MaBoSS: SBML-qual compatibility and a web interface for performing simulations

Vincent Noël[1]

[1]Computational Systems Biology of Cancer,
Institut Curie, INSERM U900, Mines ParisTech, PSL Research University, Paris

Toward a common framework for annotated, accessible, reproducible and interoperable computational models in biology, [BC]^2 2021

September 13, 2021









MaBoSS

› **Ma**rkovian **Bo**olean **S**tochastic **S**imulator



https://maboss.curie.fr/

- Boolean
- Model asymptotic and transient behavior
- Physical time
- Handle different time scale processes (transcription, phosphorylation, etc.)

⇒ Fills the gap between ODE and Boolean modeling

MaBoSS

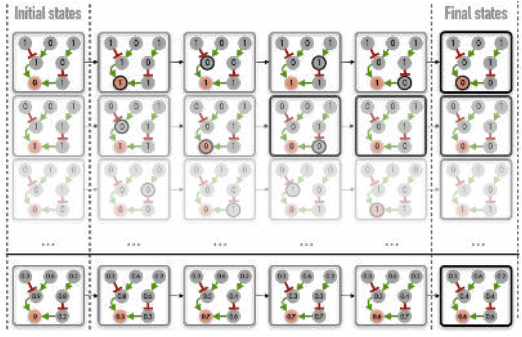
› **Ma**rkovian **Bo**olean **S**tochastic **S**imulator

› Simulate transitions from a boolean network state to another using a markov chain

› Transition rate :

$$p(S \to S') = \begin{cases} R_{up}(S), & \text{if} S_i = 0 \\ R_{down}(S), & \text{if} S_i = 1 \end{cases}$$

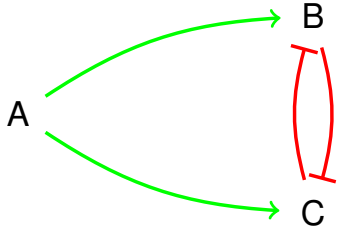
› These transitions are stochastic : multiple trajectories are possible



⇒ From a set of simulated trajectories, we compute distributions trajectories

MaBoSS

- Simple differentiation model




```
Node A
{
  logic = A;
  rate_up = @logic ? 1.0 : 0.0;
  rate_down = 0.0 ;
}

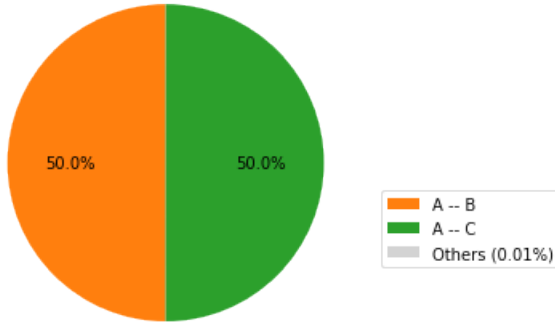
Node B
{
  logic = A & !C;
  rate_up = @logic ? 1.0 : 0.0;
  rate_down = 0.0;
}

Node C
{
  logic = A & !B;
  rate_up = @logic ? 1.0 : 0.0;
  rate_down = 0.0;
}
```

- A simple fork : A will activate either B or C
- A cells will either differentiate in A – B or A – C

MaBoSS

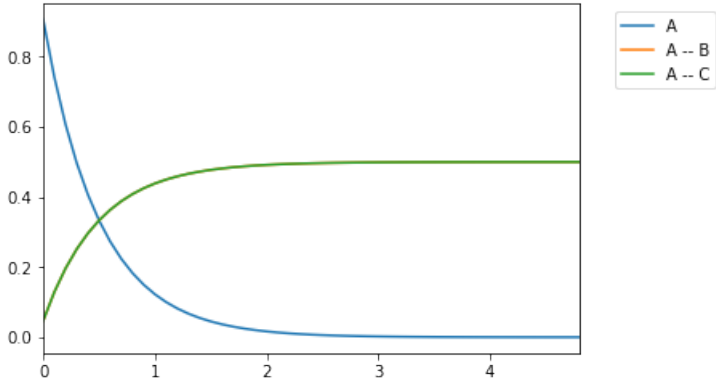
- Final state probability



- If both branches are equiprobable, then we end up with a split population

MaBoSS

- Probability trajectories



- We can look at state probability trajectories to observe transient behaviour

MaBoSS

› BND and CFG files

```
node A {
    logic = 1;
    rate_up = @logic ? 1 : 0;
    rate_down = 0;
}

node B {
    logic = A;
    rate_up = @logic ? $act_B : 0;
    rate_down = 0;
}

node C {
    logic = !A & !B;
    rate_up = @logic ? $act_C : 0;
    rate_down = @logic ? 0 : $act_C;
}
```

```
A.istate = TRUE;
B.istate = FALSE;
C.istate = TRUE;

A.is_internal = TRUE;

$act_B = 1;
$act_C = 0.1;

max_time = 50;
sample_count = 10000;
use_physrandgen = 1;
thread_count = 1;
```

Using MaBoSS : pyMaBoSS

- Initially developped by Nicolas Levy
- Maintained by Aurelien Naldi, Loic Pauleve, me
  https://github.com/colomoto/pyMaBoSS
- Available on Pypi:
  $ pip install maboss
- Available on Conda:
  $ conda install -c colomoto pymaboss


Nicolas Levy

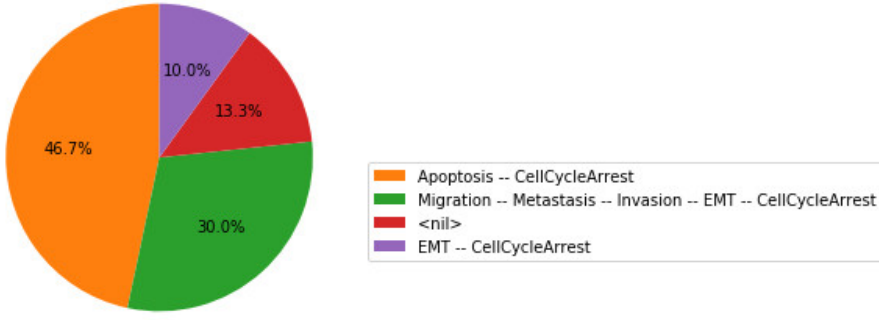

Aurelien Naldi


Loïc Paulevé


Vincent Noël

Using MaBoSS : pyMaBoSS

```
import maboss
sim = maboss.load("metastasis.bnd", "metastasis.cfg")
res = sim.run()
res.plot_piechart()
```



› A few lines of code to simulate the model and plot the final states distribution

MaBoSS SBML-qual compatibility

- Using BioLQM
  - SBML-qual models can be converted to MaBoSS models via GINsim
  - SBML-qual models can also be converted via BioLQM python bindings

```
biolqm_model = biolqm.load("Cohen.sbml")
maboss_model = biolqm.to_maboss(biolqm_model)
```

⇒ We wanted to have native compatibility with SBML-qual

MaBoSS SBML-qual compatibility

- Native SBML-qual compatibility
  - Run MaBoSS Simulation using classic model files

```
MaBoSS [-c metastasis.cfg] -o res metastasis.bnd
```

  - Run MaBoSS simulation using SBML-qual model file

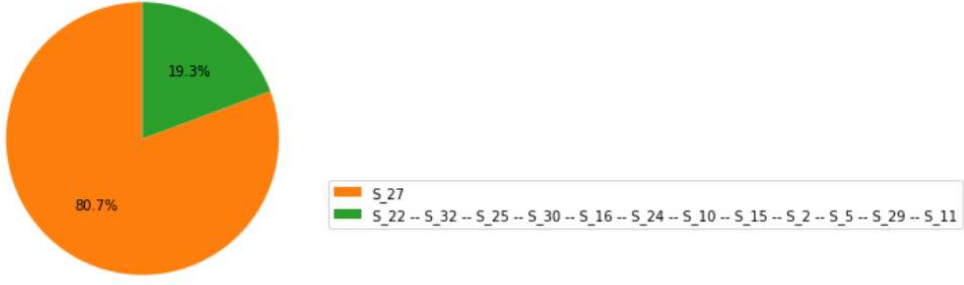
```
MaBoSS [-use-sbml-names] [-c metastasis.cfg] -o res Cohen.sbml
```

  - Just replace the bnd file with a SBML
  - CFG file needs to be compatible
  - Also work with BNet files

MaBoSS SBML-qual compatibility

› Loading SBML-qual from pyMaBoSS

```
maboss_model = maboss.loadSBML("cohen.sbml")
res = maboss_model.run()
res.plot_piechart()
```



MaBoSS SBML-qual compatibility

› Use SBML names

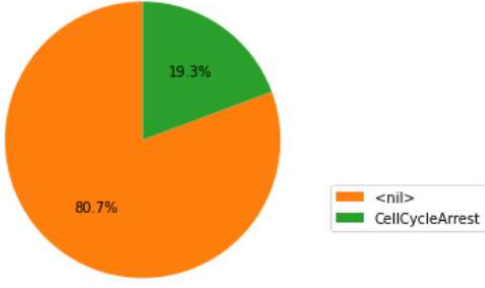
```
maboss_model = maboss.loadSBML("cohen.sbml", use_sbml_names=True)
res = maboss_model.run()
res.plot_piechart()
```



MaBoSS SBML-qual compatibility

› Optional CFG file

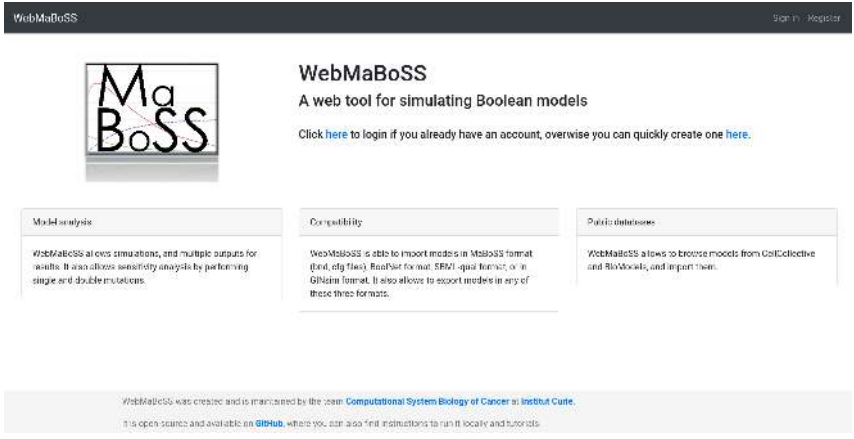
```
maboss_model = maboss.loadSBML("cohen.sbml", "cohen_names.cfg", use_sbml_names=True)
res = maboss_model.run()
res.plot_piechart()
```



MaBoSS SBML-qual compatibility

- Perspectives
  - Available on Conda & PyPi, using MaBoSS 2.4.0 and pyMaBoSS 0.8.0
  - Not fully compatible with multi-valued boolean models
  - Development of proprietary tags to store MaBoSS specific fields ?
  - Please test !

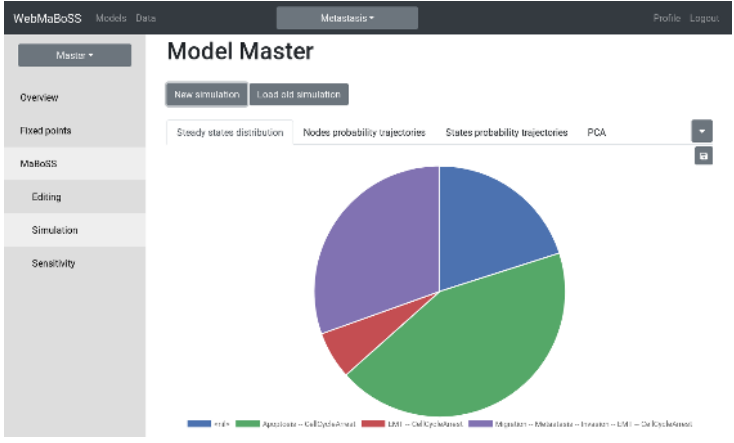
WebMaBoSS



https://maboss.curie.fr/webmaboss/

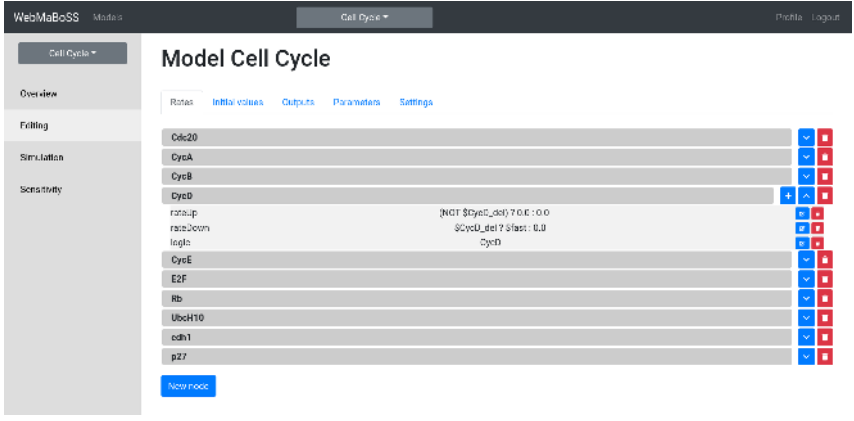
WebMaBoSS

- Simple MaBoSS simulations



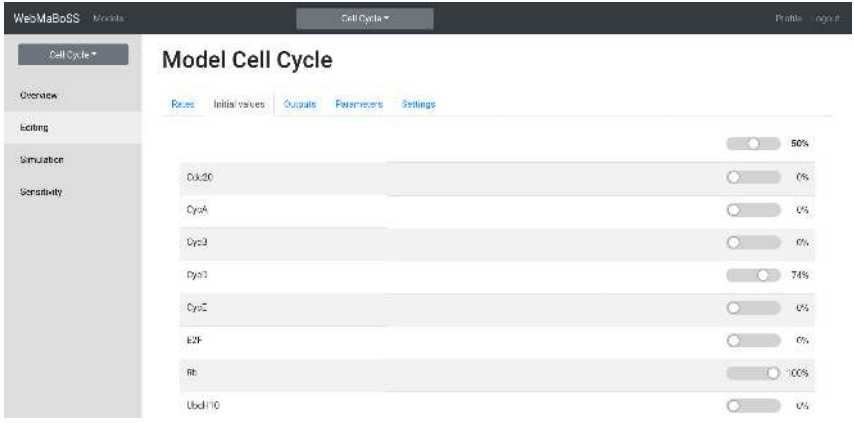
WebMaBoSS

› Editing the model



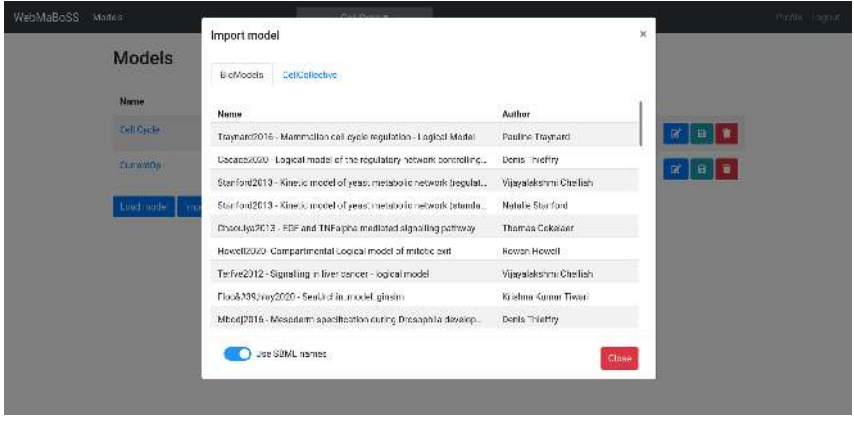
WebMaBoSS

› Editing the model



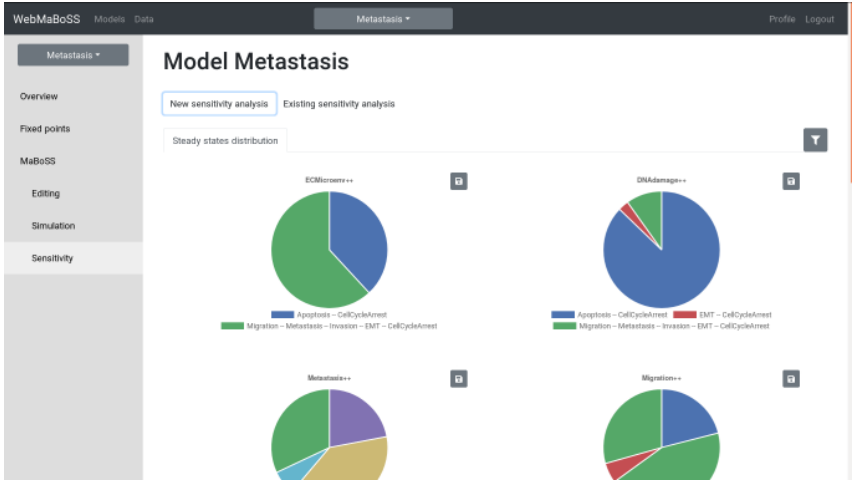
WebMaBoSS

› Import SBML-qual from databases



WebMaBoSS

- Sensitivity analysis



WebMaBoSS

› Submitted !



1

WebMaBoSS: A web interface for simulating Boolean models stochastically

Vincent Noël [1,2,3,*], **Marco Ruscone** [1,2,3], **Gautier Stoll** [4], **Eric Viara** [5], **Andrei Zinovyev**[1,2,3], **Emmanuel Barillot**[1,2,3] **and Laurence Calzone**[1,2,3]

[1] *Institut Curie, PSL Research University, Paris, France*
[2] *INSERM, U900, Paris, France*
[3] *MINES ParisTech, PSL Research University, CBIO-Centre for Computational Biology, Paris, France*
[4] *Equipe 11 labellisée par la Ligue Nationale contre le Cancer, Centre de Recherche des Cordeliers, INSERM U1138, Universite de Paris, Sorbonne Universite, Paris, France*
[5] *Sysra, Yerres, France*

WebMaBoSS

- Perspectives
  - Support for SBML layout
  - New automatic pipelines
  - Annotation editor in WebMaBoSS ?
  - Interfacing to HPC to support large computations ?

Aknowledgments

MaBoSS early team


Gautier Stoll


Eric Viara


Laurence Calzone


Emmanuel Barillot

MaBoSS ecosystem team


Aurelien Naldi


Denis Thieffry


Loïc Paulevé


Nicolas Levy


Stéphanie Chevalier


Arnau Montagud


Vincent Noël


Gaëlle Letort


Marco Ruscone


Andrei Zinovyev

https://maboss.curie.fr
vincent.noel@curie.fr