE, dllname=NULL, hmin=0, hmax=Inf)
  - parms: any parameters used in func that should be modifiable without rewriting the function.
  - rtol:relative error tolerance, either a scalar or an array as long as y.
  - atol: absolute error tolerance, either a scalar or an array as long as y.

Practice makes perfect

4. Exercise

1. Numerically solve ODE dx/dt = 8x from t = 1 to t = 8 with initial condition x(1) = 5

2. Numerically solve system of ODEs

   dw/dt = 8w,

   dz/dt =3w + 7z

   from t = 1 to t = 8 with initial conditions w(1) = 5, z(1) = 6.5

   (Ex. 3~5 skipped)

Steps for coding

- First implement function: return a list, multiple variables
- Then do **y**=lsoda(y0, t, f, NA)
  - The first column of **y**, namely y[,1] contains the time values;
  - the second , third …columns y[,2], y[,3]… contains the corresponding function values: multiple variables
  - Here we have two ODEs, in systems biology it could be 100s to 1000s of ODEs.

Steps for coding

- First implement function: return a list, multiple variables
- Then do **y**=lsoda(y0, t, f, NA)
  - The first column of **y**, namely y[,1] contains the time values;
  - the second , third …columns y[,2], y[,3]… contains the corresponding function values: multiple variables

```
f = function(t, x, parms) {
return(list(8*x))
}
```

```
# Then to obtain solution values
# at times 1; 1.1; 1.2; : : : ; 7. 9; 8.

library(odesolve)
t<-seq(1,8,0.1);
y=lsoda(5, t, f,NA)
```

Demo 2

C. Introduction to SBML and R packages for systems biology

- What is SBML

  A machine-readable format for representing computational models in systems biology

  - Expressed in XML using an XML Schema
  - Intended for software tools not for humans (Although it is text-based and therefore readable)

  Intended to be a tool-neutral exchange language for software applications in systems biology

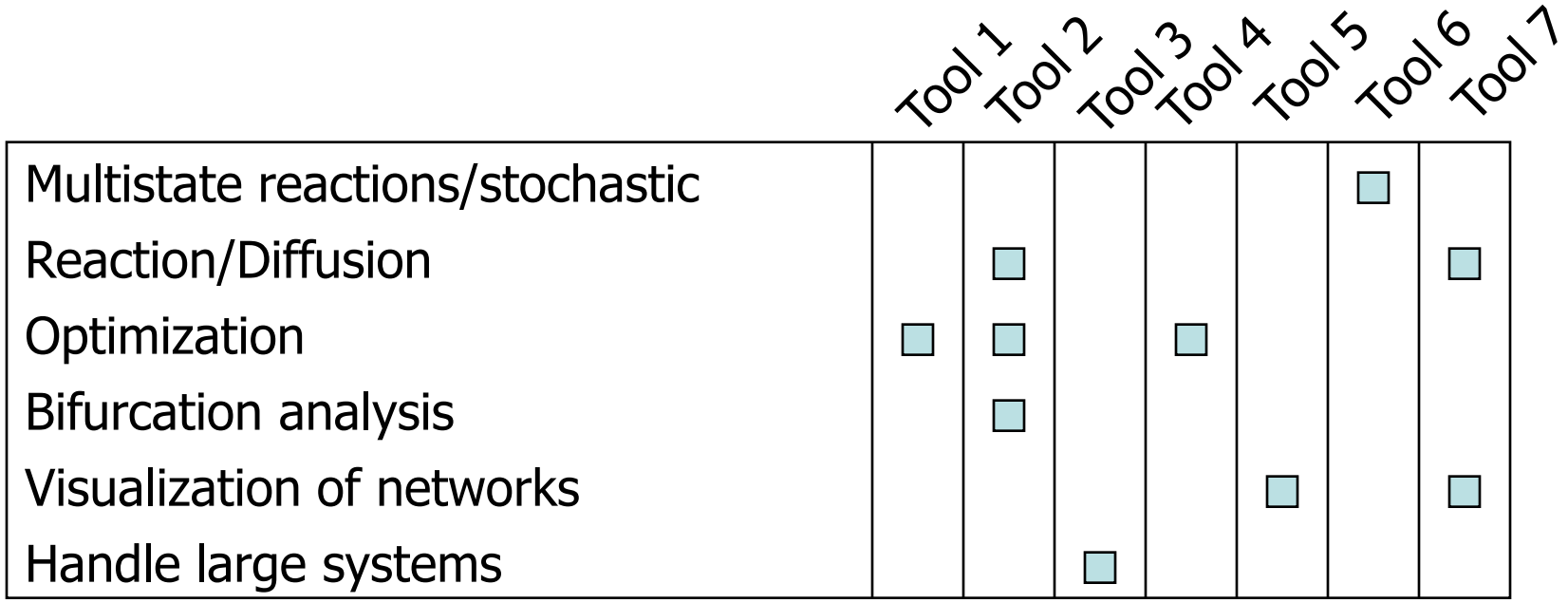
  - Simply an enabling technology

- R packages for systems biology: SBMLR

Why SBML?

- Modeling, simulation & analysis are critical
  - Huge volumes of data
  - Many disparate findings
- Rapid rate of software tool development
  - Roles: data filtering, model creation, model simulation
  - Many groups are creating many tools
    - Different packages have different niche strengths reflecting expertise & preferences of the group
    - Strengths are often complementary to those of other packages

Complementary Strengths of Tools



Goal & Approach

Systems Biology Workbench goal: to provide software infrastructure that

- Enables sharing of simulation/analysis software & models
- Enables collaboration between software developers

Two-pronged approach:

- Develop a common model exchange language
  - SBML: Systems Biology Markup **Language**
- Develop an environment that enables tools to interact
  - SBW: Systems Biology **Workbench**

Structure of Models Expressed in SBML

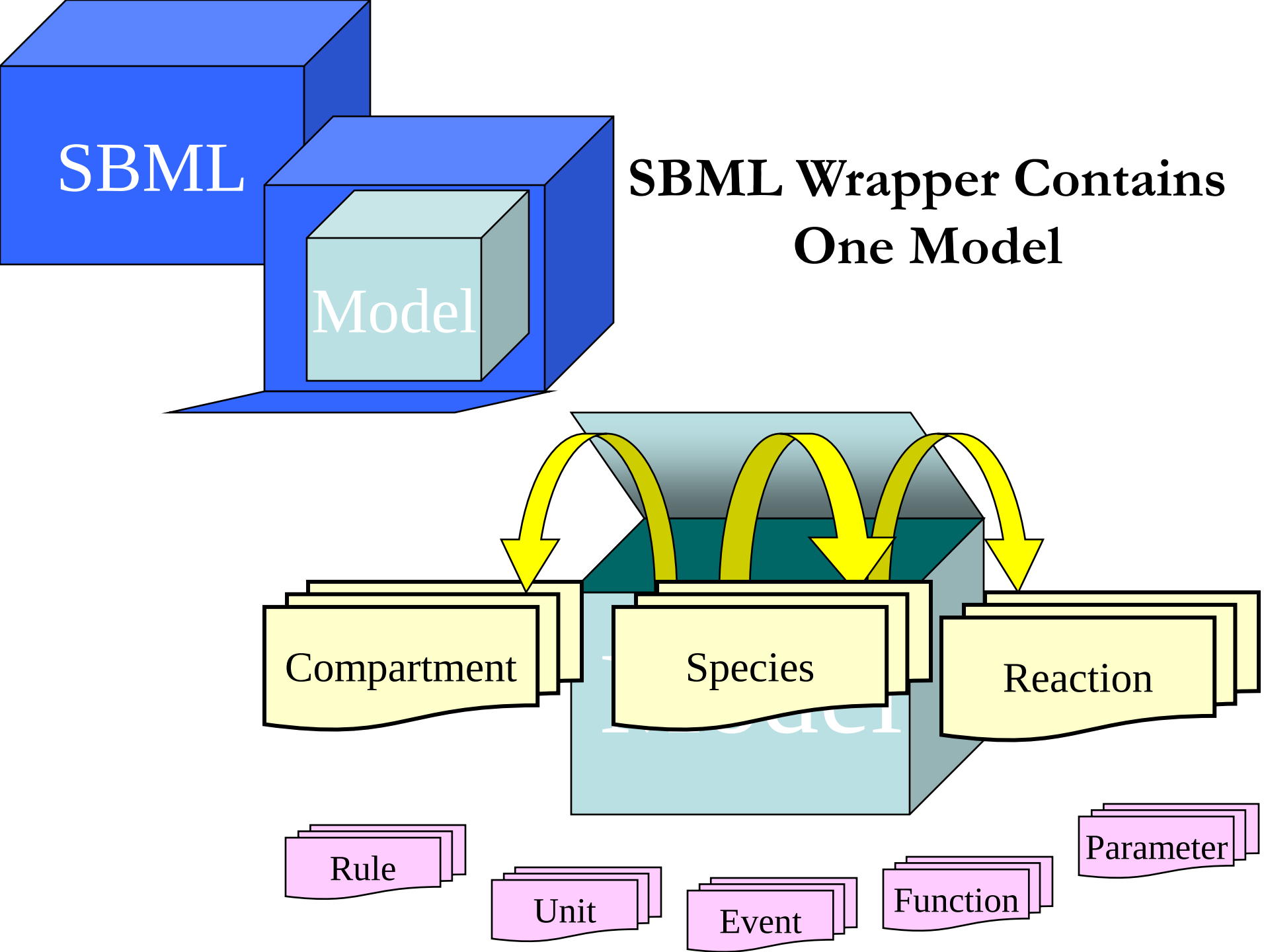
Reaction networks described by list of components:

Beginning of SBML model definition

- List of function definitions (new in Level 2)
- List of unit definitions
- List of compartments
- List of species
- List of parameters
- List of rules
- List of reactions
- List of events (new in Level 2)

End of SBML model definition

SBML Wrapper Contains One Model



Compartment List

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns = "http://www.sbml.org/sbml/level1" level = "1" version = "1">
  <model name = "ATitle">
    <listOfCompartments>
    </listOfCompartments>
    <listOfSpecies>
    </listOfSpecies>
    <listOfReactions>
    </listOfReactions>
  </model>
</sbml>
```

Species List

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns = "http://www.sbml.org/sbml/level1" level = "1" version =
    "1">
  <model name = "ATitle">
    <listOfCompartments>
    </listOfCompartments>
    <listOfSpecies>
    </listOfSpecies>
    <listOfReactions>
    </listOfReactions>
  </model>
</sbml>
```

Reaction List

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns = "http://www.sbml.org/sbml/level1" level = "1" version =
    "1">
  <model name = "ATitle">
    <listOfCompartments>
    </listOfCompartments>
    <listOfSpecies>
    </listOfSpecies>
    <listOfReactions>
    </listOfReactions>
  </model>
</sbml>
```

Reactions in Close-up

<listOfReactions>

```
<reaction id="R01">
    <listOfReactants>
        <speciesReference species="X0" stoichiometry="1"/>
    <listOfReactants>
    <listOfProducts>
        <speciesReference species="S1" stoichiometry="1"/>
    </listOfProducts>
    <listOfModifiers>
        <modifierSpeciesReference species="M1"/>
    </listOfModifiers>
    <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
            <times/>
                <ci> k1 </ci>
                <ci> X0 </ci>
                <ci> M1 </ci>
            </apply>
        </math>
        <listOfParameters>
            <parameter id="k1" value="0"/>
        </listOfParameters>
    </kineticLaw>
</reaction>
```

</listOfReactions>

Reaction Wrapper

```
<listOfReactions>
  <reaction id="R01">
    <listOfReactants>
      <speciesReference species="X0" stoichiometry="1"/>
    <listOfReactants>
    <listOfProducts>
      <speciesReference species="S1" stoichiometry="1"/>
    </listOfProducts>
    <listOfModifiers>
      <modifierSpeciesReference species="M1"/>
    </listOfModifiers>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
        <times/>
          <ci> k1 </ci>
          <ci> X0 </ci>
          <ci> M1 </ci>
        </apply>
      </math>
      <listOfParameters>
        <parameter id="k1" value="0"/>
      </listOfParameters>
    </kineticLaw>
  </reaction>
</listOfReactions>
```

Reactants

```
<listOfReactions>
    <reaction id="R01">
        <listOfReactants>
            <speciesReference species="X0" stoichiometry="1"/>
        <listOfReactants>
        <listOfProducts>
            <speciesReference species="S1" stoichiometry="1"/>
        </listOfProducts>
        <listOfModifiers>
            <modifierSpeciesReference species="M1"/>
        </listOfModifiers>
        <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                <times/>
                    <ci> k1 </ci>
                    <ci> X0 </ci>
                    <ci> M1 </ci>
                </apply>
            </math>
            <listOfParameters>
                <parameter id="k1" value="0"/>
            </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
```

Products

```
<listOfReactions>
    <reaction id="R01">
        <listOfReactants>
            <speciesReference species="X0" stoichiometry="1"/>
        <listOfReactants>
        <listOfProducts>
            <speciesReference species="S1" stoichiometry="1"/>
        </listOfProducts>
        <listOfModifiers>
            <modifierSpeciesReference species="M1"/>
        </listOfModifiers>
        <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                <times/>
                    <ci> k1 </ci>
                    <ci> X0 </ci>
                    <ci> M1 </ci>
                </apply>
            </math>
            <listOfParameters>
                <parameter id="k1" value="0"/>
            </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
```

Modifiers

```
<listOfReactions>
    <reaction id="R01">
        <listOfReactants>
            <speciesReference species="X0" stoichiometry="1"/>
        <listOfReactants>
        <listOfProducts>
            <speciesReference species="S1" stoichiometry="1"/>
        </listOfProducts>
        <listOfModifiers>
            <modifierSpeciesReference species="M1"/>
        </listOfModifiers>
        <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                <times/>
                    <ci> k1 </ci>
                    <ci> X0 </ci>
                    <ci> M1 </ci>
                </apply>
            </math>
            <listOfParameters>
                <parameter id="k1" value="0"/>
            </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
```

Kinetic Law

```
<listOfReactions>
    <reaction id="R01">
        <listOfReactants>
            <speciesReference species="X0" stoichiometry="1"/>
        <listOfReactants>
        <listOfProducts>
            <speciesReference species="S1" stoichiometry="1"/>
        </listOfProducts>
        <listOfModifiers>
            <modifierSpeciesReference species="M1"/>
        </listOfModifiers>
        <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                <times/>
                    <ci> k1 </ci>
                    <ci> X0 </ci>
                    <ci> M1 </ci>
                </apply>
            </math>
            <listOfParameters>
                <parameter id="k1" value="0"/>
            </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
```

MathML Describes Equation (Level 2)

```
<listOfReactions>
    <reaction id="R01">
        <listOfReactants>
            <speciesReference species="X0" stoichiometry="1"/>
        <listOfReactants>
        <listOfProducts>
            <speciesReference species="S1" stoichiometry="1"/>
        </listOfProducts>
        <listOfModifiers>
            <modifierSpeciesReference species="M1"/>
        </listOfModifiers>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
            <times/>
              <ci> k1 </ci>
              <ci> X0 </ci>
              <ci> M1 </ci>
            </apply>
          </math>
          <listOfParameters>
              <parameter id="k1" value="0"/>
          </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
```

Parameters Used in Equation

```
<listOfReactions>
    <reaction id="R01">
        <listOfReactants>
            <speciesReference species="X0" stoichiometry="1"/>
        <listOfReactants>
        <listOfProducts>
            <speciesReference species="S1" stoichiometry="1"/>
        </listOfProducts>
        <listOfModifiers>
            <modifierSpeciesReference species="M1"/>
        </listOfModifiers>
        <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                <times/>
                    <ci> k1 </ci>
                    <ci> X0 </ci>
                    <ci> M1 </ci>
                </apply>
            </math>
            <listOfParameters>
              <parameter id="k1" value="0"/>
            </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
```

Notes and Annotations

Notes

To be read by humans (e.g. comments)

Annotations

To be read by machines (e.g. tool-specific information)

Some Common Themes

SBML is useful as a common exchange format for transferring computational biochemical reaction models between software tools

- It is an intermediate, common-denominator format
- Therefore, it may not capture everything that every tool can represent the transformation may be lossy
  - (But: tools can add their own annotations)

It is *not* suited for representing experimental data

It is *not* suited for representing numerical results

It is *not* suited to be a database format for molecular databases

R packages for systems biology: SBMLR

Quick Intro to SBMLR

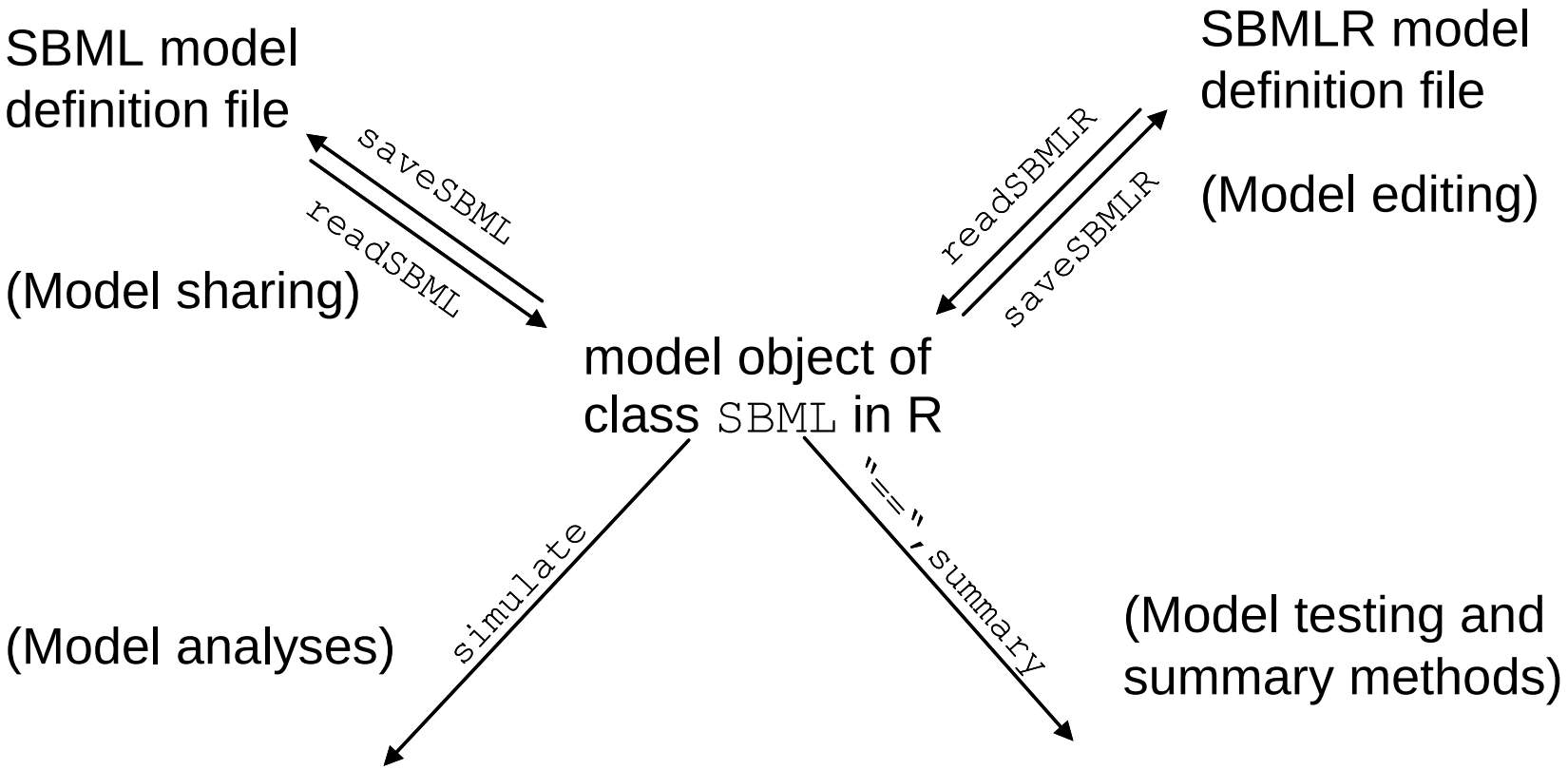
Tom Radivoyevitch

March 30, 2012

Introduction

SBMLR reads SBML files to and from an SBML-like R list of lists core object of class SBML, and it reads and writes these core objects into R text files that are well structured and light weight for editing. It also facilitates model simulations and model summaries.

Current version of SBMLR



D. Case study

Mathematical models of purine metabolism in man

Raul Curto [a], Eberhard O. Voit [b], Albert Sorribas [c], Marta Cascante [a,*]

[a] *Departament de Bioquímica i Biología Molecular, Facultat de Químiques, Universitat de Barcelona, Ci Marti i Franques, 1, 08028 Barcelona, Catalunya, Spain*

[b] *Department of Biometry and Epidemiology, Medical University of South Carolina, Charleston, SC 29425-2503, USA*

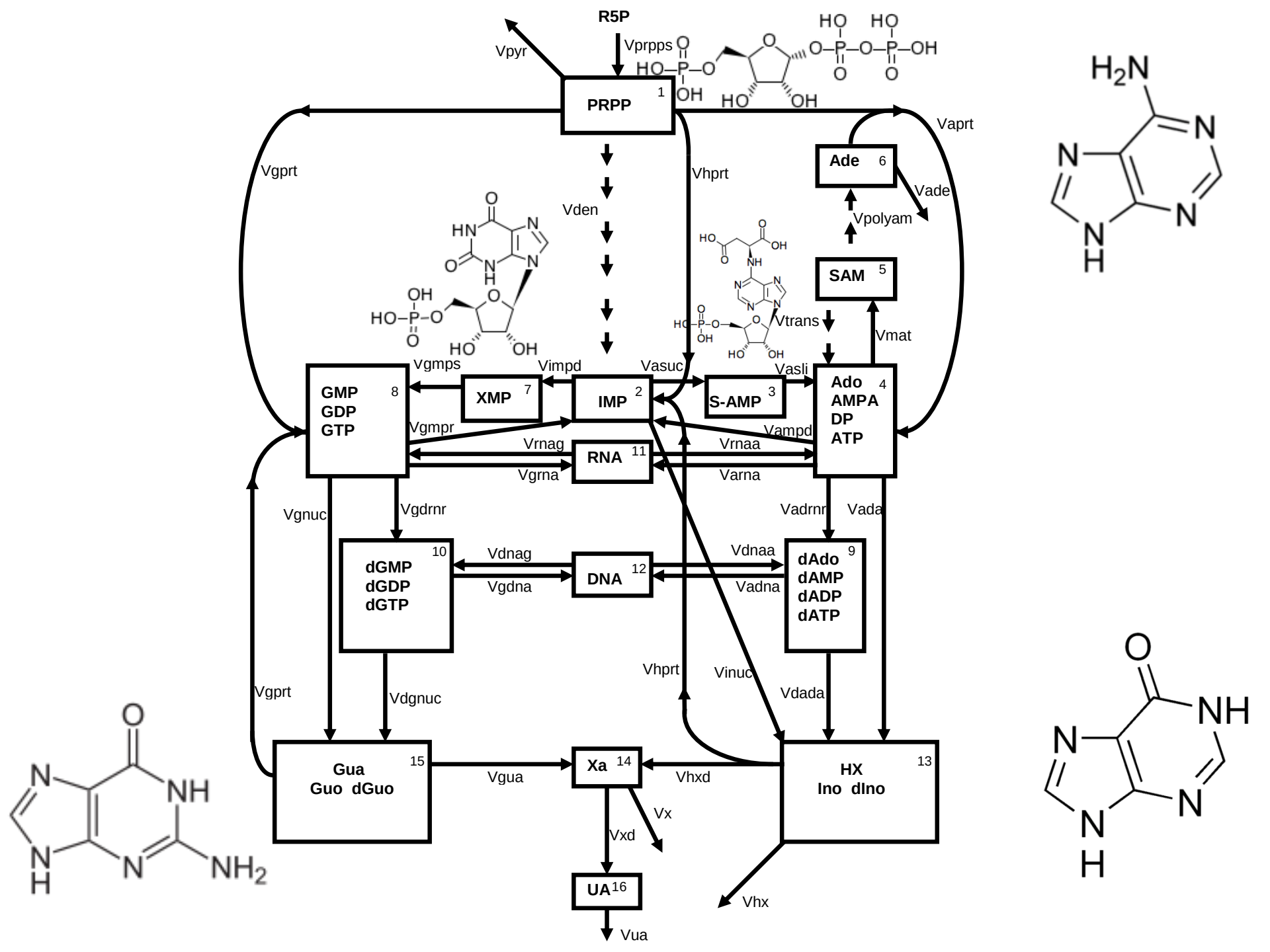
[c] *Departament de Ciències Mèdiques Bàsiques, Facultat de Medecina, Universitat de Lleida, 25198 Lleida, Catalunya, Spain*

Received 27 November 1996; received in revised form 2 February 1998

Abbreviations

Abbreviation	*Metabolite*	*Variable*
PRPP	Phosphoribosylpyrophosphate	X_1
IMP	Inosine monophosphate	X_2
S-AMP	Adenylosuccinate	X_3
Ado	Adenosine	X_4
AMP	Adenosine monophosphate	
ADP	Adenosine diphosphate	
ATP	Adenosine triphosphate	
SAM	S-adenosyl-L-methionine	X_5
Ade	Adenine	X_6
XMP	Xanthosine monophosphate	X_7
GMP	Guanosine monophosphate	X_8
GDP	Guanosine diphosphate	
GTP	Guanosine triphosphate	
dAdo	Deoxyadenosine	X_9
dAMP	Deoxyadenosine monophosphate	
dADP	Deoxyadenosine diphosphate	
dATP	Deoxyadenosine triphosphate	

dGMP	Deoxyguanosine monophosphate	X_{10}
dGDP	Deoxyguanosine diphosphate	
dGTP	Deoxyguanosine triphosphate	
RNA	Ribonucleic acid	X_{11}
DNA	Deoxyribonucleic acid	X_{12}
HX	Hypoxanthine	X_{13}
Ino	Inosine	
dIno	Deoxyinosine	
Xa	Xanthine	X_{14}
Gua	Guanine	X_{15}
Guo	Guanosine	
dGuo	Deoxyguanosine	
UA	Uric acid	X_{16}
R5P	Ribose-5-phosphate	X_{17}
Pi	Phosphate	X_{18}


R5P
Vpyr
Vprpps
PRPP 1
Vaprt
Vgprt
Vhprt
Ade 6
Vade
Vpolyam
Vden
SAM 5
Vtrans
Vmat
Vgmps
Vimpd
Vasuc
Vasli
GMP
GDP
GTP 8
XMP 7
IMP 2
S-AMP 3
Ado
AMPA
DP
ATP 4
Vgmpr
Vrnag
Vgrna
RNA 11
Vrnaa
Varna
Vampd
Vgnuc
Vgdrnr
Vadrnr
Vada
dGMP
dGDP
dGTP 10
Vdnag
Vgdna
DNA 12
Vdnaa
Vadna
dAdo
dAMP
dADP
dATP 9
Vhprt
Vinuc
Vgprt
Vdgnuc
Vdada
Gua
Guo dGuo 15
Vgua
Xa 14
Vhxd
HX
Ino dIno 13
Vx
Vxd
UA 16
Vhx
Vua

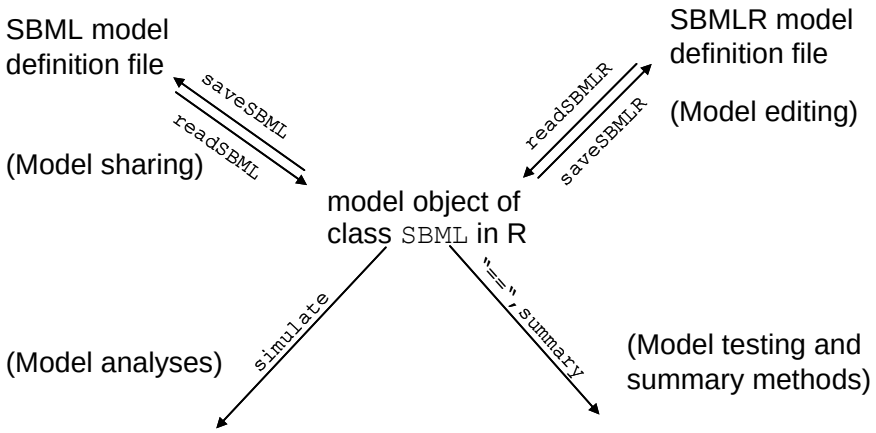
Demo 3

R3a-curtoNatural.r

Curto.xml:

It cannot be run directly by R, but SBMLR can transfer …

curto=readSBML("curto.xml")
curtoR=readSBMLR("curto.r")
curto==curtoR



(R3b-**run**Curto.r)