libSBML

Sarah Keating
Using MATLAB/Octave ...
Installing libSBML MATLAB binding
Installing libSBML MATLAB binding on Windows
Installing libSBML Octave binding on Windows

- Install libSBML using the windows installer.
- Open Octave and navigate to the binding/octave directory
- Run buildLibSBMLOctave
Installing libSBML MATLAB binding on Linux/Mac

./configure --with-matlab
make
sudo make install
Installing libSBML Octave binding on Linux/Mac

./configure --with-octave
make
sudo make install
Import SBML

Your preferred environment
m =

    { 'SBML_MODEL'
    , 'hh'
    , '
    , '
    , 3
    , 1
    , 'ml'
    , '13_all'
    , 'time'
    , 'mole'
    , 'litre'
    , 'area'
    , 'metre'
    , 'mole'
    , 'd'
    , -1
    , [1x1 struct]
    , [1x5 struct]
    , [1x2 struct]
    , [1x3 struct]
    , [1x6 struct]
    , [1x1 struct]
    , [1x4 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]
    , [1x1 struct]};
MATLAB_SBML structure

User conversion

SBMLToolbox
MATLAB_SBML structure

```
m =

typecode: 'SBML_MODEL'
metaid: 'hh'
notes: ''
annotation: ''
SBML_level: 3
SBML_version: 1
```
MATLAB_SBML structure

time_symbol: ''
delay_symbol: ''
namespaces: [1x1 struct]
MATLAB_SBML structure

name: 'm1'

id: '13_all'
timeUnits: 'time'
substanceUnits: 'mole'
volumeUnits: 'litre'
areaUnits: 'area'
lengthUnits: 'metre'
extentUnits: 'mole'
conversionFactor: 'd'
sboTerm: -1
MATLAB_SBML structure

```
functionDefinition: [1x1 struct]
unitDefinition: [1x5 struct]
compartment: [1x2 struct]
species: [1x3 struct]
parameter: [1x6 struct]
initialAssignment: [1x1 struct]
rule: [1x4 struct]
constraint: [1x1 struct]
reaction: [1x1 struct]
event: [1x1 struct]
```
MATLAB_SBML structure

```matlab
>> m.species(1)

ans =

    typecode: 'SBML_SPECIES'
    metaid: ''
    notes: ''
    annotation: ''
    sboTerm: -1
    name: ''
    id: 's'
    compartment: 'a'
    initialAmount: 0
    initialConcentration: NaN
    substanceUnits: 'substance'
    hasOnlySubstanceUnits: 0
    boundaryCondition: 0
    constant: 0
    conversionFactor: 'd'
    setIsInitialState: 1
    setIsInitialStateConcentration: 0
```
MATLAB_SBML structure

>> m.species(1)

ans =

    typecode: 'SBML_SPECIES'
    metaid: ''
    notes: ''
    annotation: ''
    sboTerm: -1
    name: ''
    id: 's'
    compartment: 'a'
    initialAmount: 0
    initialConcentration: NaN
    substanceUnits: 'substance'
    hasOnlySubstanceUnits: 0
    boundaryCondition: 0
    constant: 0
    conversionFactor: 'd'
    isSetInitialAmount: 1
    isSetInitialConcentration: 0
>> numParams = length(m.parameter)

numParams =

6

>> param1 = m.parameter(1).id

param1 =

p

>> compSize = m.compartment(2).size

compSize =

NaN

>> hascompSize = m.compartment(2).isSetSize

hascompSize =

0
Export SBML

Your preferred environment

Must be a structure of the required format

Use the ‘isSBML_Model’ script to determine whether the structure is as expected.
Export SBML
Checking the structure

isSBML_Model

✓ checks typecode

✓ all required fields present

✓ have appropriate types of data
Acknowledgements

Akiya Jouraku
Keio, Japan

Frank Bergmann
Caltech, USA

Mike Hucka
Caltech, USA