Package 'SBMLR'

June 19, 2025

Title SBML-R Interface and Analysis Tools

Version 2.5.0

Date 2022-03-28

Author Tomas Radivoyevitch, Vishak Venkateswaran

Description This package contains a systems biology markup language (SBML) interface to R.

Maintainer Tomas Radivoyevitch <radivot@gmail.com>

Depends XML, deSolve

Suggests rsbml

License GPL-2

URL http://epbi-radivot.cwru.edu/SBMLR/SBMLR.html

biocViews GraphAndNetwork, Pathways, Network

git_url https://git.bioconductor.org/packages/SBMLR

git_branch devel

git_last_commit 30466b5

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-06-18

Contents

`Ops.SBMLR` *Check the equality of the species and reactions of two SBMLR models*

Description

This function tests the equivalence of two models with respect to the species and reaction data frames generated by summary.

Usage

```
## S3 method for class 'SBMLR'
Ops(e1,e2)
```

Arguments

`e1`	The first of the two model objects of class SBML which are to be compared.
`e2`	The second model object.

Value

A list containing the following two boolean dataframes

`species`	The equality of species information tabularized as a data frame.
`reactions`	The equality of reaction information tabularized as a dataframe.

Author(s)

Tom Radivoyevitch

See Also

`summary.SBMLR`

Examples

```
library(SBMLR)
curto1=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
curto2=readSBML(file.path(system.file(package="SBMLR"), "models/curto.xml"))
curto1==curto2
```

readSBML *Convert an SBML file into an R object of class SBMLR*

Description

This function converts an SBML level 2 file into a corresponding R model structure of class SBMLR.

Usage

```
readSBML(filename)
```

Arguments

filename	An SBML level 2 model input file.

Details

A limited subset of SBML level 2 models is currently supported, e.g. events and function definitions are not covered.

Value

A corresponding SBMLR model object in R.

Author(s)

Tom Radivoyevitch

See Also

readSBMLR

Examples

```
library(SBMLR)
curtoX=readSBML(file.path(system.file(package="SBMLR"), "models/curto.xml"))
curtoR=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
curtoX==curtoR
```

readSBMLR *Convert an SBMLR file into an R model object of class SBMLR*

Description

This function converts an SBMLR model file into a corresponding SBMLR model object. This is more than a source-ing: the file is simpler than the object since things are generated, such as, rate law and rule R expressions and functions, and mathML.

Usage

```
readSBMLR(filename)
```

Arguments

filename	An SBMLR model definition file.

Details

A limited subset of SBML level 2 models is currently supported, e.g. events and function definitions are not covered.

Value

A corresponding SBMLR model object.

Note

This function replaces the use of source in older versions of SBMLR. It converts rate law and rule strings to R functions and expressions and to MathML.

Author(s)

Tom Radivoyevitch

See Also

readSBML

Examples

```
library(SBMLR)
curtoX=readSBML(file.path(system.file(package="SBMLR"), "models/curto.xml"))
curtoR=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
curtoX==curtoR
```

S4toS3 *Converts an S4 class SBML object created by rsbml into an S3 object of class SBMLR*

Description

This function provides a path from rsbml to SBMLR. The latter, being S3, is less cluttered with empty fields/slots than the former. The advantage of the S4 object is that it comes from more robust SBML reading: rsbml uses libsbml to parse SBML, SBMLR uses the R package XML. NOTE: As rsbml is no longer supported on the MAC, this function no longer works on the Mac.

Usage

```
S4toS3(dom)
```

Arguments

dom	An S4 DOM object of class SBML produced by rsbml.

Details

Carried over are compartments, species, global parameters, rules and reactions.

Value

A corresponding SBMLR model object, i.e. an S3 list of lists type of object.

Author(s)

Tom Radivoyevitch

Examples

```
## Not run:
library(rsbml)
(dom <- rsbml_read(file.path(system.file(package="SBMLR"), "models/sod.xml")))
library(SBMLR)
(mod=S4toS3(dom))
summary(mod)

## End(Not run)
```

---

saveSBML *Saves an SBMLR object to an SBML file.*

Description

This function converts a class SBMLR model object into an SBML level 2 version 1 file.

Usage

```
saveSBML(model,filename)
```

Arguments

model	The S3 SBMLR model object.
filename	The name of the SBML file

Details

The output file is SBML level 2.

Value

No value returned.

Warning

SBML events and function definitions are NOT implemented.

Note

The SBML file is written incrementally, rather than first built as a DOM in R and then saved using `xmlSave`.

Author(s)

Tom Radivoyevitch

References

Radivoyevitch, T. A two-way interface between limited Systems Biology Markup Language and R. BMC Bioinformatics 5, 190 (2004).

See Also

`saveSBMLR`

Examples

```
library(SBMLR)
curtoR=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
saveSBML(curtoR,"curtoR.xml")
curtoX=readSBML("curtoR.xml")
curtoX==curtoR
summary(curtoR)
unlink("curtoR.xml")
```

`saveSBMLR` *Save an R model object of class SBMLR to a file.*

Description

This function converts an `SBMLR` model object in R into an SBMLR model definition file. Rate laws are provided only in string form. Redundancy is eliminated to make the file easier to edit.

Usage

```
saveSBMLR(model,filename)
```

Arguments

`model`	The `SBMLR` model object to be mapped into the SBMLR model definition file.
`filename`	The file name of the destination SBMLR model definition file.

Value

No value returned.

Warning

SBML events and function definitions are NOT implemented.

Note

Similar to saveSBML, the file is written incrementally.

Author(s)

Tom Radivoyevitch

References

Radivoyevitch, T. A two-way interface between limited Systems Biology Markup Language and R. BMC Bioinformatics 5, 190 (2004).

See Also

saveSBML

Examples

```
library(SBMLR)
curto=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
saveSBMLR(curto,"curtoR.r")
curtoR=readSBMLR("curtoR.r")
curto==curtoR
summary(curtoR)
unlink("curtoR.r")
```

---

sim *Simulate a model of S3 class SBMLR*

Description

This function simulates a model given report times and optional modulators. It uses `lsoda` of the deSolve package.

Usage

```
sim(model, times, modulator=NULL,X0=NULL, ...)
```

Arguments

`model`	The S3 model object to be simulated. Initial conditions are passed through this object.
`times`	The sequence of time points to be sampled and provided as rows of the output matrix.
`modulator`	Null if there are no modulators (default), a vector of numbers if there are steady state Vmax modulators, and a list of interpolating functions if there are time course Vmax modulators.
`X0`	Override model initial conditions in simulations, particularly piece-wise perturbation simulations.
`...`	To pass extra args such as event data frames to deSolve.

Details

This is a wrapper for ode.

Value

The data frame output that comes out of ode.

Note

Rules are implemented through time varying boundary conditions updated at each time point as a side effect within the (now internal) function `fderiv`.

Author(s)

Tom Radivoyevitch

References

For the folate cycle example given below: Morrison PF, Allegra CJ: Folate cycle kinetics in human breast cancer cells. JBiolChem 1989, 264(18):10552-10566.

Examples

```
##---- The following perturbs PRPP from 5 to 50 uM in Curto et al.'s model.
library(SBMLR)
curto=readSBML(file.path(system.file(package="SBMLR"), "models/curto.xml"))
(dPRPP10 <- data.frame(var = "PRPP", time = 0, value = 10,method = "mult"))
(out=sim(curto,times=seq(-20,70,1),events = list(data = dPRPP10) ) )
plot(out,which=c("PRPP","den","IMP","HX","Gua","aprt","XMP","Xa","UA"))

# which should be the same plots as
curto=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
(dPRPP10 <- data.frame(var = "PRPP", time = 0, value = 10,method = "mult"))
(out=sim(curto,times=seq(-20,70,1),events = list(data = dPRPP10) ) )
plot(out,which=c("PRPP","den","IMP","HX","Gua","aprt","XMP","Xa","UA"))


##---- The following generates Morrison's folate system response to 1uM MTX
morr=readSBMLR(file.path(system.file(package="SBMLR"), "models/morrison.r"))
out1=sim(morr,seq(-20,0,1))
morr$species$EMTX$ic=1
out2=sim(morr,0:30)
outs=data.frame(rbind(out1,out2))
attach(outs)
par(mfrow=c(3,4))
plot(time,FH2b,type="l",xlab="Hours")
plot(time,FH2f,type="l",xlab="Hours")
plot(time,DHFRf,type="l",xlab="Hours")
plot(time,DHFRtot,type="l",xlab="Hours")
plot(time,CHOFH4,type="l",xlab="Hours")
plot(time,FH4,type="l",xlab="Hours")
plot(time,CH2FH4,type="l",xlab="Hours")
plot(time,CH3FH4,type="l",xlab="Hours")
plot(time,AICARsyn,type="l",xlab="Hours")
plot(time,MTR,type="l",xlab="Hours")
plot(time,TYMS,type="l",xlab="Hours")
```

```
#plot(time,EMTX,type="l",xlab="Hours")
plot(time,DHFReductase,type="l",xlab="Hours")
par(mfrow=c(1,1))
detach(outs)
morr$species$EMTX$ic=0

## Note: This does not work, since EMTX is not a state variable, it is a bc
##(dEMTX1 <- data.frame(var = "EMTX", time = 0, value = 1,method = "add"))
##(out=simulate(morr,times=seq(-20,30,1),events = list(data = dEMTX1) ) )
```

`summary.SBMLR` *Get summary information from an SBMLR model*

Description

This function extracts information from a model of class `SBMLR` and returns it as a list. The list includes species and reaction information tabularized as data frames.

Usage

```
## S3 method for class 'SBMLR'
summary(object,...)
```

Arguments

`object`	A model object of class `SBMLR` from which information is to be extracted.
`...`	For compatibility with `summary` of the base package.

Details

no details

Value

A list containing the following elements

`BC`	A logical vector indicating which species are not state variables, i.e. which species are boundary conditions or auxillary variables.
`y0`	The initial state (boundary conditions excluded!).
`nStates`	The length of the state vector, i.e. the number of system states.
`S0`	The full set of species initial values.
`nReactions`	The number of reactions.
`nSpecies`	The number of species, including states, boundary conditions and possibly auxillary variables such as the total concentration of dihydofolate reductase in the morrison.r model.
`incid`	The incidence/stoichiometry matrix. This usually contains ones and minus ones corresponding to fluxes either synthesizing or degrading (respectively) a state variable chemical species. This matrix multiplied by the flux vector on its right yields the corresponding concentration state variable time derivatives.

species	Species information (i.e. names, ICs, BCs, and compartments) as a data frame.
reactions	Reaction information tabularized as a dataframe, including string laws and initial fluxes.

Note

The list output can be attached to immediately define many model variables of interest.

Author(s)

Tom Radivoyevitch

Examples

```
library(SBMLR)
curto=readSBMLR(file.path(system.file(package="SBMLR"), "models/curto.r"))
summary(curto)
```

Index