SigNetSim : A web-based framework for designing kinetic models of molecular signaling networks

Vincent Noël[1], Marcelo S. Reis[1], Matheus H.S. Dias[1], Lulu Wu[1,2], Amanda S. Guimares[1,2], Daniel F. Reverbel[2], Junior Barrera[1,2], and Hugo A. Armelin[1,3]

[1]Center of Toxins, Immune-response and Cell Signaling (CeTICS), Instituto Butantan, Brazil
[2]Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil
[3]Instituto de Química, Universidade de São Paulo, Brazil

Introduction and objective

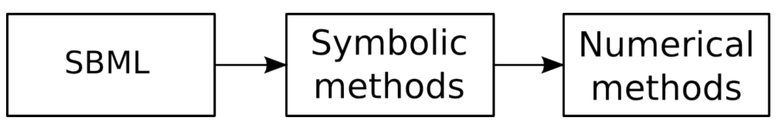
Molecular biology is experiencing a revolution, thanks to data acquisition becoming increasingly cheaper, and to the development of Systems Biology, an emergent research field that shows new ways to study such high-throughput data. However, biologists need a new generation of tools capable of handling large quantitative data sets, and also performing rigorous mathematical analysis of the kinetic models that may be derived from them.

SigNetSim is a framework whose main component is a web tool through which a user can design and modify models, perform simulations, input experimental data into a simple database and use it to adjust models them, and perform dynamical analysis. More advanced users can directly use its core library, libSigNetSim, to build, adjust and analyze quantitative biological models.

libSigNetSim

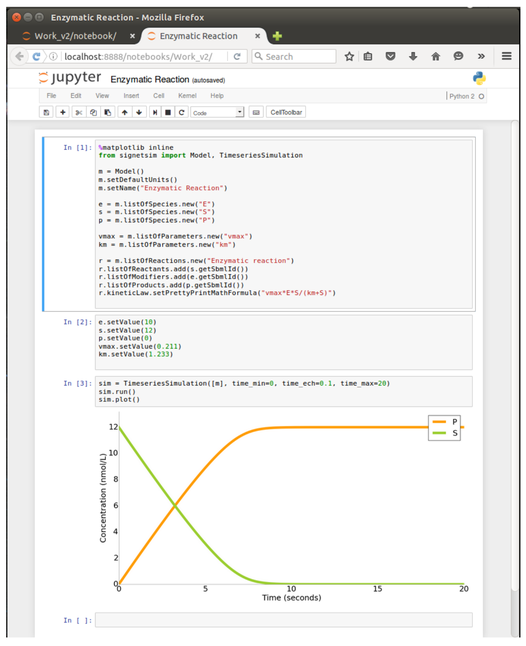
libSigNetSim is a library coded in Python 2.7, whose purpose is to create or edit biological quantitative models, and is compatible with all versions of SBML. It can also notably make use of the Hierarchical Model Composition package to write more modular, reusable models.

Once loaded, the model can be converted into a formal mathematical representation, using the computer algebra system Sympy, thus allowing usage of formal methods on the model. This enable the user to look for exact solutions, find stationnary states, perform model reduction.

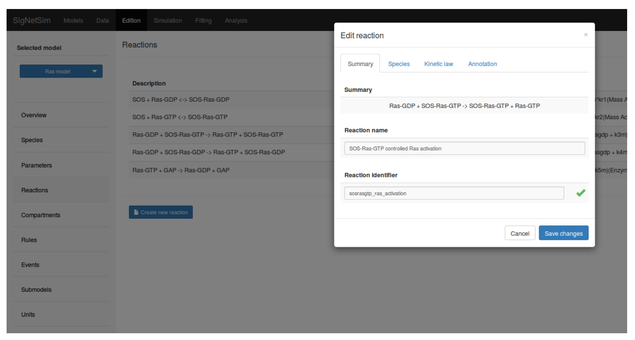


When numerical methods are necessary, libSigNetSim can generate C code, which can be executed in parallel. To perform numerical integration of the system of differential equations (ODE or DAE), the library Sundials is used. For model fitting, libSigNetSim use a parallelized simulated annealing[1]. This algorithm is particularly suited for optimizing large non-linear systems.

libSigNetSim is designed to be run directly on computational servers. While its purpose is mainly to be used via the web interface, libSigNetSim can also be used directly in a python environment like Jupyter Notebook. Users can build models with a few lines of code. One advantage of this use case is to be able to analyse the model directly with computer algebra system Sympy.

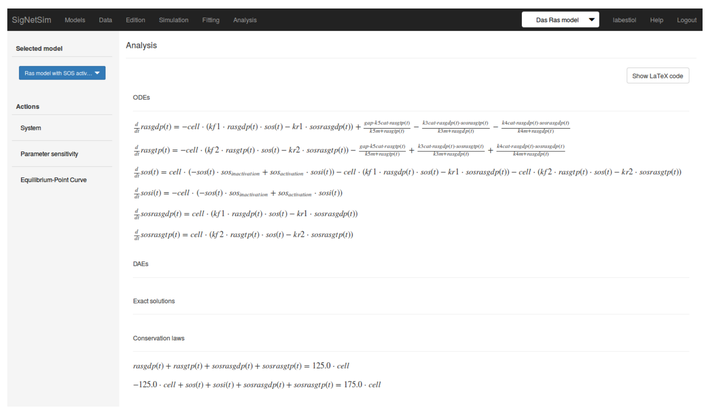


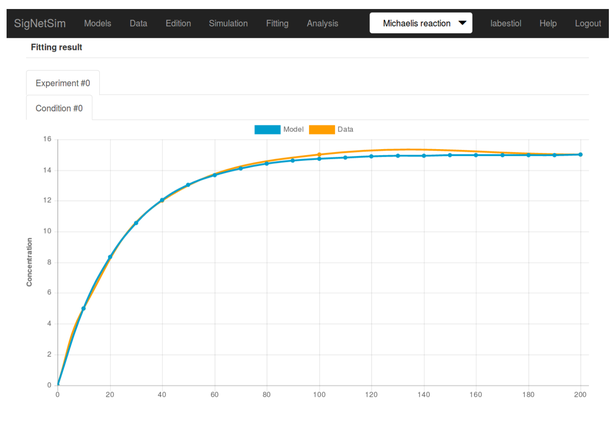
Web Interface



SigNetSim is a web interface build with the Django framework, which uses Bootstrap as a front-end framework. It is designed to function in various devices, from smartphone to desktop computer.

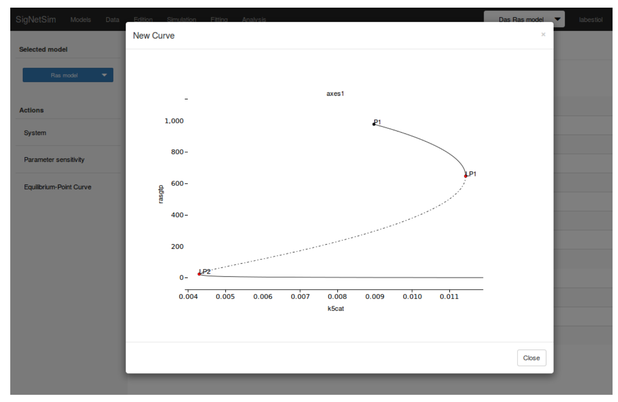
SigNetSim uses various other JavaScript libraries to be able to plot simulations, or display the mathematical model. Users can also easily export the mathematical model description as LaTeX.

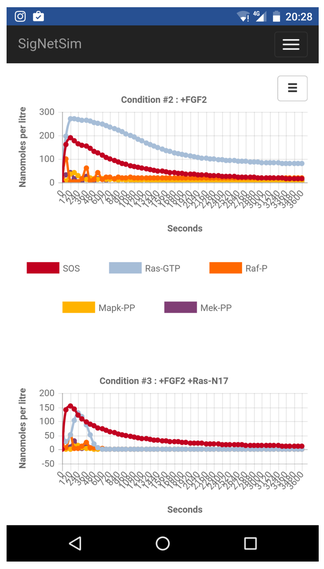




A simple database is available to store experimental data, which can later be used to fit models. These optimisation can be done to reproduce the result of various initial conditions.

SigNetSim can perform various dynamical analysis, including bifurcation analysis, in a simple and intuitive way. This makes possible to look for emergent properties of the model, like bistability, or the existence of limit cycles (oscillations)





SigNetSim can perform multiple simulations, using the different initial conditions stored into the database. The Javasript plotting library Chart.js produces beautiful, interactive plots of the timeseries, which can later be saved as image.

The web interface is fully compatible with smartphones, enable the user to quickly use SigNetSim from anywhere.

References

1. Chu, K. W., Deng, Y., Reinitz, J. (1999). Parallel simulated annealing by mixing of states. Journal of Computational Physics, 148(2), 646-662.

Acknowledgements

FAPESP
CNPq

Conclusions and future work

- Both libSigNetSim and SigNetSim will soon be open-sourced and available on GitHub.
- We are in the process of implementing SED-ML to be able to store simulations, and produce COMBINE archives.
- With SigNetSim now able to handle a large class of models, we are planning to add more dynamical analysis tools.
- We also intend to implement annotations and a SBGN viewer.

http://vincent-noel.fr/signetsim/
vincent.noel@butantan.gov.br