mod2sbml.py, Version 3.1.1.1 — an SBML-shorthand to SBML translation tool

Darren J Wilkinson
School of Mathematics and Statistics
University of Newcastle upon Tyne, UK
email: d.j.wilkinson@ncl.ac.uk

February 24, 2024

1 Introduction

This document describes a reference implementation of a SBML-shorthand to SBML compiler: `mod2sbml.py`, written in Python. It can be used either as a simple command-line translator, or by Python programmers as a Python module. The module requires libSBML version $\geq$ 4.1.0 (and $<$ 5.0.0) and the libSBML python bindings. Please ensure that these are installed and working before attempting to use this software. Note that the libSBML API is currently undergoing a period of rapid change, and so it really does matter which version of libSBML you have installed.

SBML-shorthand is described in a separate specification document.

There is also a reference implementation of a SBML to SBML-shorthand translator (`sbml2mod.py`) available from the SBML-shorthand web site. This works (almost) exactly like `mod2sbml` except that the conversion is in the other direction. This document therefore serves as documentation for that script too (but see the `__doc__` strings for further info).

2 Version

mod2sbml.py has a version number of the form $a.b.c.d$ where $a.b.c$ corresponds to the version number of SBML-shorthand that the software targets, and d is the version of the software for that SBML-shorthand version. So, version 3.1.1.1 is the first version of the software aimed at SBML-shorthand version 3.1.1.

3 Command-line filter

Suppose that a SBML-shorthand model is contained in a file called `mymodel.mod` in the current directory. Then either of the commands

```
% mod2sbml.py < mymodel.mod > mymodel.xml
% mod2sbml.py mymodel.mod > mymodel.xml
```

will result in the production of the SBML `mymodel.xml` corresponding to the shorthand description in `mymodel.mod`. This obviously requires that `mod2sbml.py` is executable and in the current path. If you don't know what that means, putting the `mod2sbml.py` file in the current directory and typing

```
% python mod2sbml.py mymodel.mod > mymodel.xml
```

should still work.

4 Python module

Python programmers can also use this module directly from python. A couple of example sessions are given below. First, an example using file streams.

```
>>> from mod2sbml import Parser
>>> p=Parser()
>>> inS=open("mm.mod","r")
>>> d=p.parseStream(inS)
>>> print d.getModel()
>>> print d.toSBML()
```

Next, an example using strings.

```
>>> from mod2sbml import Parser
>>> p=Parser()
>>> s=open("mm.mod","r").read()
>>> d=p.parse(s)
>>> print d.toSBML()
```

The returned `libsbml` object, `d`, is a libSBML document. The module raises the exception `ParseError` if it encounters a fatal error during parsing.

Note that Python `__doc__` strings are provided for the module, the Parser class, and the two public methods of the Parser class. For further details, see the source code...

5 Links

SBML-shorthand	`www.staff.ncl.ac.uk/d.j.wilkinson/software/sbml-sh/`
SBML	`sbml.org`
Python	`www.python.org`