

Toolbox

4 MATLAB[®]

Sarah Keating

MATLAB[®] Toolbox

➤ converts

➤ stores

➤ views

SBML models in MATLAB[®] environment

Converting SBML Models into MATLAB

Link to libsbml

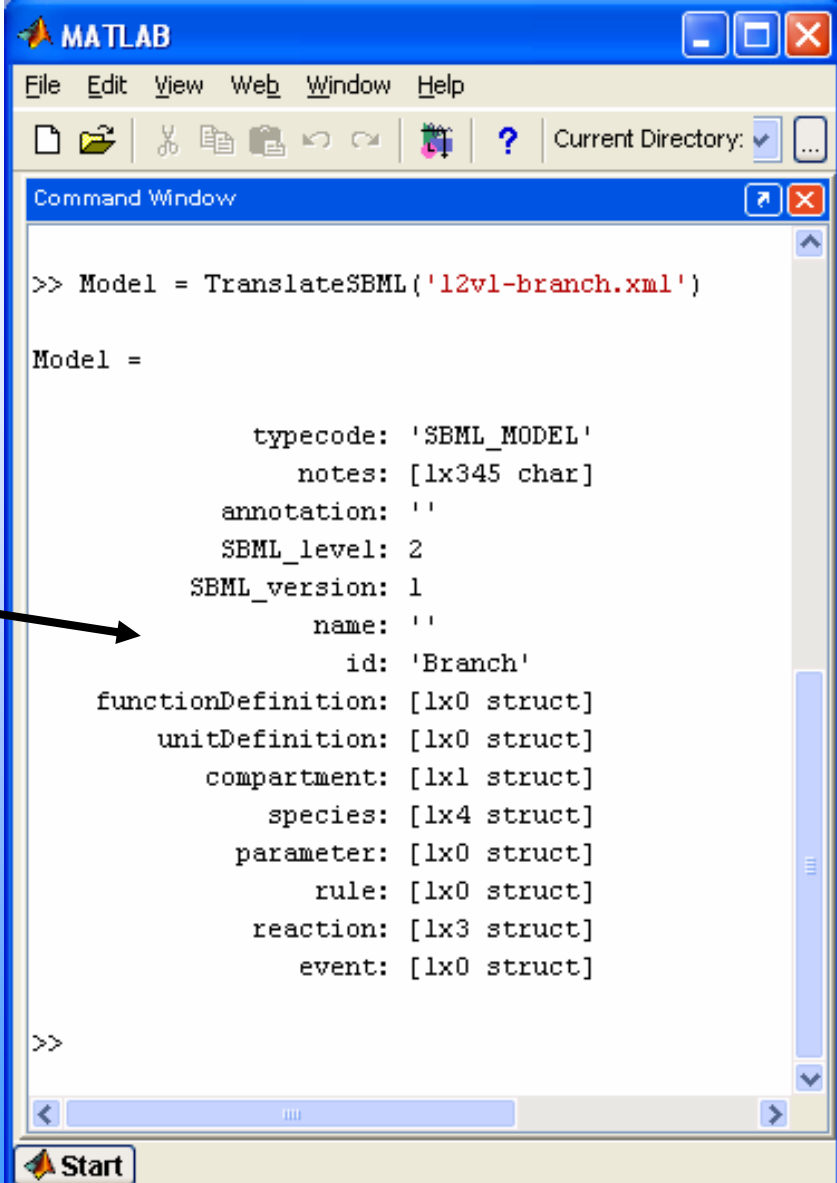
Build a MATLAB executable

TranslateSBML

(.dll in Windows)

 Model = TranslateSBML('filename')

Matlab Structure
that replicates the
C structures used
by libsbml



```
MATLAB
File Edit View Web Window Help
Current Directory: ...
Command Window
>> Model = TranslateSBML('12v1-branch.xml')

Model =

        typecode: 'SBML_MODEL'
          notes: [1x345 char]
    annotation: ''
    SBML_level: 2
    SBML_version: 1
          name: ''
          id: 'Branch'
functionDefinition: [1x0 struct]
    unitDefinition: [1x0 struct]
    compartment: [1x1 struct]
        species: [1x4 struct]
    parameter: [1x0 struct]
          rule: [1x0 struct]
    reaction: [1x3 struct]
        event: [1x0 struct]

>>
```

The image shows a MATLAB Command Window with a blue title bar. The window contains the following text: `>> Model = TranslateSBML('12v1-branch.xml')`. Below this, the output is displayed as a structure: `Model =` followed by a list of fields and their values: `typecode: 'SBML_MODEL'`, `notes: [1x345 char]`, `annotation: ''`, `SBML_level: 2`, `SBML_version: 1`, `name: ''`, `id: 'Branch'`, `functionDefinition: [1x0 struct]`, `unitDefinition: [1x0 struct]`, `compartment: [1x1 struct]`, `species: [1x4 struct]`, `parameter: [1x0 struct]`, `rule: [1x0 struct]`, `reaction: [1x3 struct]`, and `event: [1x0 struct]`. At the bottom of the window, there is a `>>` prompt. The MATLAB logo and the word "MATLAB" are in the top left corner of the window. The Windows taskbar is visible at the bottom of the image, showing the Start button.

```

MATLAB
File Edit View Web Window Help
Current Directory:
Command Window
>> Model.reaction(1)

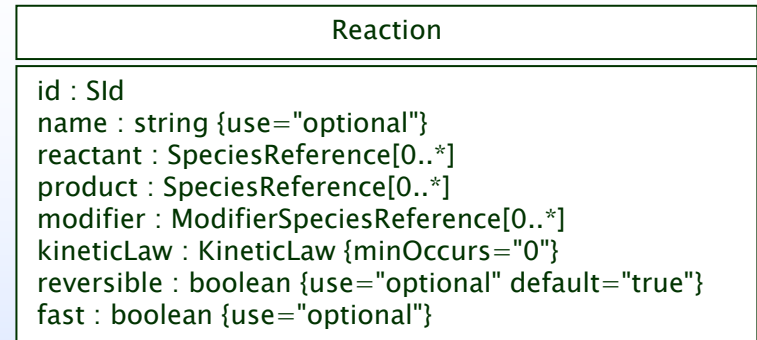
ans =

    typecode: 'SBML_REACTION'
      notes: ''
  annotation: ''
      name: ''
         id: 'reaction_1'
  reactant: [1x1 struct]
  product: [1x1 struct]
  modifier: [1x0 struct]
 kineticLaw: [1x1 struct]
  reversible: 0
           fast: 0
    IsSetFast: 0

>> |

```

MATLAB



UML

```

typedef struct
{
  SBASE_FIELDS;
  char      *id;
  char      *name;
  ListOf_t  *reactant;
  ListOf_t  *product;
  ListOf_t  *modifier;
  KineticLaw_t *kineticLaw;
  int       reversible;
  int       fast;

  struct
  {
    unsigned int fast:1;
  }
  isSet;
} Reaction_t;

```

C structure

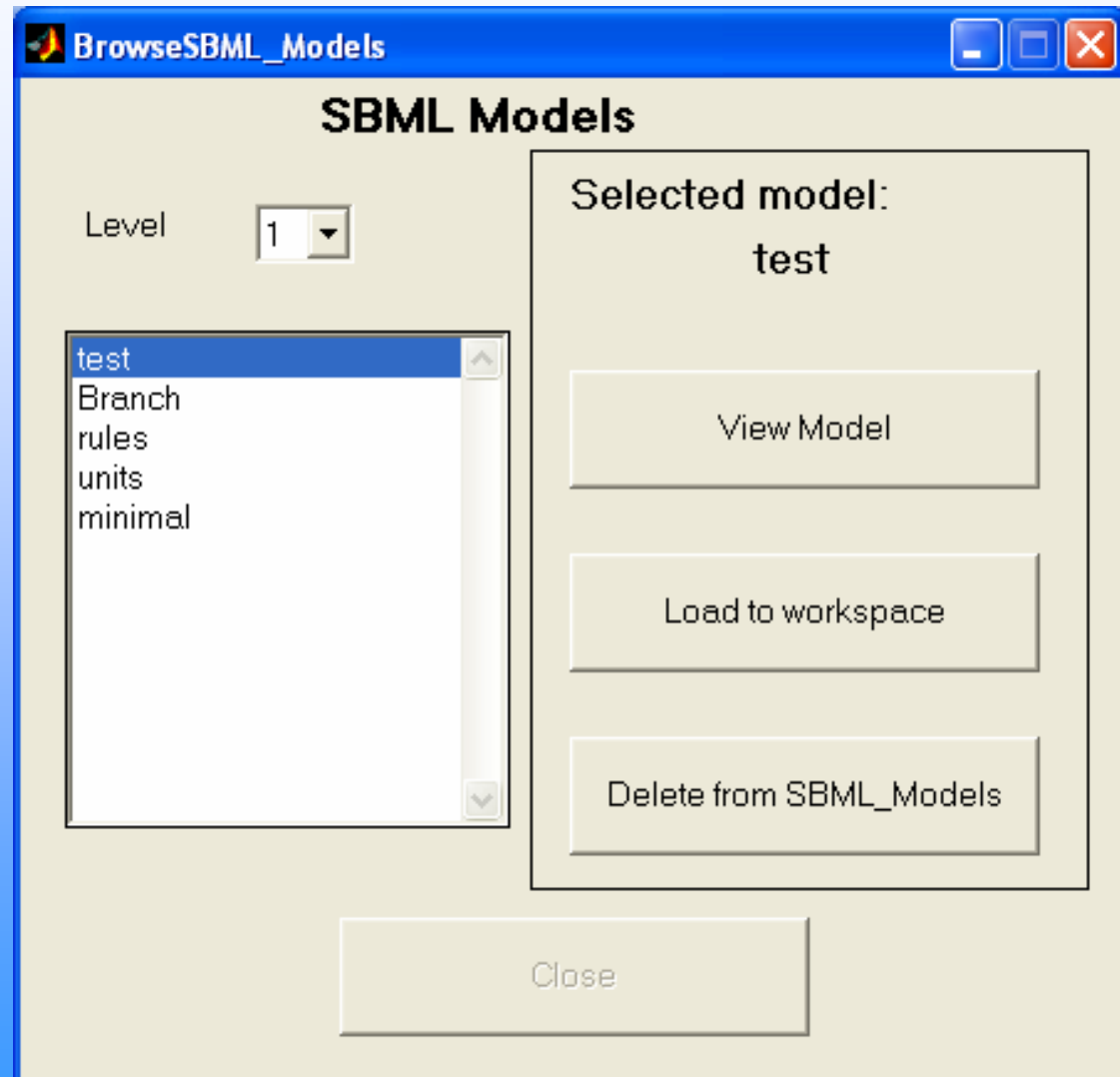
Storing SBML Models in MATLAB

Functions (MATLAB .m scripts)

- save to
- load from
- delete from
- list models in

a MATLAB data file (.mat)

or use the viewing options



MATLAB_SBML_Model [-] [x]

Model: BoundaryCondExampleModel

SBML Level 2 Version 1

Reactions	<input type="text" value="reaction_1"/>	<input type="button" value="View Reaction"/>
Species	<input type="text" value="S1"/>	<input type="button" value="View Species"/>
UnitDefinitions	<input type="text" value="NO UNIT DEFINITIONS"/>	<input type="button" value="View Unit Definition"/>
Compartments	<input type="text" value="compartmentOne"/>	<input type="button" value="View Compartment"/>
Parameters	<input type="text" value="k1"/>	<input type="button" value="View Parameter"/>
Rules	<input type="text" value="Rule1"/>	<input type="button" value="View Rule"/>
Functions	<input type="text" value="NO FUNCTION DEFINITI"/>	<input type="button" value="View Function"/>
Events	<input type="text" value="NO EVENTS"/>	<input type="button" value="View Event"/>

Species

Species: S1

Compartment compartmentOne

Boundary Condition

Constant

Charge

Initial concentration

Value

Substance units

Spatial size units

Close

Edit boxes are disabled at present

Matlab applications

- Stability and robustness
 - Model (in)validation
 - Parameter value determination
-
- Data fitting
 - Data interpolation ???

Control and Dynamical Systems

CALTECH



MATLAB Toolbox

– solving sum of squares problems

Lyapunov stability

- determines whether an equilibrium point is stable

If $x = 0$ is equilibrium for $\dot{x} = f(x)$

If can find a function $V(x)$ such that

$$\begin{array}{l} V(0) = 0 \\ V(x) > 0 \end{array} \quad \frac{\partial V}{\partial x} \dot{x} \leq 0$$

Then $x = 0$ is stable

- can be done with SOSTools

$$\dot{x} = \begin{bmatrix} -x_1^3 + x_2 \\ -x_1 - x_2 \end{bmatrix}$$

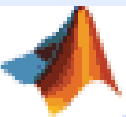
 `syms x1 x2`

```
V = findlyap([-x1^3+x2; -x1-x2], [x1;x2], 2)
```

V =

```
1.2964*x1^2+.21232e-4*x1*x2+1.2966*x2^2
```

- bridge gap between an SBML model and SOSTools



`stable = IsStableEquilibrium(SBMLModel)`

```
stable =  
      1
```

```
stable =  
      0
```