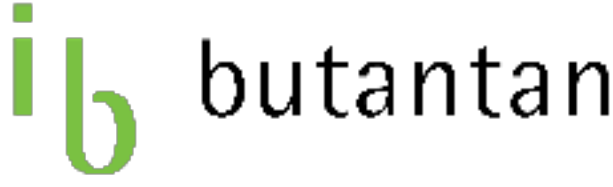
SigNetSim

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. Guimarães[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, São Paulo, Brasil
[2]Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil
[3]Instituto de Química, Universidade de São Paulo, Brazil





Introduction

- Instituto Butantan

  - Biomedical center founded in 1901
  - Research and production center (anti-venoms, vaccines)
  - Large collection of snakes, spiders, scorpions, ...
  - Chosen by the OMS as a center to develop vaccines for Zika





Introduction

- Need for a web tool
  - Easy to use for biologist/students
  - Able to do math
  - Use a parallel simulated annealing algorithm
  - Able to do model reduction
  - Something to do waiting for experimental data

Introduction

- Structure

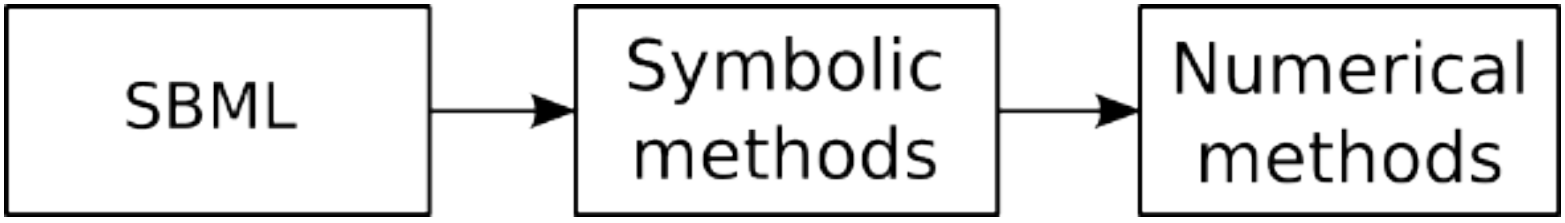
The SigNetSim framework is divided into two main components :

- libSigNetSim, the core library
- SigNetSim, the web interface

Both are coded in Python 2.7, with some part of the library in C

libSigNetSim

- Pipeline



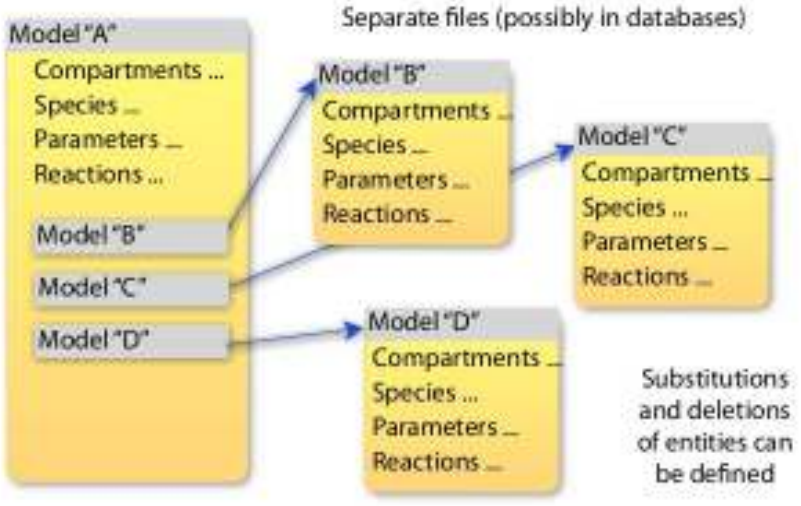
- Sbml Model is read
- Symbolic math model is built (Sympy)
- Math is written as C files

libSigNetSim

- SBML Compatibility

- Compatible with SBML version up to L3V1 (except CSymbolDelay)

- Also support the Hierarchical Model Composition package (comp)



libSigNetSim

› Simulation

The integrations are performed using C-generated code, which can be executed in parallel.

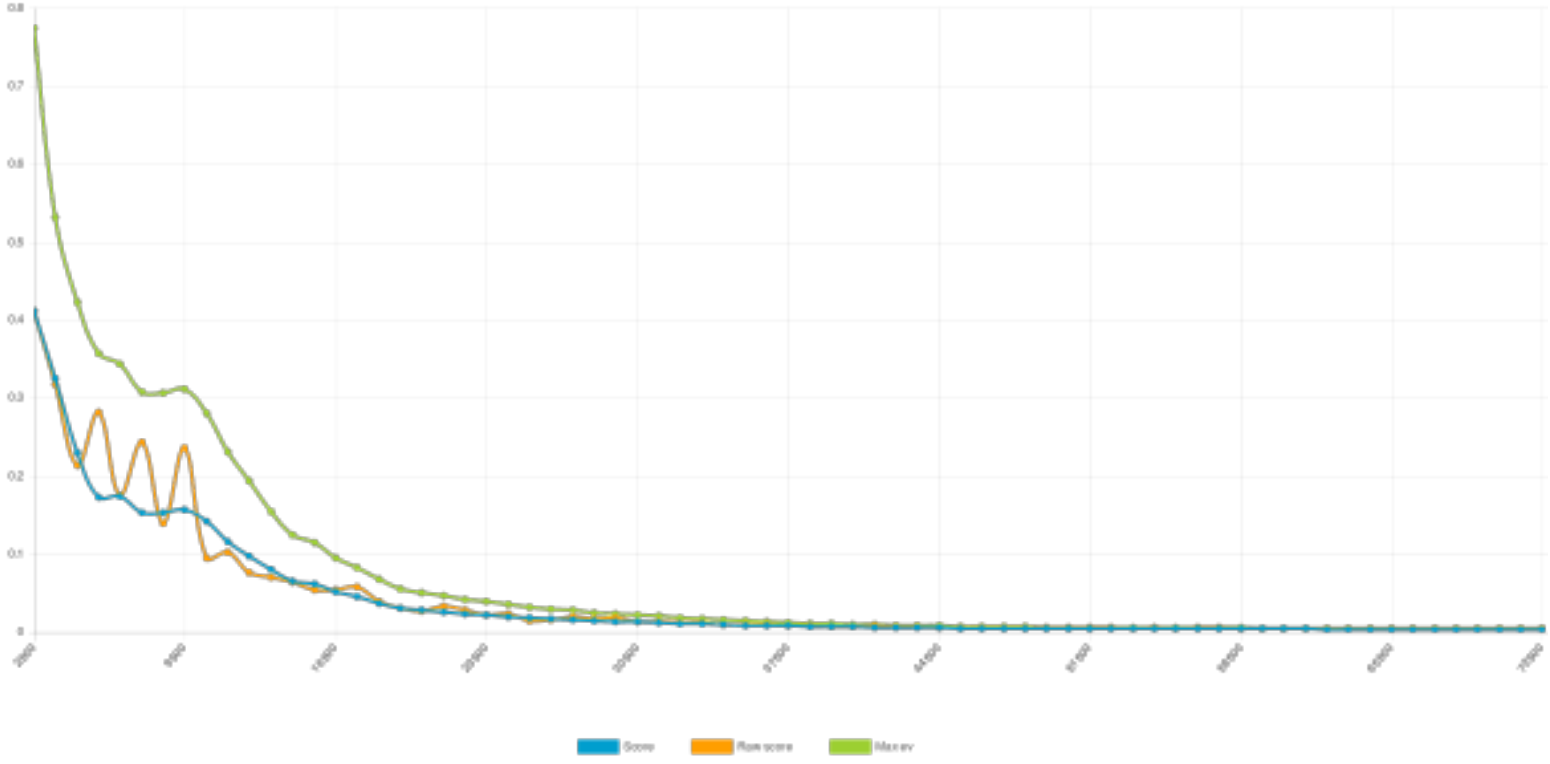
To perform numerical integration for systems of differential equations (ODEs or DAEs), libSigNetSim uses the Sundials library (CVODE or IDA).

Before building the C code, the model is reduced to minimize the computational ressources.

libSigNetSim

› Optimization

To perform model fitting, libSigNetSim uses parallel simulated annealing [1].



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

› Analysis

$$\frac{d}{dt}E(t) = -E(t) \cdot S(t) \cdot k_f + ES(t) \cdot k_r + ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}ES(t) = E(t) \cdot S(t) \cdot k_f - ES(t) \cdot k_r - ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}P(t) = ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}S(t) = -E(t) \cdot S(t) \cdot k_f + ES(t) \cdot k_r$$

- Analysis

$$\frac{d}{dt}E(t) = -E(t) \cdot S(t) \cdot k_f + ES(t) \cdot k_r + ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}ES(t) = E(t) \cdot S(t) \cdot k_f - ES(t) \cdot k_r - ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}P(t) = ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}S(t) = -E(t) \cdot S(t) \cdot k_f + ES(t) \cdot k_r$$

$\Rightarrow$Continuation techniques don't like that

› Analysis

$$\frac{d}{dt}E(t) = -E(t)\cdot S(t)\cdot k_f + \cdot ES(t)\cdot k_r + ES(t)\cdot k_{cat}$$

$$\frac{d}{dt}ES(t) = E(t)\cdot S(t)\cdot k_f - ES(t)\cdot k_r - ES(t)\cdot k_{cat}$$

$$\frac{d}{dt}P(t) = ES(t)\cdot k_{cat}$$

$$\frac{d}{dt}S(t) = -E(t)\cdot S(t)\cdot k_f + ES(t)\cdot k_r$$

$$\Rightarrow E(t) + ES(t) = 1.0$$

$$\Rightarrow ES(t) + P(t) + S(t) = 15.0$$

libSigNetSim

- Analysis

$$\frac{d}{dt}ES(t) = k_f \cdot E(t) \cdot S(t) - ES(t) \cdot k_r - ES(t) \cdot k_{cat}$$

$$\frac{d}{dt}P(t) = ES(t) \cdot k_{cat}$$

$$E(t) = -ES(t) + 1.0$$

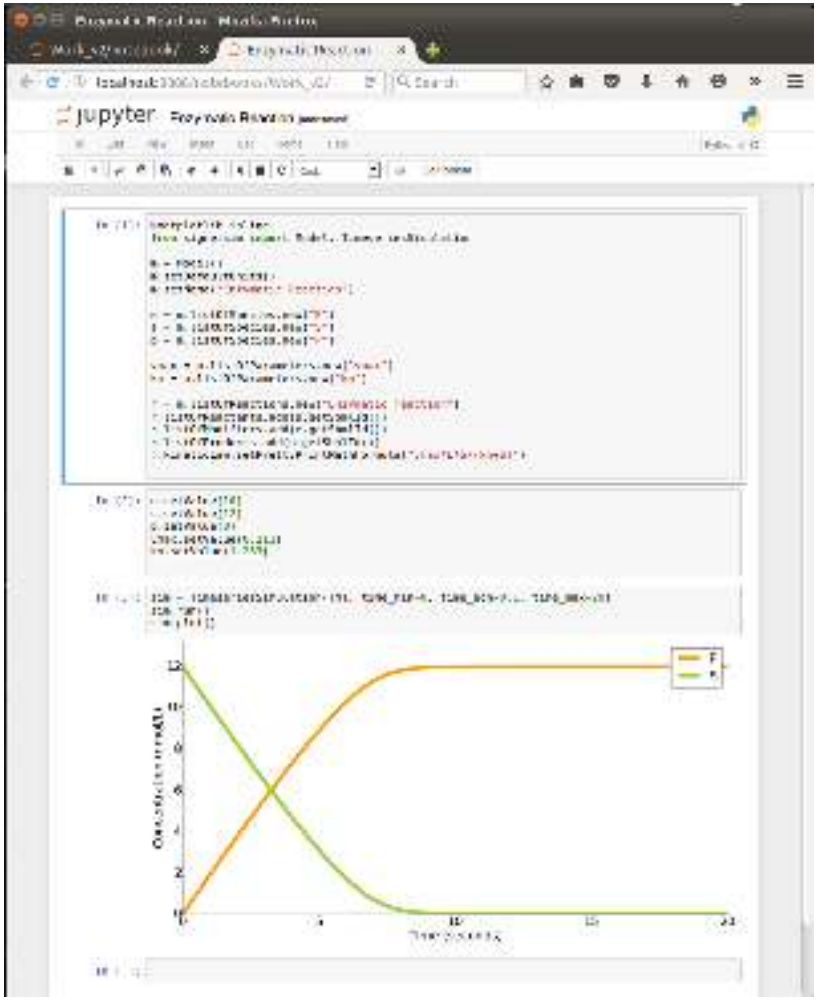
$$S(t) = -ES(t) - P(t) + 15.0$$

$\Rightarrow$Cheaper to integrate, enable to use continuation techniques

$\Rightarrow$All done automatically with symbolic math

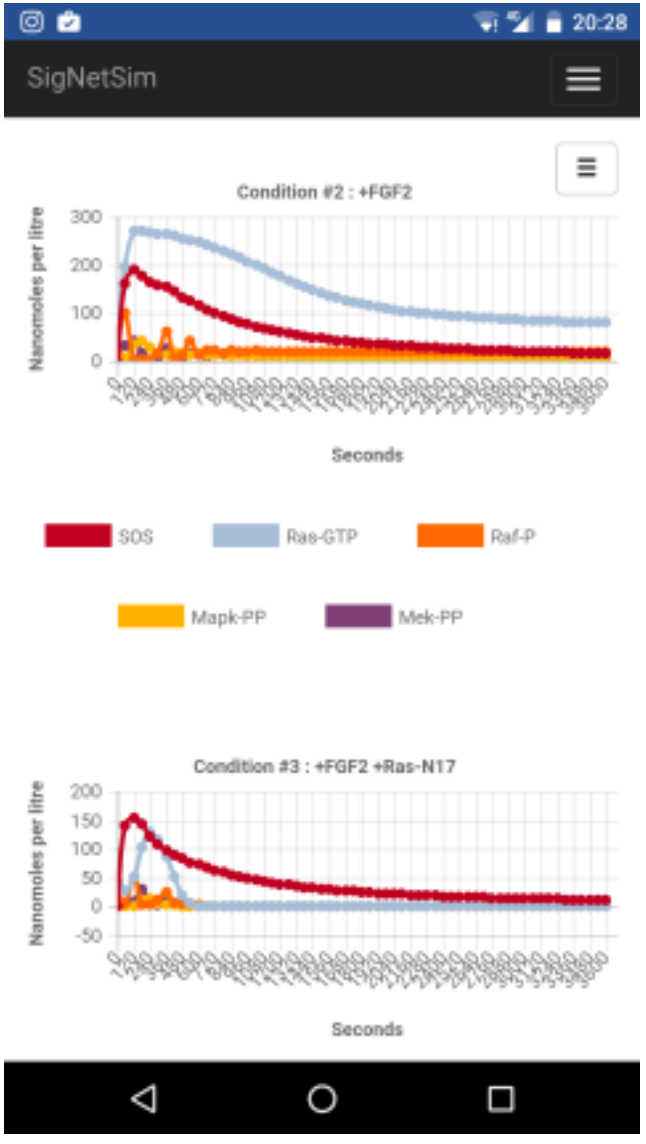
libSigNetSim

- Jupyter
  - Very basic compatibility with Jupyter
  - Mainly to be able to play directly with the symbolic math model
  - Still work in progress



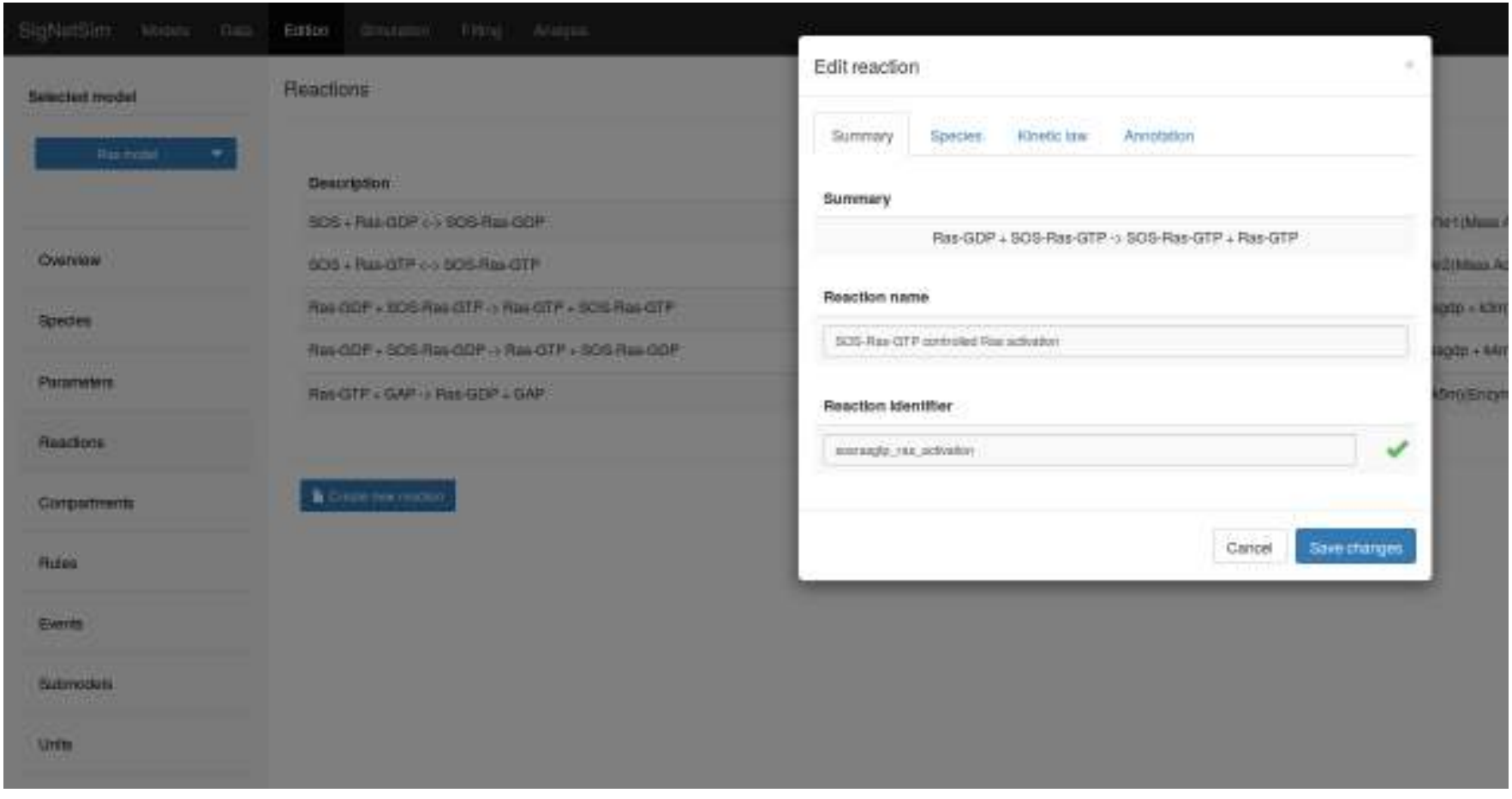
SigNetSim

- Introduction
  - Using Django framework
  - Bootstrap as a front-end
  - Chart.js, MathJax, MPLD3, ...
  - Compatible



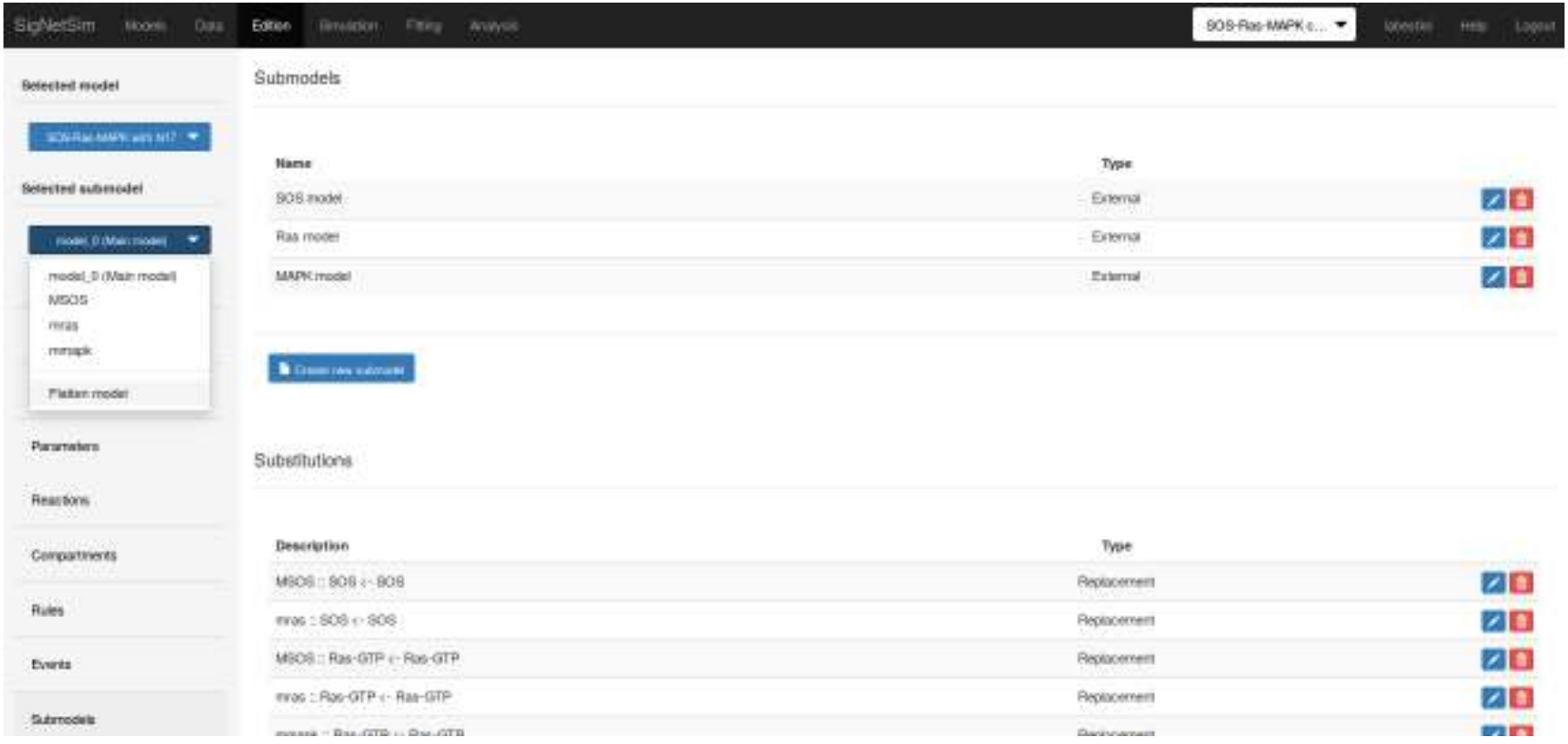
SigNetSim

› Editing



SigNetSim

- Editing



SigNetSim

› Database

Very basic DB to store experimental data

- › Experiment : A list of conditions
- › Condition : A list of treatments associated to a list of observations

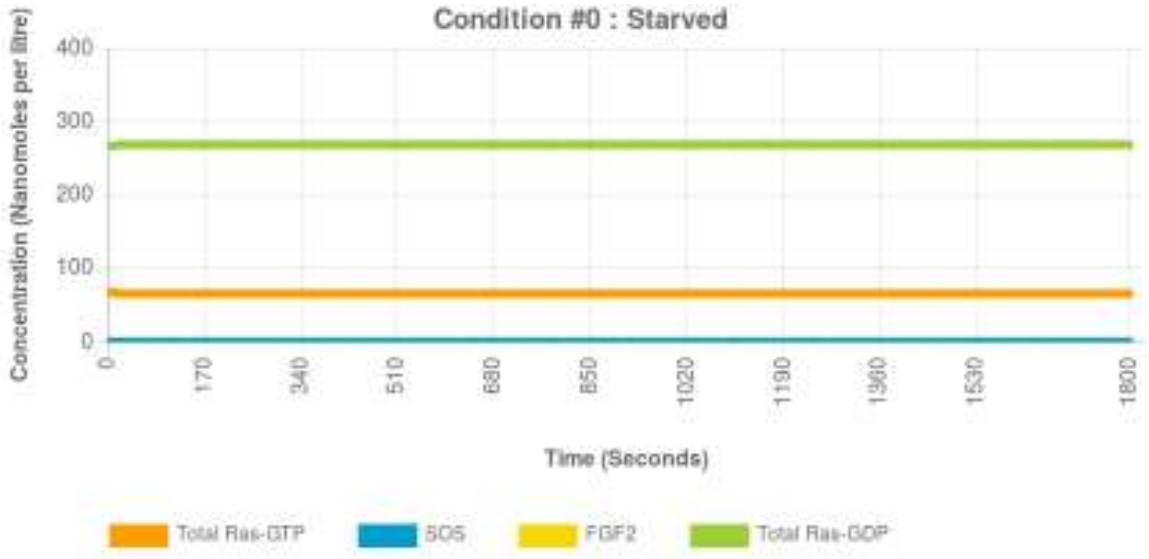
⇒Treatments can be used to do multi-conditions simulations

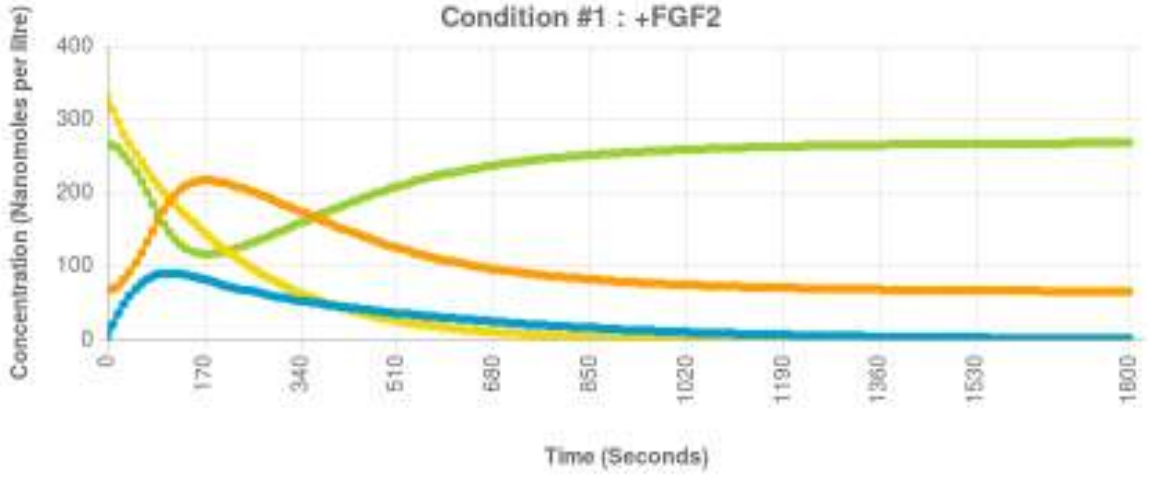
⇒Treatments and observations are used for model fitting



SigNetSim

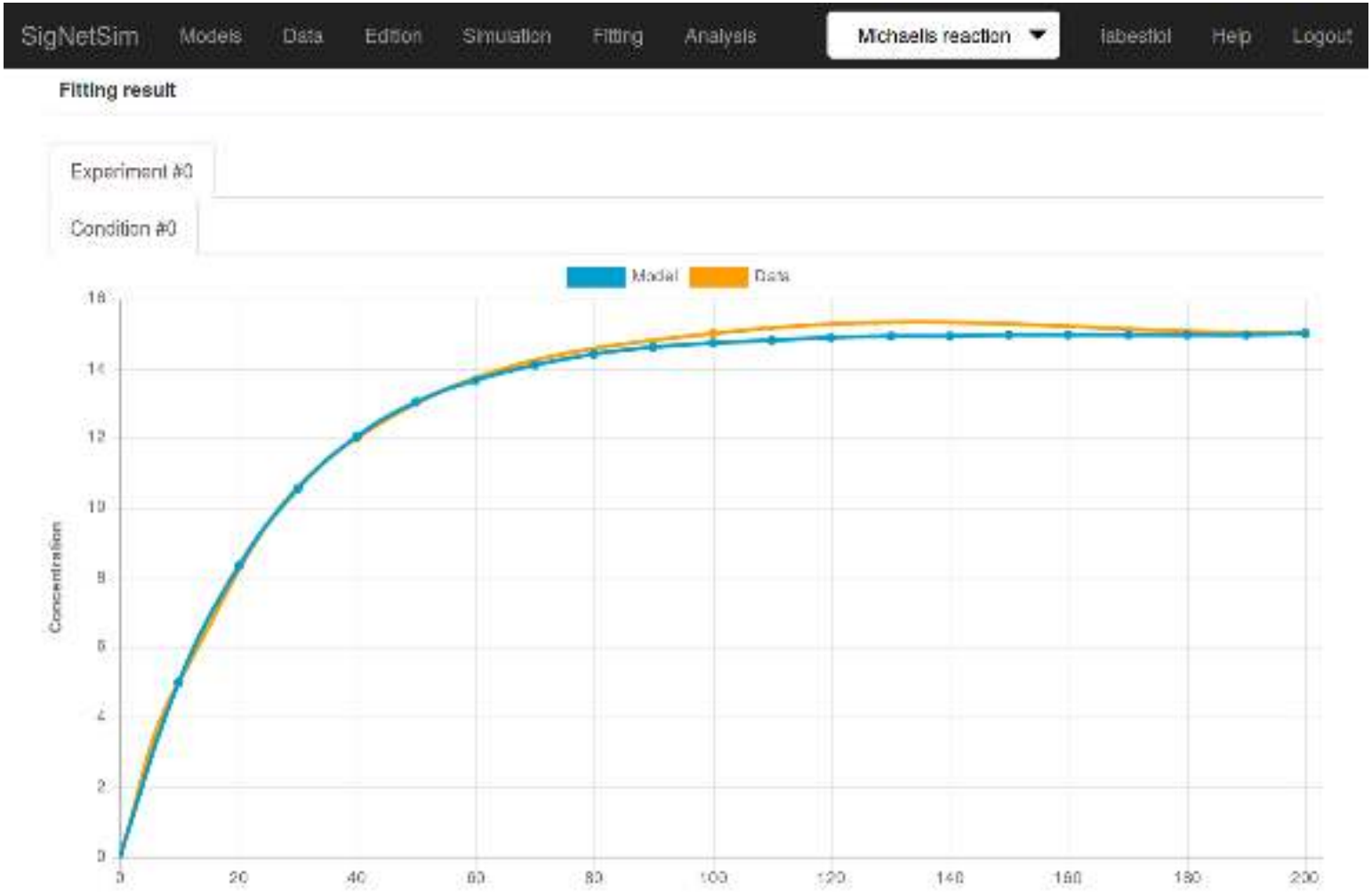
› Simulating





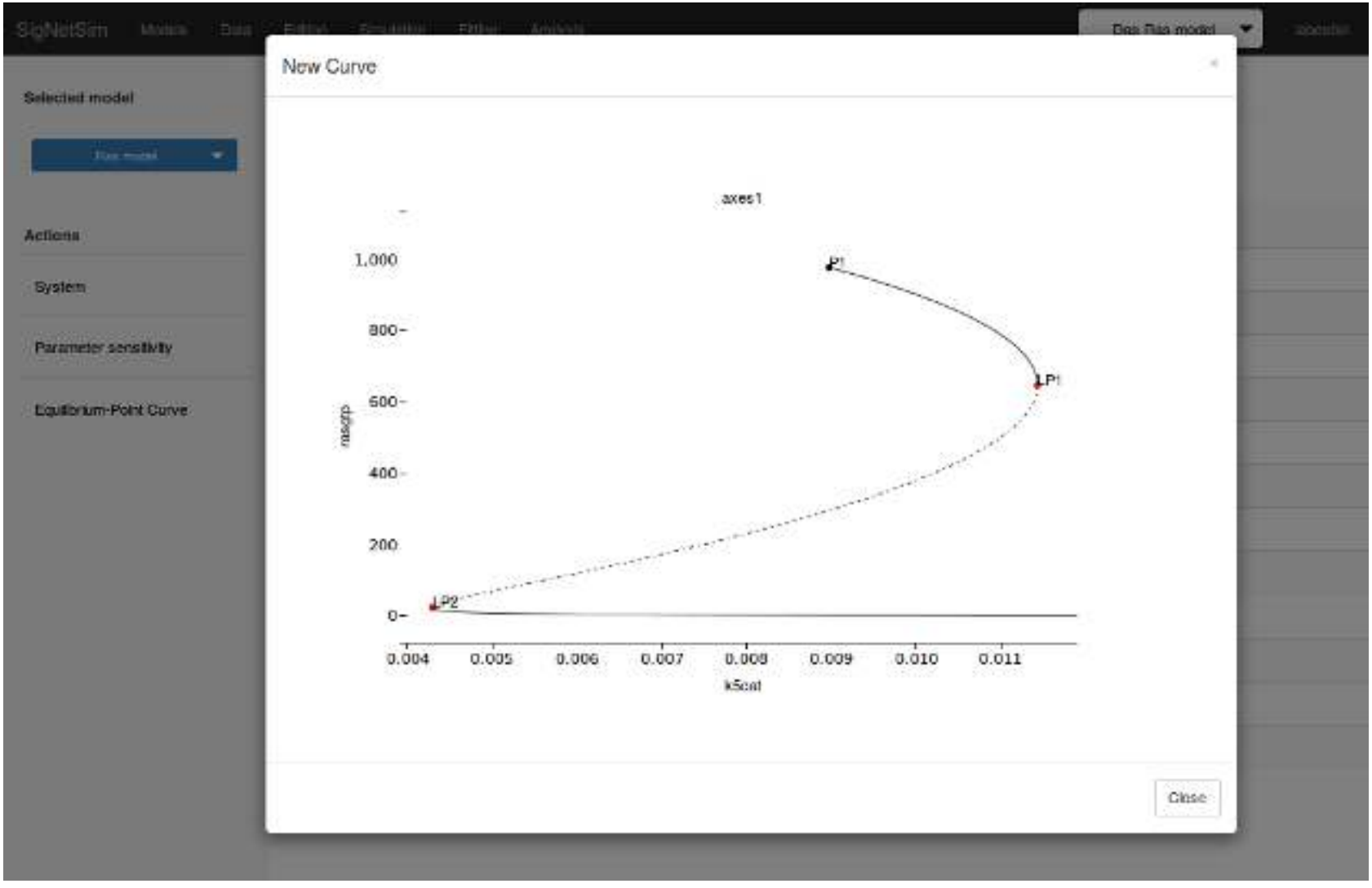
SigNetSim

› Fitting



SigNetSim

› Analysis



Conclusion

- Getting ready
  - "Core" nearly done
  - Soon to have a first version
  - The library and the web interface will be soon on GitHub
  - Paper should be submitted in the next months

Conclusion

- Future work
  - Performance improvements
  - Annotations
  - SED-ML
  - Simple SBGN vizualization (SBGNViz ?)
  - Distribution, Multi, Math ?

Conclusion

- Funding





Conclusion

- Live test version

cetics.butantan.gov.br/signetsim