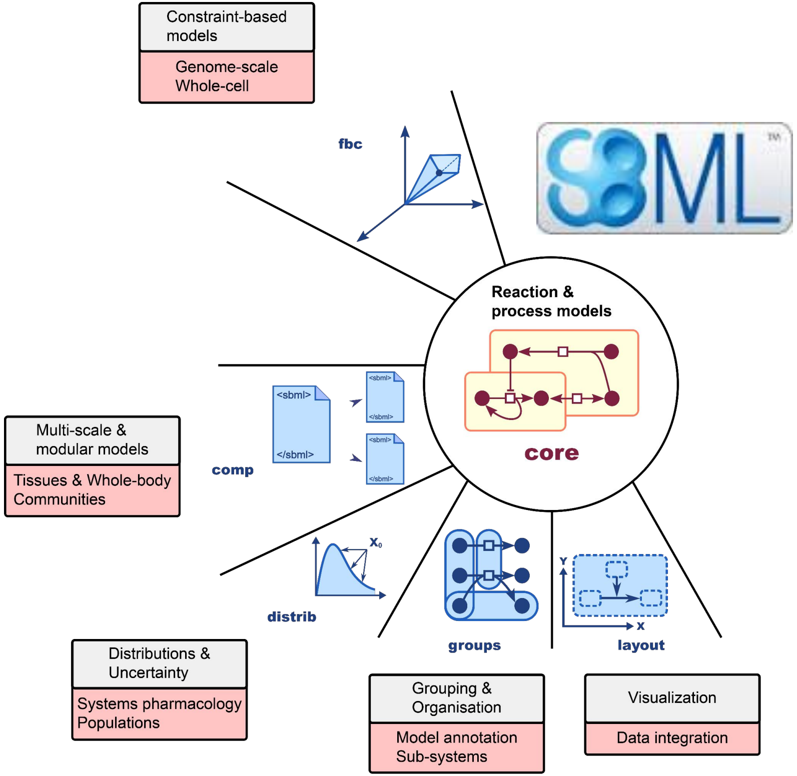
sbmlutils:
Python utilities for SBML

Matthias König
@konigmatt
https://livermetabolism.com

What is SBML?

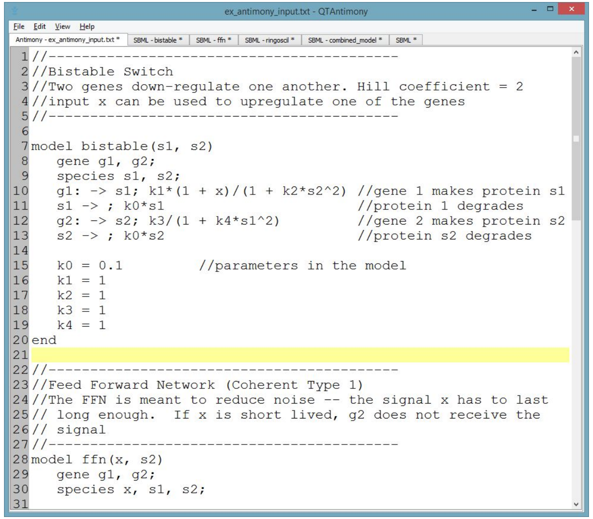
- Systems Biology Markup Language
  - core language
  - packages for additional functionality
- Standard exchange format for computational models in biology
  - > 300 tools support
  - large ecosystem: model building, visualization, simulation, …
- http://sbml.org



SBML Level 3: an extensible format for the exchange and reuse of biological models.
*Keating SM, Waltemath D, **König M**, Zhang F, … Hucka M; SBML Level 3 Community members.*
Mol Syst Biol. 2020 Aug;16(8):e9110. doi: 10.15252/msb.20199110. PMID: 32845085.

The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 2 Core Release 2.
*Hucka M, Bergmann FT, Chaouiya C, Dräger A, Hoops S, Keating SM, **König M**, Novère NL, Myers CJ, Olivier BG, Sahle S, Schaff JC, Sheriff R, Smith LP, Waltemath D, Wilkinson DJ, Zhang F.*
J Integr Bioinform. 2019 Jun 20;16(2):20190021. doi: 10.1515/jib-2019-0021. PMID: 31219795; PMCID: PMC6798823.

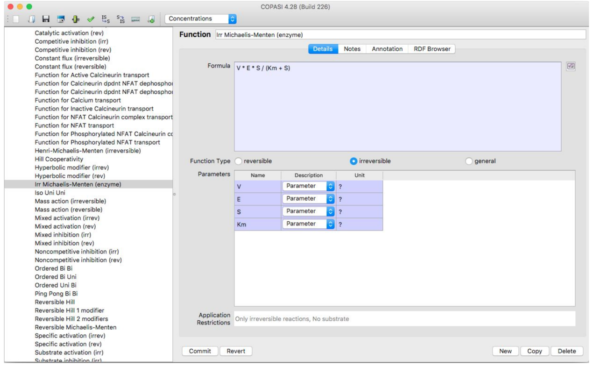
How to create an SBML model?



```
//-------------------------------------------
//Bistable Switch
//Two genes down-regulate one another. Hill coefficient = 2
//input x can be used to upregulate one of the genes
//-------------------------------------------

model bistable(s1, s2)
   gene g1, g2;
   species s1, s2;
   g1: -> s1; k1*(1 + x)/(1 + k2*s2^2) //gene 1 makes protein s1
   s1 -> ; k0*s1                       //protein 1 degrades
   g2: -> s2; k3/(1 + k4*s1^2)         //gene 2 makes protein s2
   s2 -> ; k0*s2                       //protein s2 degrades

   k0 = 0.1         //parameters in the model
   k1 = 1
   k2 = 1
   k3 = 1
   k4 = 1
end

//-------------------------------------------
//Feed Forward Network (Coherent Type 1)
//The FFN is meant to reduce noise -- the signal x has to last
// long enough.  If x is short lived, g2 does not receive the
// signal
//-------------------------------------------
model ffn(x, s2)
   gene g1, g2;
   species x, s1, s2;
```

Antimony

- **text-based format**
- no validation
- no type hinting
- SBML core & distrib



Catalytic activation (rev)
Competitive inhibition (irr)
Competitive inhibition (rev)
Constant flux (irreversible)
Constant flux (reversible)
Function for Active Calcineurin transport
Function for Calcineurin dpdnt NFAT dephosphor
Function for Calcineurin dpdnt NFAT dephosphor
Function for Calcium transport
Function for Inactive Calcineurin transport
Function for NFAT Calcineurin complex transport
Function for NFAT transport
Function for Phosphorylated NFAT Calcineurin co
Function for Phosphorylated NFAT transport
Henri-Michaelis-Menten (irreversible)
Hill Cooperativity
Hyperbolic modifier (irrev)
Hyperbolic modifier (rev)
Irr Michaelis-Menten (enzyme)
Iso Uni Uni
Mass action (irreversible)
Mass action (reversible)
Mixed activation (irrev)
Mixed activation (rev)
Mixed inhibition (irr)
Mixed inhibition (rev)
Noncompetitive inhibition (irr)
Noncompetitive inhibition (rev)
Ordered Bi Bi
Ordered Bi Uni
Ordered Uni Bi
Ping Pong Bi Bi
Reversible Hill
Reversible Hill 1 modifier
Reversible Hill 2 modifiers
Reversible Michaelis-Menten
Specific activation (irrev)
Specific activation (rev)
Substrate activation (irr)
Substrate inhibition (irr)

Function: Irr Michaelis-Menten (enzyme)

Formula: V * E * S / (Km + S)

Function Type: ○ reversible ● irreversible ○ general

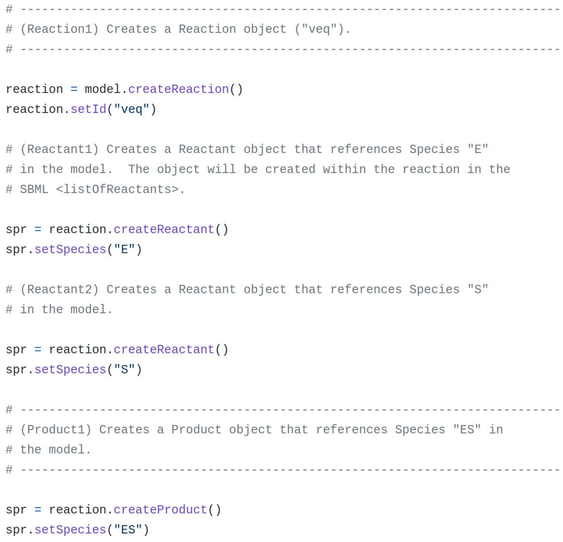
Parameters

Name	Description	Unit
V	Parameter	?
E	Parameter	?
S	Parameter	?
Km	Parameter	?

Application Restrictions: Only irreversible reactions, No substrate

COPASI (CellDesigner)

- **GUI-based**
- no/difficult programmatic interaction



```
# ---------------------------------------------------------------------------
# (Reaction1) Creates a Reaction object ("veq").
# ---------------------------------------------------------------------------

reaction = model.createReaction()
reaction.setId("veq")

# (Reactant1) Creates a Reactant object that references Species "E"
# in the model.  The object will be created within the reaction in the
# SBML <listOfReactants>.

spr = reaction.createReactant()
spr.setSpecies("E")

# (Reactant2) Creates a Reactant object that references Species "S"
# in the model.

spr = reaction.createReactant()
spr.setSpecies("S")

# ---------------------------------------------------------------------------
# (Product1) Creates a Product object that references Species "ES" in
# the model.
# ---------------------------------------------------------------------------

spr = reaction.createProduct()
spr.setSpecies("ES")
```

libSBML (JSBML)

- **very low level**
- many silent failures (return code handling)

What is sbmlutils?

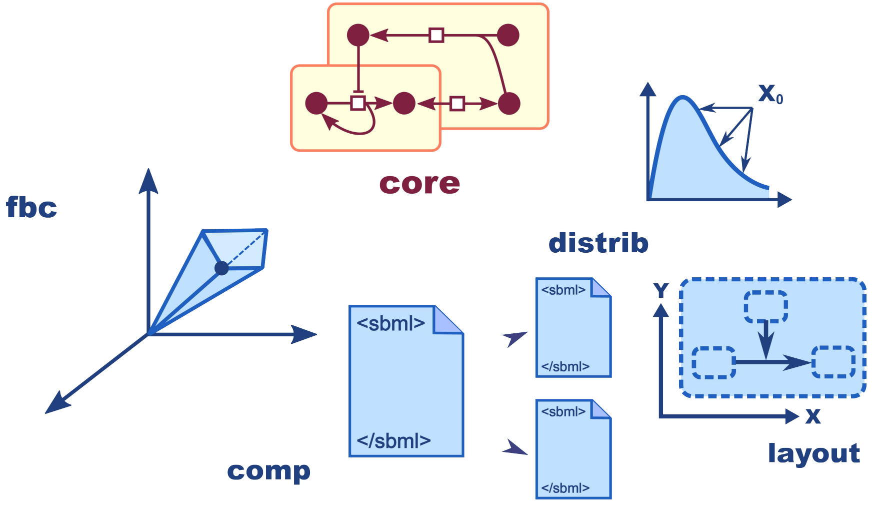
Collection of python utilities for
programmatically developing SBML models
https://github.com/matthiaskoenig/sbmlutils/
https://sbmlutils.readthedocs.io

Features

- model creation, manipulation & merging
- unit support
- type annotations
- annotation support
- model reports
- interpolations
- file converters (XPP)

Packages

- core, fbc, comp, distrib, layout

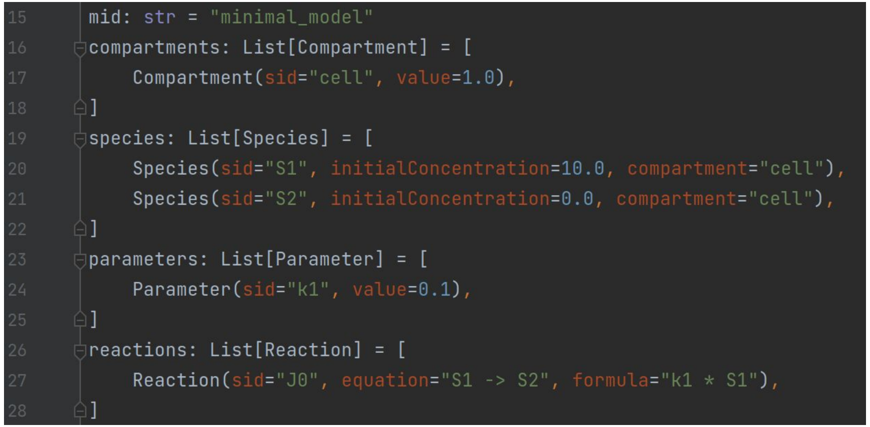


CI-CD passing | pypi v0.4.12 | python 3.7 | 3.8 | 3.9 | license LGPL-3.0 | codecov 77% | docs passing

DOI 10.5281/zenodo.4595467 | code style black | mypy checked

Model creation with sbmlutils

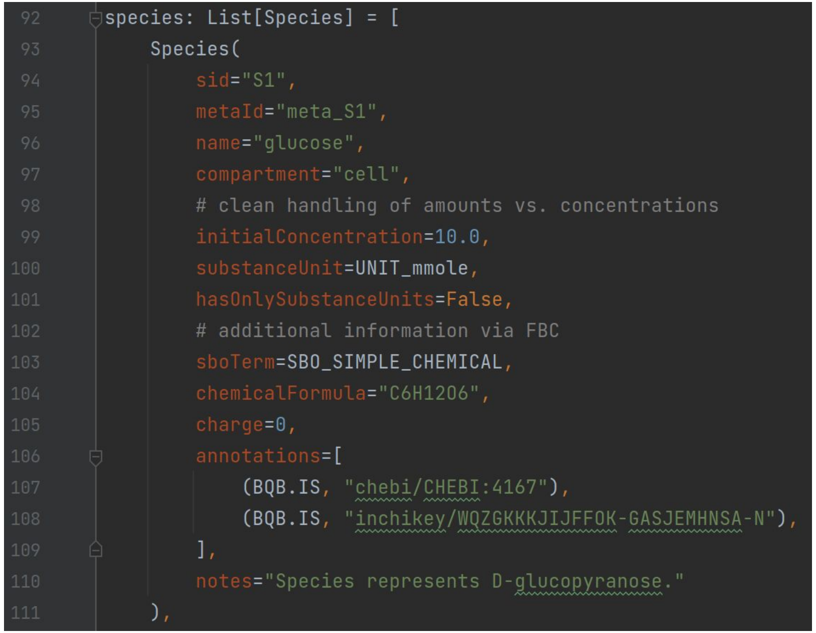
- layer around libsbml
- programmatic generation (Python)
- type annotations, type hinting, auto-completion, documentation
- validation with warnings/errors
- model reports



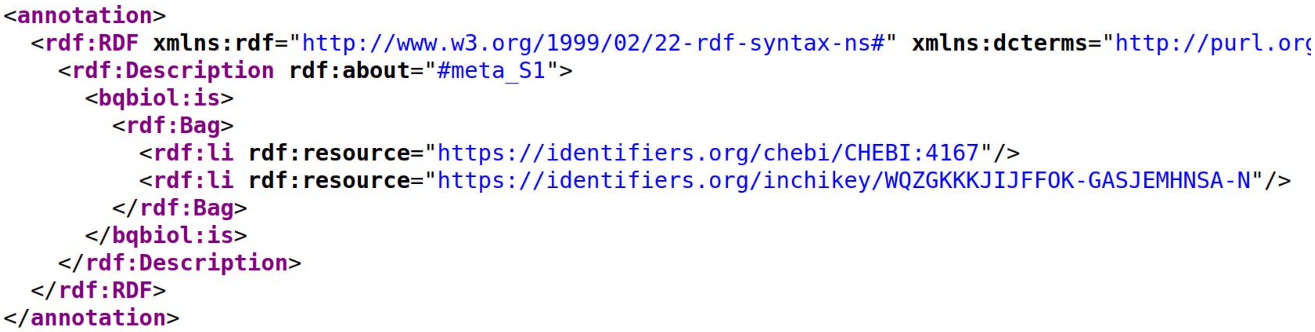
```
mid: str = "minimal_model"
compartments: List[Compartment] = [
    Compartment(sid="cell", value=1.0),
]
species: List[Species] = [
    Species(sid="S1", initialConcentration=10.0, compartment="cell"),
    Species(sid="S2", initialConcentration=0.0, compartment="cell"),
]
parameters: List[Parameter] = [
    Parameter(sid="k1", value=0.1),
]
reactions: List[Reaction] = [
    Reaction(sid="J0", equation="S1 -> S2", formula="k1 * S1"),
]
```

Model annotations

- object level annotations
  - every SBase can be annotation
  - restricted set of MIRIAM qualifiers
- pattern based annotations
  - match objects based on regular expressions
- validation against identifiers.org resources
  - regular expressions patterns



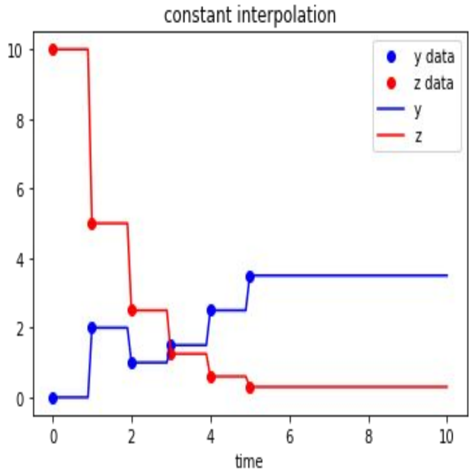
```python
species: List[Species] = [
    Species(
        sid="S1",
        metaId="meta_S1",
        name="glucose",
        compartment="cell",
        # clean handling of amounts vs. concentrations
        initialConcentration=10.0,
        substanceUnit=UNIT_mmole,
        hasOnlySubstanceUnits=False,
        # additional information via FBC
        sboTerm=SBO_SIMPLE_CHEMICAL,
        chemicalFormula="C6H12O6",
        charge=0,
        annotations=[
            (BQB.IS, "chebi/CHEBI:4167"),
            (BQB.IS, "inchikey/WQZGKKKJIJFFOK-GASJEMHNSA-N"),
        ],
        notes="Species represents D-glucopyranose."
    ),
```

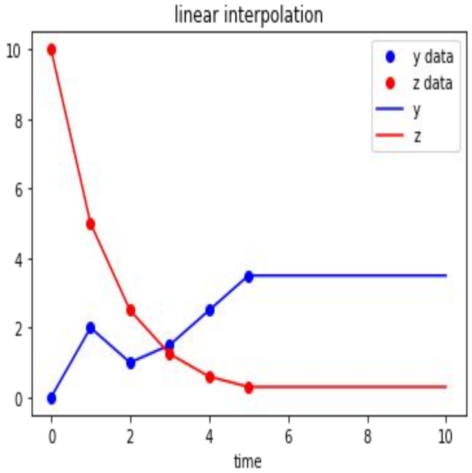


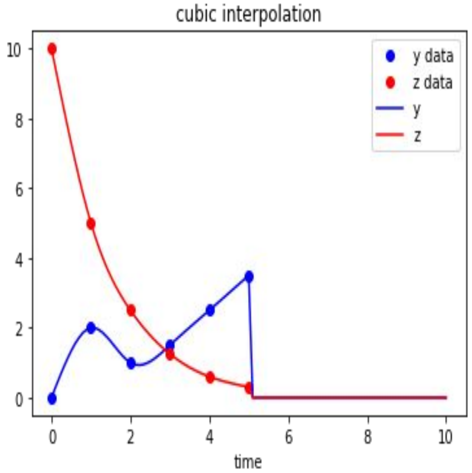
```xml
<annotation>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dcterms="http://purl.org
    <rdf:Description rdf:about="#meta_S1">
      <bqbiol:is>
        <rdf:Bag>
          <rdf:li rdf:resource="https://identifiers.org/chebi/CHEBI:4167"/>
          <rdf:li rdf:resource="https://identifiers.org/inchikey/WQZGKKKJIJFFOK-GASJEMHNSA-N"/>
        </rdf:Bag>
      </bqbiol:is>
    </rdf:Description>
  </rdf:RDF>
</annotation>
```

Data interpolation

- use data to add spline functions to models
- driving model by input; clamping

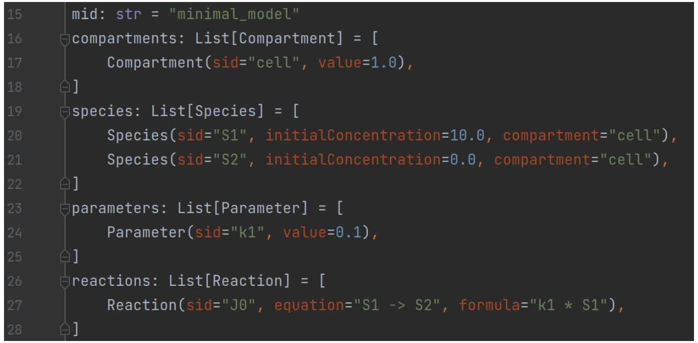

constant interpolation


linear interpolation


cubic interpolation

sbmlutils

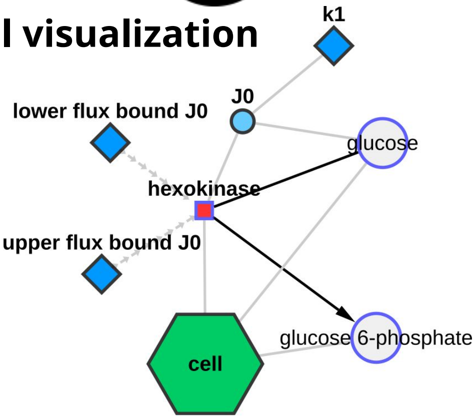
Model development



```
mid: str = "minimal_model"
compartments: List[Compartment] = [
    Compartment(sid="cell", value=1.0),
]
species: List[Species] = [
    Species(sid="S1", initialConcentration=10.0, compartment="cell"),
    Species(sid="S2", initialConcentration=0.0, compartment="cell"),
]
parameters: List[Parameter] = [
    Parameter(sid="k1", value=0.1),
]
reactions: List[Reaction] = [
    Reaction(sid="J0", equation="S1 -> S2", formula="k1 * S1"),
]
```

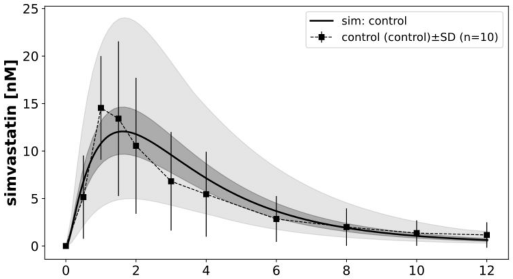
cysbml

Model visualization



sbmlsim

Model simulation & analysis

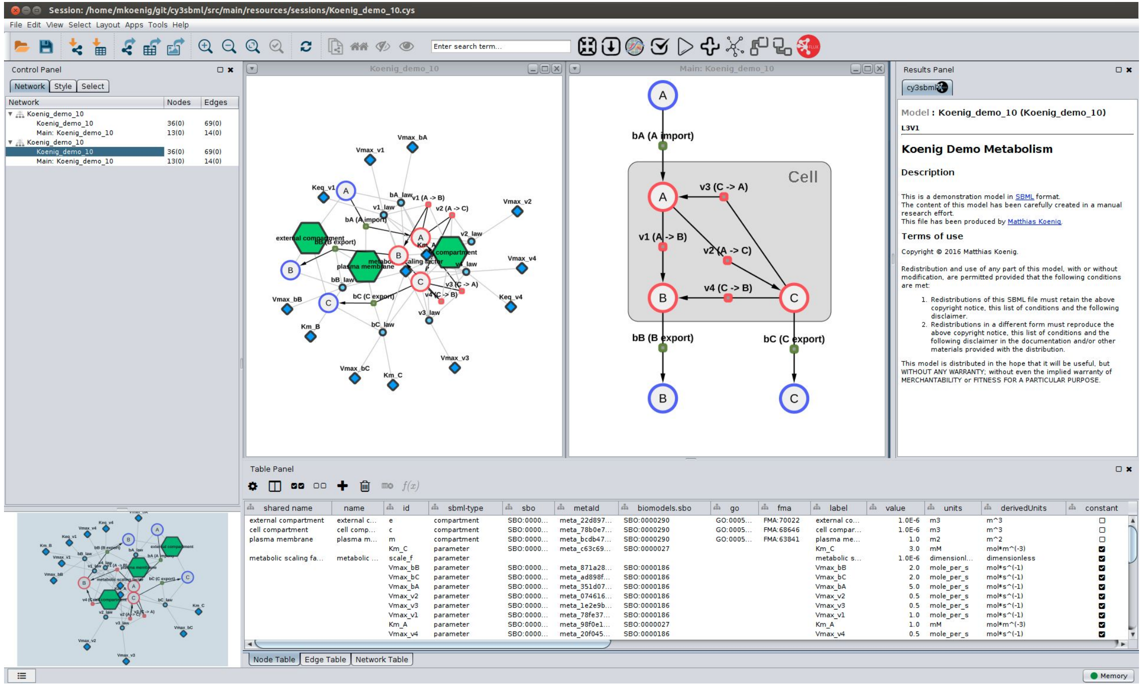


What is cysbml?

Cytoscape app for <u>visualizing</u> SBML models (ODE)
https://github.com/matthiaskoenig/cysbml

Features

- kinetic & reaction-species view
- subgraphs & filtering
- annotation support
- works for large scale networks (genome-scale)



Session: /home/mkoenig/git/cy3sbml/src/main/resources/sessions/Koenig_demo_10.cys

Network	Nodes	Edges
Koenig_demo_10		
Koenig_demo_10	36(0)	69(0)
Main: Koenig_demo_10	13(0)	14(0)
Koenig_demo_10		
Koenig_demo_10	36(0)	69(0)
Main: Koenig_demo_10	13(0)	14(0)

Model : **Koenig_demo_10 (Koenig_demo_10)**
L3V1

Koenig Demo Metabolism

Description

This is a demonstration model in SBML format.
The content of this model has been carefully created in a manual research effort.
This file has been produced by Matthias Koenig.

Terms of use

Copyright © 2016 Matthias Koenig.

Redistribution and use of any part of this model, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of this SBML file must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in a different form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This model is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

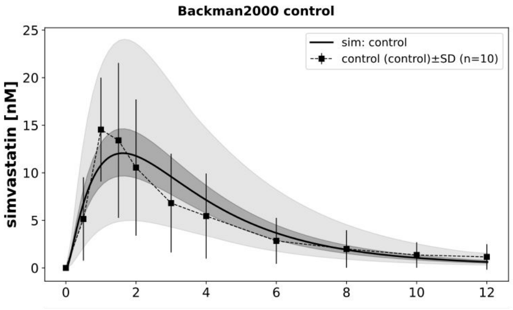
shared name	name	id	sbml-type	sbo	metaId	biomodels.sbo	go	fma	label	value	units	derivedUnits	constant
external compartment	external c...	e	compartment	SBO:0000...	meta_22d897...	SBO:0000290	GO:0005...	FMA:70022	external co...	1.0E-6	m3	m^3	☐
cell compartment	cell comp...	c	compartment	SBO:0000...	meta_78b0e7...	SBO:0000290	GO:0005...	FMA:68646	cell compar...	1.0E-6	m3	m^3	☐
plasma membrane	plasma m...	m	compartment	SBO:0000...	meta_bcdb47...	SBO:0000290	GO:0005...	FMA:63841	plasma me...	1.0	m2	m^2	☐
		Km_C	parameter	SBO:0000...	meta_c63c69...	SBO:0000027			Km_C	3.0	mM	mol*m^(-3)	☑
metabolic scaling fa...	metabolic ...	scale_f	parameter						metabolic s...	1.0E-6	dimensionl...	dimensionless	☑
		Vmax_bB	parameter	SBO:0000...	meta_871a28...	SBO:0000186			Vmax_bB	2.0	mole_per_s	mol*s^(-1)	☑
		Vmax_bC	parameter	SBO:0000...	meta_ad898f...	SBO:0000186			Vmax_bC	2.0	mole_per_s	mol*s^(-1)	☑
		Vmax_bA	parameter	SBO:0000...	meta_351d07...	SBO:0000186			Vmax_bA	5.0	mole_per_s	mol*s^(-1)	☑
		Vmax_v2	parameter	SBO:0000...	meta_074616...	SBO:0000186			Vmax_v2	0.5	mole_per_s	mol*s^(-1)	☑
		Vmax_v3	parameter	SBO:0000...	meta_1e2e9b...	SBO:0000186			Vmax_v3	0.5	mole_per_s	mol*s^(-1)	☑
		Vmax_v1	parameter	SBO:0000...	meta_78fe37...	SBO:0000186			Vmax_v1	1.0	mole_per_s	mol*s^(-1)	☑
		Km_A	parameter	SBO:0000...	meta_98f0e1...	SBO:0000027			Km_A	1.0	mM	mol*m^(-3)	☑
		Vmax_v4	parameter	SBO:0000...	meta_20f045...	SBO:0000186			Vmax_v4	0.5	mole_per_s	mol*s^(-1)	☑

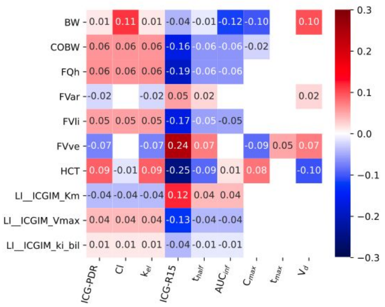
What is sbmlsim?

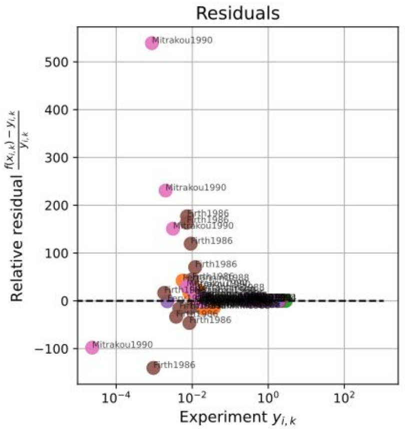
Collection of python utilities for simulating SBML models (ODE)
https://github.com/matthiaskoenig/sbmlsim
https://sbmlsim.readthedocs.io

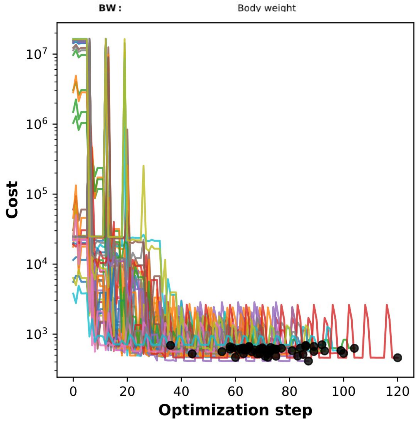
Features

- model simulation
- data support
- unit support (& conversion)
- multi-core execution (ray)
- simulation experiments (publications)
- parameter scans
- parameter fitting









CI-CD passing | pypi v0.2.0 | python 3.7 | 3.8 | license LGPL-3.0 | codecov 49% | docs failing

DOI 10.5281/zenodo.3597770 | code style black

Acknowledgements

- MK was supported by the Federal Ministry of Education and Research (**BMBF**, Germany) within the research network **Systems Medicine of the Liver** (LiSyM, grant number 031L0054).
- MK was supported by the German Research Foundation (**DFG**) within the Research Unit Programme FOR 5151 "**QuaLiPerF** (Quantifying Liver Perfusion-Function Relationship in Complex Resection - A Systems Medicine Approach)" by grant number 436883643.
- The sbmlutils report was supported by **Google Summer of Code** 2021.
- We acknowledge support by the BMBF-funded **de.NBI Cloud** within the German Network for Bioinformatics Infrastructure (de.NBI) (031A537B, 031A533A, 031A538A, 031A533B, 031A535A, 031A537C, 031A534A, 031A532B) for the PK-DB infrastructure.

DFG

Federal Ministry
of Education
and Research

Google
Summer of Code

EOSC-Life