

A portable library to support the SBML Layout and Render Extension

J. Kyle Medley^{1*}, Kiri Choi¹, Natalie Hawkins¹ and Herbert M. Sauro¹

¹Department of Bioengineering, University of Washington, Seattle, WA 98195, USA.

ABSTRACT

Motivation: The SBML layout extension enables SBML models to encode layout information which describes the graphical depiction of model elements, and the render extension enables SBML models to encode style elements. In this application note, we describe libSBNW and libsbmlDraw, portable libraries that support the SBML layout and can automatically generate layout for SBML models. Additionally, libsbmlDraw supports the SBML render elements of color, line widths, and font information for compartments, species, and reactions.

Results: The libraries can be used to automatically generate layout information for SBML models lacking it, or to edit coordinate information for the nodes already encoded in a model. We provide C and Python APIs to allow other applications to host the library or to use it directly from the Python console. We show that both libraries are sufficient for creating a graphical application for displaying and editing layout information. libsbmlDraw allows editing of render information.

Availability: The libraries are open-source, libSBNW is licensed under the BSD 3-clause license and libsbmlDraw under the MIT license. Project source code, downloads, documentation and binaries for Windows and Mac OS X are available, for libSBNW, at: <https://github.com/0u812/sbnw>. The library is also included in Tellurium, available at <http://tellurium.analogmachine.org/>. Video tutorials for libSBNW are available at <http://0u812.github.io/sbnw/tutorials/>. The code for libsbmlDraw can be found at: <https://github.com/sysbio/libsbml-draw>. Documentation for libsbmlDraw is at: <https://libsbml-draw.readthedocs.io/en/latest/index.html>.

Contact: medleyj@uw.edu, for libsbmlDraw nhawkins@uw.edu

1 INTRODUCTION

SBML (Hucka et al., 2003) is the de facto standard for exchanging biochemical network models (Sauro, 2014). The SBML effort has spawned a great variety of extensions and other formats (Dräger et al., 2014). One such extension is the SBML layout extension (Gauges et al., 2006), which is embedded in an SBML document. The extension allows software to describe the graphical layout of a biochemical network in terms of species, reactions, compartments and modifiers. The render extension provides information on the detailed rendering of the network, i.e. colors, line width, and font information (Gauges, 2009).

However, many SBML models contain no layout information (Bergman and Sauro, 2006), and this is a barrier to developing graphical software tools for interacting with SBML. Therefore, it is desirable to have a library for assigning layout information to

SBML models created before the introduction of the layout extension, and to provide an easy way to encode layout information in future SBML models.

In this application note we present an updated library based on the original SBW layout engine (Deckard et al., 2006). The original library was written in C# and was tailored specifically for use by SBW. Here, we describe a new version, written in C/C++, with fewer dependencies, Python bindings, sample applications and additional enhancements.

2 METHODS

SBML Layout Extension Support Both libraries are designed for compatibility across SBML level 2/3, and use libSBML for reading and writing content (Bornstein et al., 2008). The SBML layout extension is used to store visual layout information. If the input model does not have a layout, one can be automatically generated by the library.

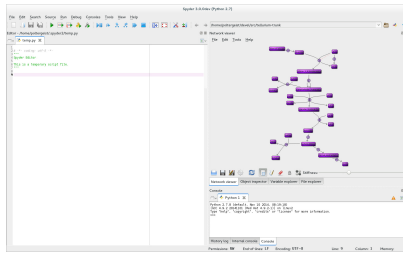
Autolayout Algorithm Both libraries automatically generate layout information encoding node and reaction centroid coordinates using the Fruchterman-Reingold (FR) algorithm, which has been shown to be robust in the face of variegated graph topologies and faithfully reproduces the underlying symmetry (Fruchterman and Reingold, 1991).

Selectively lock Nodes Users can specify one or more nodes to be locked when the layout algorithm is executed. This ensures that the positions of these nodes do not change when the layout algorithm is applied. Node locking is particularly useful for users who want to fine-tune the automatic layout process.

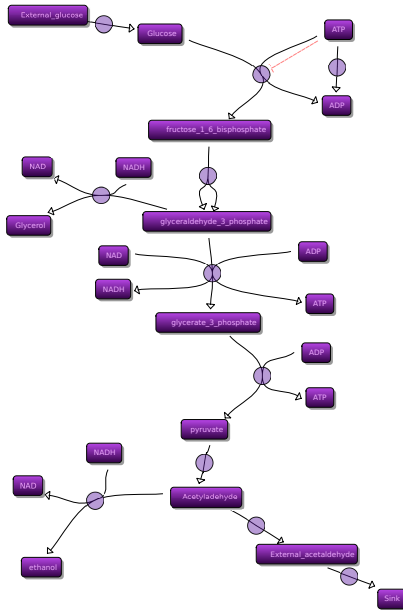
Support for Alias Nodes Models with a high degree of connectivity can be problematic for visualization methods due to the overlap between edges that occurs when the underlying graph is embedded in 2D space. We solve this problem by providing the user with the ability to create *alias nodes*. In general, any node of degree n can be decomposed into n alias nodes, each of which is rendered separately. Using this method, we can reduce the connectivity of any network graph. The FR-algorithm is particularly adept at laying out such reduced graphs without overlapping edges (Fruchterman and Reingold, 1991), enabling lucid visualization of complex networks.

SBML Render Extension Support libsbmlDraw supports colors, line widths, and font information for compartments, species, and reactions in an SBML model.

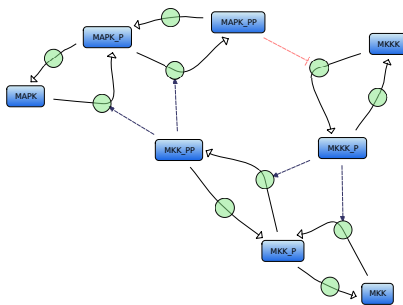
*to whom correspondence should be addressed



(a) PyQt-based viewer displaying the layout of the glycolysis model (Wolf and Heinrich, 2000) in the Tellurium/Spyder environment.



(b) A rendered version of the model displayed above



(c) MAPK signaling cascade (Kholodenko, 2000)

Fig. 1. Demonstration of the Python-based network viewer

Export of Rendered Model A rendered model may be exported to a raster format (Portable Network Graphics) or one of several vector formats (Scalable Vector Graphics, TikZ (libSBNW), PDF (libsbmlDraw)).

Supply Ancillary graphical Information Visual features such as arrowheads/endcaps are supplied by libSBNW, and may be used to

render network diagrams as shown in Figure 1. Five endcap styles are supported and are tied to species roles in SBML reactions: Substrate, product, modifier, activator and inhibitor.

Language Bindings Public APIs for C and Python are provided. For example, using libSBNW, the following Python code loads an SBML model, applies the FR autolayout algorithm to the network, and saves the result as `output.xml` in the current directory:

```
# import the sbnw Python module
import sbnw
# load the model
model = sbnw.loadsbml('model.xml')
# seed node coordinates randomly
model.network.randomize()
# apply the FR-algorithm
model.network.autolayout()
# Save new SBML to file
model.save('output.xml')
```

Using libsbmlDraw, the following Python code loads an SBML model, applies the FR autolayout algorithm to the network, and saves the result as `output.xml` in the current directory:

```
# import the libsbmlDraw Python module
import libsbml_draw
# load the model
model = SBMLlayout('model.xml')
# Save new SBML to file
model.writeSBMLFile('output.xml')
```

3 RESULTS

We have deployed libSBNW in the context of the Tellurium biological modeling environment (<http://tellurium.analogmachine.org/>) using the Tellurium's Spyder-based plugin architecture. Tellurium includes the `nwed` (network editor) Python module, which may be used to communicate with the layout plugin, and the SBML model may be set or retrieved via `nwed.setsbml` and `nwed.getsbml` respectively.

```
sbmlStr = '''...'''
import nwed
nwed.setsbml(sbmlStr)
```

ACKNOWLEDGMENTS

We acknowledge Stanley Gu for his shared web development expertise, Lucian Smith for his work on the Antimony language, Jonathan Gallimore for his work on the C API, Andy Somogyi for his shared expertise in Python extension modules, Kaylene Stocking for testing the network viewer, and Joshua Katz for testing libsbmlDraw.

Funding: The authors are most grateful to generous funding from the National Institute of General Medical Sciences of the National Institutes of Health under award R01-GM081070. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

- Bergmann, F. T., and Sauro, H. M. (2006) SBW-a modular framework for systems biology, *Proceedings of the 38th conference on Winter simulation*, 1637–1645.
- Bornstein, B., Keating, S., Jouraku, A., and Hucka (2008) LibSBML: an API Library for SBML, *Bioinformatics*, **24** 6, 880–881.
- Deckard, A., Bergmann, F.T and Sauro H.M. (2006) Supporting the SBML layout extension, *Bioinformatics* **22** 23 2966–2967.
- Dräger, A and Palsson B (2014) Improving Collaboration by Standardization Efforts in Systems Biology, *Frontiers in Bioengineering and Biotechnology* **2** 61.
- Fruchterman, T.M.J. and Reingold, E.M. (1991) Graph drawing by force-directed placement, *Software: Practice and Experience*, **21** 11, 1129–1164.
- Gauges, R, *et al* (2006) A model diagram layout extension for SBML, *Bioinformatics* **22** 15 1879–1885.
- Gauges, R (2009) Complementing layout information with render information in SBML files. University of Heidelberg. Available at <http://otto.bioquant.uni-heidelberg.de/sbml/level2/20091029/SBMLRenderExtension-20091029.pdf>
- Hucka, M, Finney, A, Sauro, H.M, Bolouri, H, Doyle, J, Kitano, H and the rest of the SBML Forum. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**(4), 524–531.
- Kholodenko, B. N. (2000) Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades, *European journal of biochemistry / FEBS*, **267** 6, 1583–1588.
- Sauro H. M. (2014) *Systems Biology: An Introduction to Pathway Modeling*, Ambrosius Publishing, ISBN-10: 0982477376
- Wolf, J and Reinhard H (2000) Effect of cellular interaction on glycolytic oscillations in yeast: a theoretical investigation, *Biochemical Journal*, **345**, 321–334.