BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–2

libsbmlwrapper: R bindings for libSBML

John Casement *[1], Colin Gillespie [2] and Frank Bergmann [3]

[1]Bioinformatics Support Unit, Newcastle University.
[2]School of Mathematics and Statistics, Newcastle University.
[3] University of Heidelberg, or SBML Team?

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT
Motivation: R bindings for libSBML based on SWIG code generation were made available in 2012, providing an interface between R and libSBML and offering basic functionality. However, the bindings are not user friendly and do not offer generic functions that an R user might expect to have available.
Results: The R package libsbmlwrapper has been developed to address this issue by wrapping the bindings and enabling libSBML functions to be called via R-like commands from the console. This paper describes functions offered by the package and how they can be applied to view, extract information from and manipulate SBML documents and models within R.
Availability: libsbmlwrapper can be installed from GitHub at https://github.com/jwcasement/libsbmlwrapper.
Contact: john.casement@newcastle.ac.uk

1 INTRODUCTION

The Systems Biology Markup Language (SBML) is a modelling language for representing models of biochemical reaction networks. It can be applied to metabolic networks, cell signalling pathways and regulatory networks (Hucka *et al.*, 2003). SBML is a free and open interchange format providing a machine-readable "common intermediate format" to enable the most important aspects of models to be communicated.

libSBML (Bornstein *et al.*, 2008) is a programming library for reading, writing, manipulating, translating and validating SBML. It is written in ISO standard C and C++ but provides language bindings allowing it to be used from other programming languages including MATLAB, Java and Python. libSBML version 5.6.0 (released on 3rd August 2012) was the first to include a binding to R, generated using the Simplified Wrapper and Interface Generator (SWIG) (Beazley, 2003).

The SWIG generated bindings offer a range of functions which make it possible to create, manipulate and validate SBML documents from the statistical programming environment R. However, the format of the SWIG functions is not in keeping with typical R functions, and their use may prove discouraging to non-programmers. For example, to view the number of each component within a loaded model `m` using SWIG functions would require a series of function calls such as `Model_getNumFunctionDefinitions(m)` and `Model_getNumUnitDefinitions(m)`. To get the attributes of a species within the model m would require a series of function calls such as `sp = Model_getSpecies(m,i)` to get the (i+1)th species, and `Species_getId(sp)` to get the species Id.

These examples highlight that using the SWIG functions can be time consuming. Many of the commands are also prone to causing program crashes if not used carefully. In the example above, the call to `Model_getSpecies(m,i)` will cause a system crash if an invalid value of i is used, so this would have to be preceded by a call to `Model_getNumSpecies(m)` to find the range of valid index values. Additionally, there is the danger of out of bounds errors (resulting in inaccurate results or program crashes) due to the difference in indexing methods between R (indexes start at 1) and C and C++ (indexes start at 0).

Such issues provided the motivation to develop an R package that makes utilizing the SWIG generated bindings more user friendly by incorporating bounds checking and providing generic R functions such as `summary()` and standard indexing via "`[`" and "`[[`", thus providing a more native feel for R users in comparison to the functions provided by SWIG. The libsbmlwrapper package is intended to be a reliable and user friendly tool for viewing and editing SBML documents in R.

2 METHODS

To read SBML content, an R object of class _p_SBMLDocument is created via a call to the libsbmlwrapper function `readSBML` (if reading a document from a file) or `readSBMLFromString` (if reading a document as an XML string). The corresponding libSBML functions `readSBML` and `readSBMLFromString` are masked when libsbmlwrapper is loaded, because the libsbmlwrapper functions perform additional operations to generate and store default values for component attributes which may be required when adding components. The model stored within the SBML document is created as an R object of class _p_Model via a call to the libSBML function `SBMLDocument_getModel`.

To enable selection from a _p_Model object, libsbmlwrapper defines an implementation of the generic extract function "`[[`". Components can be selected by index or by a selection string. Valid index values and selection strings are shown in Table 1 (columns 1 and 2). For example, to return a list containing all species in the model m, use `m[[ListOfSpecies]]`. Equivalently, this could be achieved via `m[[6]]`, using the index number corresponding to the ListOfSpecies container.

All components within the model object are grouped within such listOf containers. libsbmlwrapper implements a method to coerce these containers to R_ListOf objects which possess all of the properties of R lists. Table 1

*to whom correspondence should be addressed

Table 1. Selection from a _p_Model object.

Index	Selection string (model)	Class of selected object(s)	Selection string (attributes and math)
1	ListOfFunctionDefinitions	_p_FunctionDefinition	Id, Name, Math
2	ListOfUnitDefinitions	_p_UnitDefinition	Id, Name
3	ListOfCompartmentTypes	_p_CompartmentType	Id, Name
4	ListOfSpeciesTypes	_p_SpeciesType	Id, Name
5	ListOfCompartments	_p_Compartment	Id, Name, SpatialDimensions, Size, Units, Constant
6	ListOfSpecies	_p_Species	Id, Name, Compartment, InitialAmount, InitialConcentration, SubstanceUnits HasOnlySubstanceUnits, BoundaryCondition, Constant, ConversionFactor
7	ListOfParameters	_p_Parameter	Id, Name, Value, Units, Constant
8	ListOfInitialAssignments	_p_InitialAssignment	Symbol, Math
9	ListOfRules	_p_Rule	Id, Formula, Type, Units, Variable, Math
10	ListOfConstraints	_p_Constraint	Math
11	ListOfReactions	_p_Reaction	Id, Name, Reversible, Fast, Compartment
12	ListOfEvents	_p_Event	Id, Name, UseValuesFromTriggerTime

(column 3) shows the class of the objects contained in the list, which can be selected via "`[[`" to return a single object, or "`[`" to return another list. (This is in keeping with the usual form of indexing in R.) For example, to return an R_ListOf object containing the first three species from the model m, use `m[[6]][1:3]`. Towards fulfillment of the aim of improving user-friendliness, libsbmlwrapper performs bounds checking to prevent program crashes, and prints informative message strings if invalid index values are used. A further application of "`[[`" allows specific attributes to be selected. Column 4 of Table 1 shows valid selection strings for each object listed in column 3. For example, to return the value of the InitialAmount attribute of the 3rd species, use `m[[6]][[3]][["InitialAmount"]]`. In addition to attribute names, the string "Math" can also be used here for components that might contain math elements. Selection of a "Math" element returns an object of class _p_ASTNode, containing the math element of the component as an abstract syntax tree. When _p_ASTNode objects are returned, libsbmlwrapper informatively displays the formula contained in the object as a string.

Several components within the _p_Model container contain other components (for example, the "UnitDefinition" component contains a "Unit" component). Selection is via "`[[`" as before, although in these cases not all selected objects are contained in lists.

libsbmlwrapper defines an implementation of the generic summary function for objects of class _p_SBMLDocument, _p_Model, _p_Component, and for R_ListOfs containing these objects. Use `summary(d)` to display top level information (level, version, xml namespace and number of errors) for the loaded SBML document d, and `summary(m)` to display all model attributes, including the number of instances of each component present in the loaded model m. `summary(m[[6]])` compactly displays all attributes of all species in the model m. All attributes of (say) the 3rd species can be viewed via `summary(m[[6]][[3]])`.

An implementation of the generic replace function "`[[<-`" enables attributes of a component to be edited. To edit the value of an attribute, first select it via "`[[`" then assign the desired value. In the event that an invalid value is assigned (such as a duplicate Id) libsbmlwrapper will communicate the appropriate libSBML operation return value along with an informative string stating the error.

Generic functions named after create__ are defined for creating component objects within the _p_Model container. For example, to add a new species to the model m, use `createSpecies(m)`. Copies of existing components in the model can be added by supplying index values to indicate which instances to copy. For example, `createParameter(m,1))` appends a copy of the first parameter to the model `m`. Supplying an index value of zero results in a new object with attributes initialised to typical defaults being added, meaning that the command `createParameter(m,0)` is equivalent to `createParameter(m)`. Components can be deleted by selecting them and then setting to NULL. For example, to delete the 5th reaction, use `m[[11]][[5]] = NULL`.

3 CONCLUSION

R is becoming a de facto standard for statistics and biology. By enabling an interface between SBML and R in a way that is straightforward and consistent with standard R functionality, the libsbmlwrapper package offers non-programmers a means of accessing data in models. As a package that builds directly onto libSBML it offers a number of advantages (in addition to being much easier to maintain as new versions of libSBML are released). libSBML is built for speed, is well maintained and is extensively tested to ensure reliability. The development of the libsbmlwrapper package has demonstrated that, despite issues of user-friendliness, the SWIG generated bindings do offer a sufficient means of enabling the powerful functionality of libSBML to be exploited within R. Many more of the SWIG functions could be incorporated into libsbmlwrapper so that they could be utilised in a more user-friendly and R-like way. Further development of the libsbmlwrapper package is both possible and desirable.

REFERENCES

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A. , Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novre, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J. (2003). The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524531.

Bornstein, B. J., Keating, S.M., Jouraku, A., and Hucka M. (2008) LibSBML: An API Library for SBML. *Bioinformatics*, **24**, 880881.

Beazley, D. M. (2003) Automated scientific software scripting with SWIG. *Future Generation Computer Systems*, **19**, 599609.