

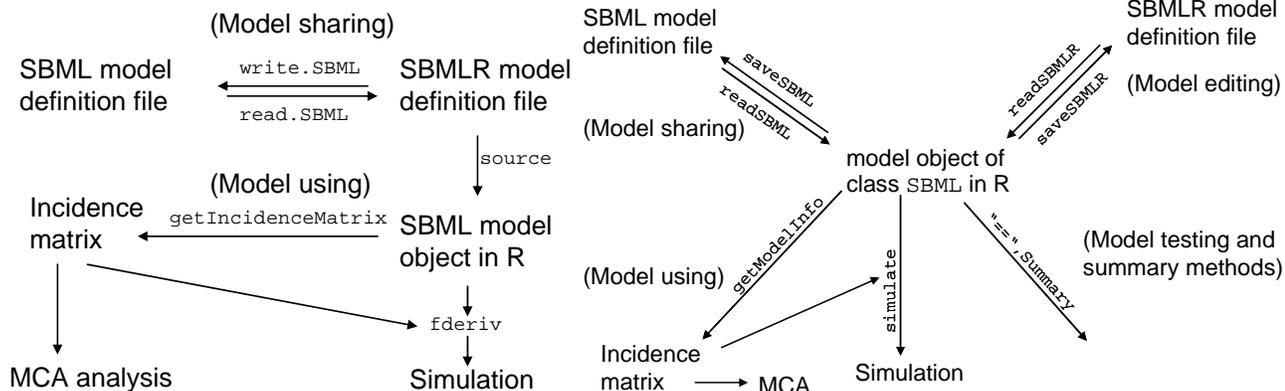
# Systems Biology Markup Language for R

## SBMLR Version 1.21 (5/5/2005)

### Update Information

Tomas Radivoyevitch, Department of Epidemiology and Biostatistics, Case Western Reserve University, Cleveland, Ohio 44106; [radivot@hal.cwru.edu](mailto:radivot@hal.cwru.edu); <http://epbi-radivot.cwru.edu/>

This report briefly describes improvements in SBMLR which have taken place between version 1.16 (2/1/2005) and version 1.21 (5/5/2005). Compared to the previous 1.16 architecture (figure 1), the new 1.21 architecture (figure 2) focuses on model objects of class SBML. These SBML objects include



equivalent MathML, R, string, and function representations for rate laws and rules. SBMLR model files used to edit models have been streamlined to include only string rate laws and rules, these being converted to their equivalent counterparts by `readSBMLR`. Parenthesis overkills that made rate law expressions cryptic after `read.SBML` file conversions no longer arise in `readSBMLR` | `saveSBMLR` file conversions. Two methods for SBML model objects, `summary.SBML` and `"==" .SBML`, present and compare, respectively, species and reactions tabularized as R data frames. The function `fderiv` is now embedded in `simulate`, a wrapper for `lsoda` of the `odesolve` package. The wrapper arguments are the model, the simulation reporting times, and optionally a  $V_{\max}$  modulator matrix or a list of  $V_{\max}$  modulator interpolating functions. Another function, `getModelInfo`, extracts from models a list of key information variables, including the incidence matrix. The list can be conveniently attached (using R's `attach`) and detached (`detach`) to make its contents available as needed in one fell swoop. This feature is used in `simulate`. The net result is that model use scripts are now clearer.

**Preliminary Findings:** It was conjectured in Specific Aim 1 that R codes might take simpler forms if SAX-based XML parsing (`xmlEventParse`) was used instead of DOM-based parsing (`xmlTreeParse`). Preliminary results suggest that this may be true for much of the SBML document, but that it is not true for mathML expressions, primarily because of their recursive nature. Parse times were minimal (<1 second) for both SAX and DOM methods and thus not an issue. For the reverse mapping of SBML R objects to SBML files, it was postulated in Aim 1 that, to enhance portability, the SBML DOM should first be built as an R object using `addTag` and `closeTag`, and then saved as an SBML file using `saveXML`. Preliminary results suggest that this may be too slow (~5 seconds compared to <1 for sample models). Thus, `saveSBML` currently writes SBML files incrementally, much as in `write.SBML` of version 1.16.