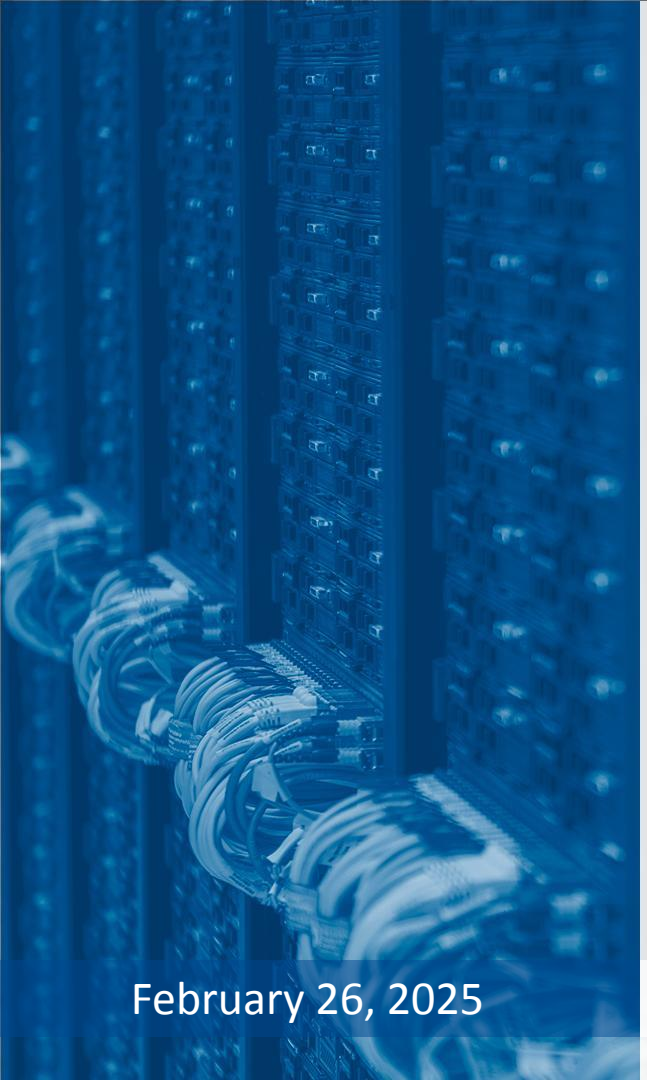


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Computational Biology Across Scales: From Pathway Modelling Tools to Cell-Level Simulations

Marco Ruscone, PhD (BSC)

February 26, 2025

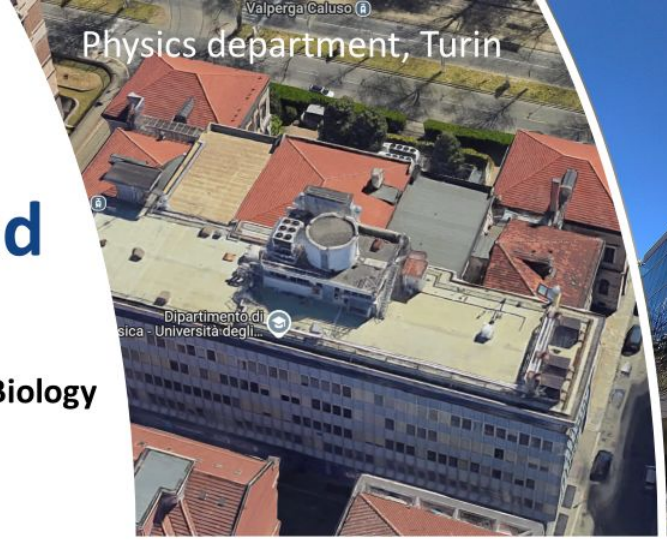
Osaka University, Japan

Academical Background

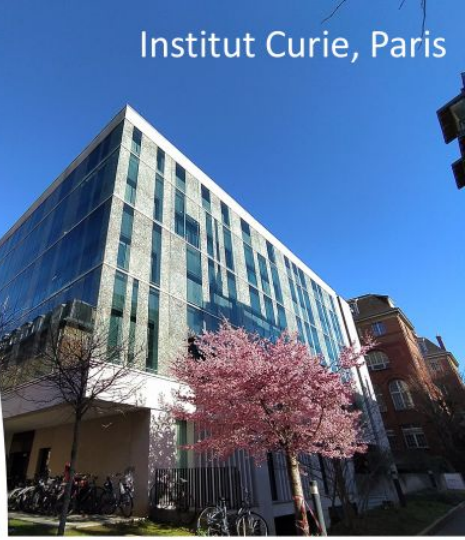
Master degree in Physics of Complex System for Biology
- Pr. Michele Caselle

PhD in Computational Biology
- Dr. Laurence Calzone
- Dr. Vincent Noël
- Pr. Emmanuell Barillot

Postdoctoral Researcher
- Pr. Alfonso Valencia



Physics department, Turin


Institut Curie, Paris

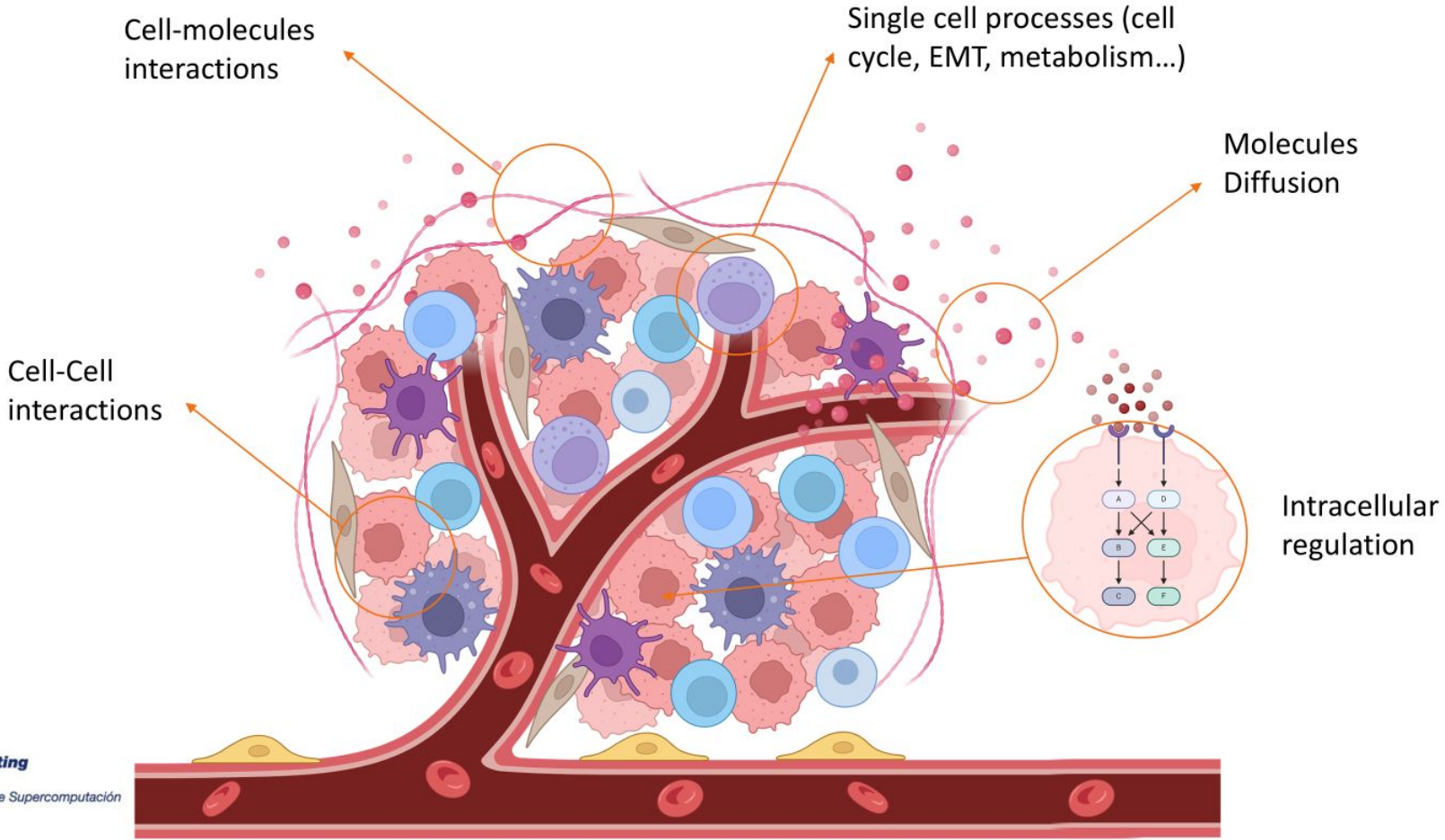
BSC, Barcelona

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

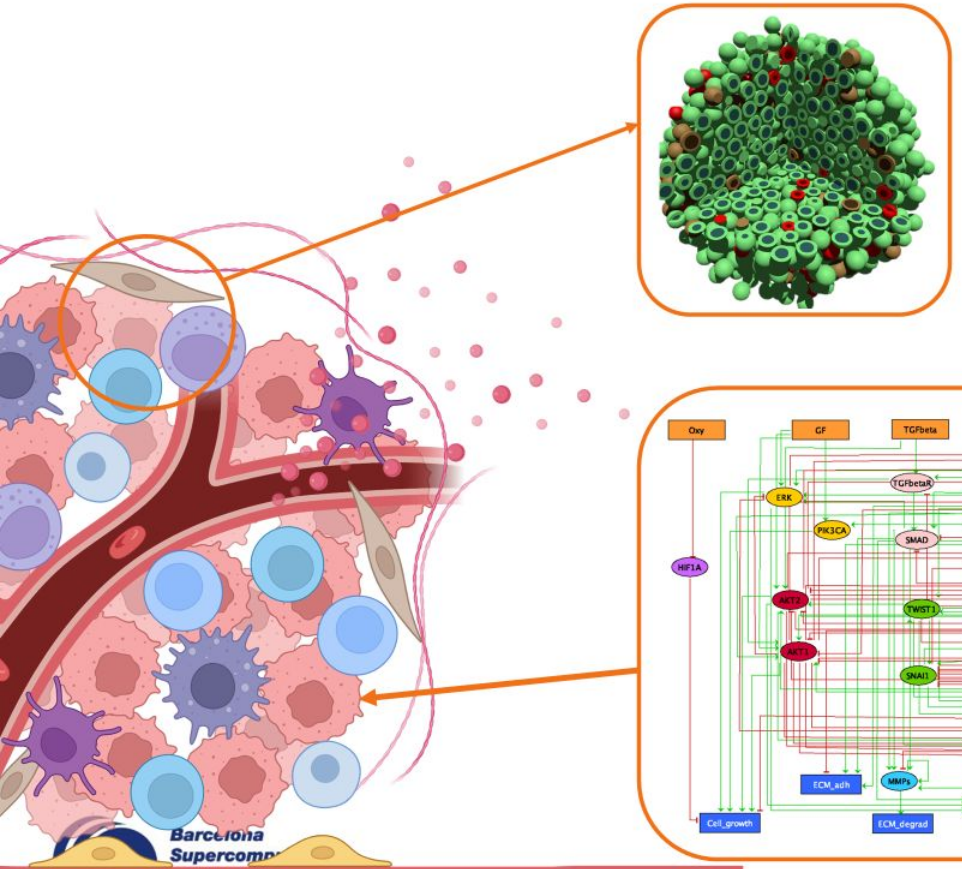
Outline

- Introduction
- Part 1 - Neko
  - Hands On - From static network to Boolean model (almost…)
- Part 2 - MaBoSS
  - Hands On - simulating the Boolean model
- Part 3 - PhysiCell/PhysiBoSS
  - Hands On - Simulating cells and their Boolean model

Multiscale processes in biology

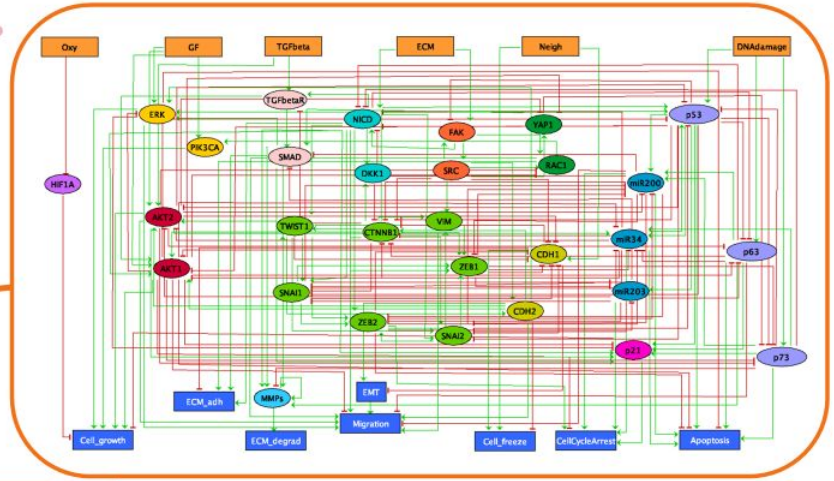


How to go multiscale



Simulating cells physical properties and molecules diffusion with an **agent-based** approach

Feedback



Simulating pathways activity using an **intracellular model**: ODEs, SDEs, Boolean…

Where do we start?

The recipe for a good multiscale model:

A Cell Cycle model!

1. A good **biological question** → what do you want/need to simulate?
2. A nice **intracellular model** → to simulate pathways activity

Cyclins activity and Apoptotic switch!

3. An **agent-based** modeling framework → to simulate cell's physical properties and phenotypes

Modeling a cell population growing and try different mutants condition!

Objective of today:
from a static network to cell population

Explore new scenarios

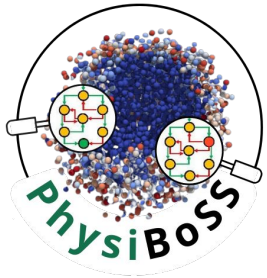
01 Create Static Network

02 Convert to Boolean Model

03 Validate Simulation

04 Integrate with PhysiBoSS

05 Start In Silico Experiments



Before we start

1. Make sure to have anaconda and git!

2. Clone the following repo: https://github.com/marcorusc/HandsOn_PRIMe.git

3. Once cloned, from the cloned folder, create the environment with conda using:
   conda create --name hands_on --file hands_on-env.txt

Conda mini-cheatsheet:

1. Get Anaconda distribution here: https://www.anaconda.com/download
2. Activate environment:
   conda activate <env_name>
3. List packages in the env:
   conda list
4. You can install packages in your env by pip or conda:
   pip install <package_name>
   conda install <package_name>

Part 1 - Introducing NeKo

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

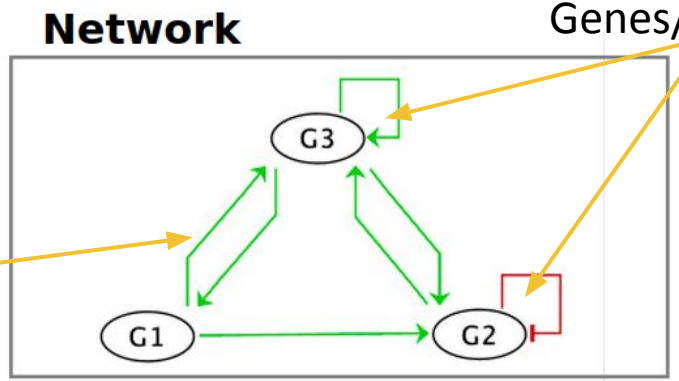
What is a Boolean model?

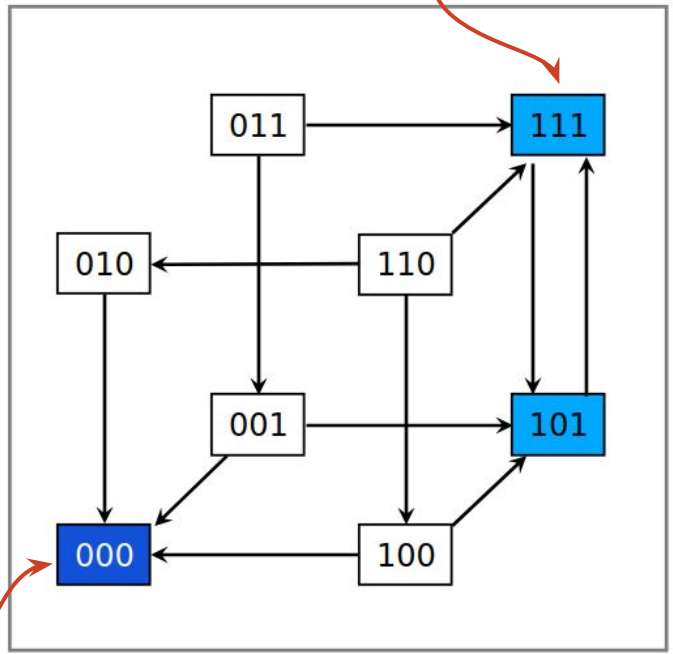
A simpler alternative to ODEs and SDEs

Interactions

Genes/Proteins

Cyclic attractor





Attractor

Courtesy of Dr. Saran Pankaew - Institut Curie

How do we build a Boolean model?

Two approaches:

1. **Data**

Carnival, Signaling profiler, Augusta...

2. **No Data**

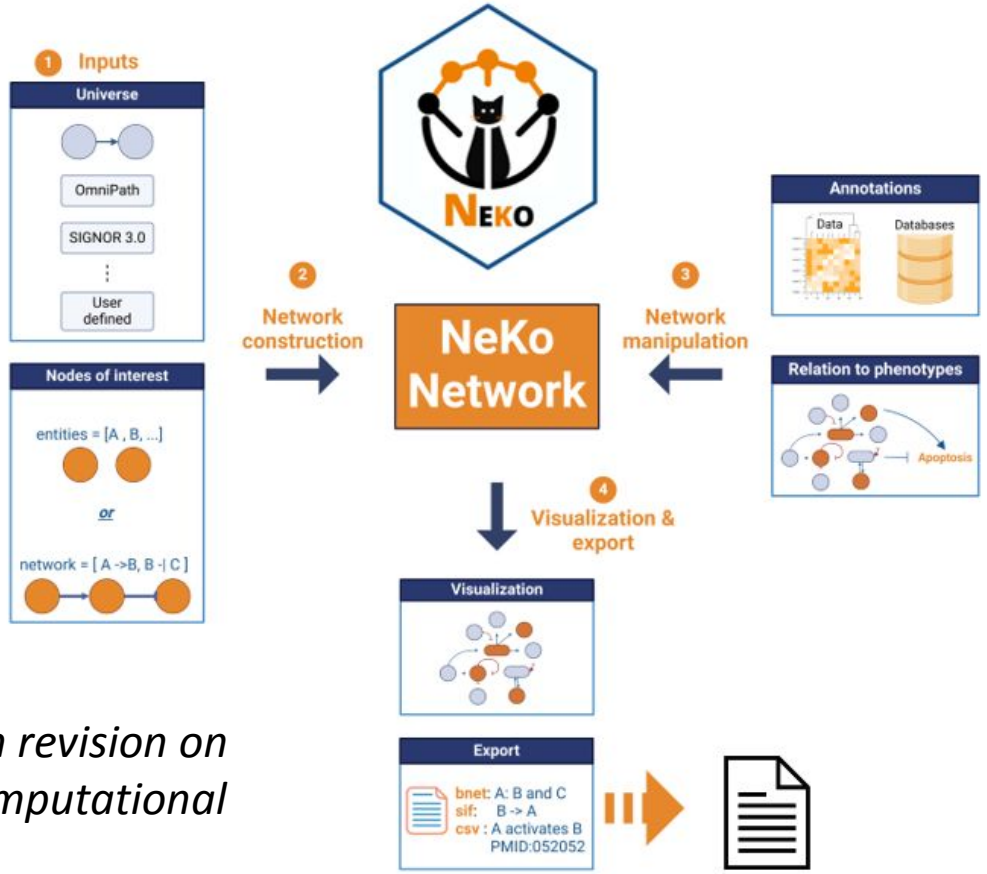
HuRI

1. Reading LOTS of publications
2. Exploring many databases AND reading LOTS of publication

It can take up to 6 months (or even more...)

Introducing NeKo

An extensible and adaptable python package to create static networks and Boolean models from prior-knowledge



Paper in revision on Plos Computational Biology

NeKo, what do you need?

1) A research focus, ex: cell invasion, metabolism, cell cycle...
2) A set of genes of interest (**SEEDS**)
3) One or more databases to interrogate: Omnipath, Signor, Huri, Kegg, Wikipathway....

Universe

Pandas Dataframe:
Source, Target, Effect

~10.000 / 100.000 interactions

NeKo_network(SEEDS , *Universe*)

Choose one or more **STRATEGIES**

We are going to discuss them during the hands on

NeKo Hands On

- Installation (if not done already)
- Opening Jupyter Notebook and importing NeKo
- Create our first Network object
- Populate the Network
- Visualize the Network
- Changing database
- Exporting the Network

How to install

1. Make sure to have your environment activated
2. pip install nekomata (the name neko was already taken…)
3. install dependencies if necessary (it should not)
4. If issues check: https://github.com/sysbio-curie/Neko or ask me!

Hands on:

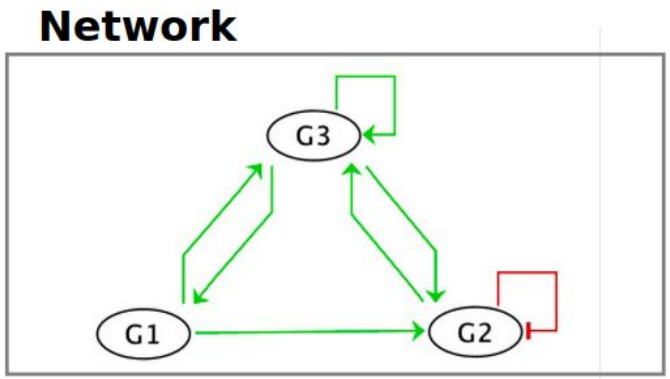
- Navigate in the folder NeKo with: *cd NeKo* (on Linux)
- Open a Jupyter Notebook: *jupyter notebook*
- **Hands On!**

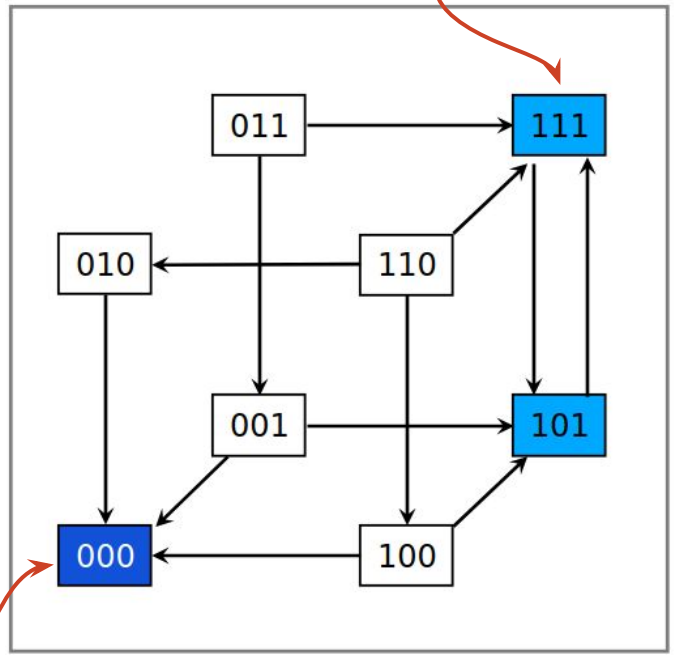
Question: Can we create a working Boolean model using just NeKo?

Part 2 - Introducing MaBoSS

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Refresh: Boolean model

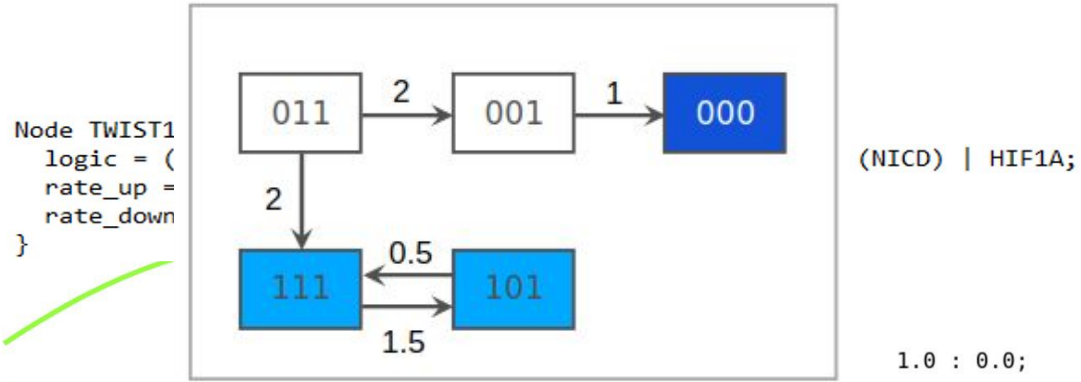




Courtesy of Dr. Saran Pankaew - Institut Curie

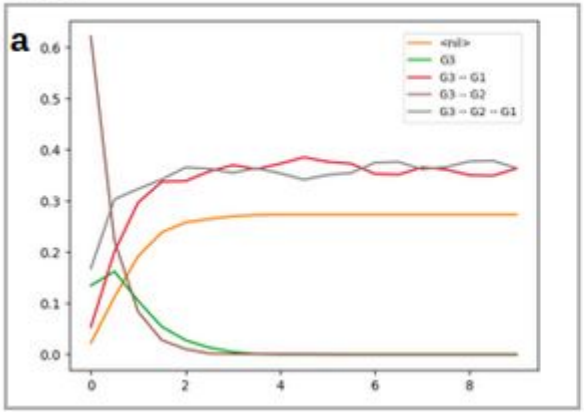
Markovian Boolean Stochastic Simulator

- C++ software with Python binding
- Simulates continuous time Markov process on BN
- Stochastic transition between nodes state through logical equation
- Leverages the Gillespie algorithm

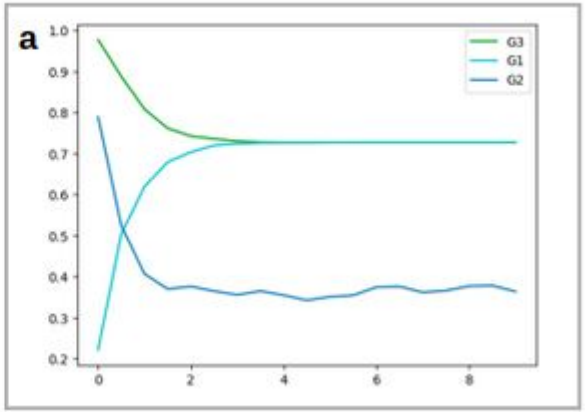


A

State



Node



MaBoSS, what do you need?

- A .bnd file containing your network model
- A .cfg file containing your network configuration (can be modified using Python)
- Possible model repositories to try: https://www.cellcollective.org/, https://www.ebi.ac.uk/biomodels/, http://ginsim.org/models_repository
- There is also an online version of MaBoSS → WebMaBoSS ! https://webmaboss.vincent-noel.fr/

MaBoSS Hands On

- Installation (if not done already)
- Opening Jupyter Notebook and importing MaBoSS
- Loading our first MaBoSS model
- Simulate the model
- Visualize the results
- Try some mutations

How to install

1. Make sure to have your environment activated

2. *conda install -c colomoto maboss*

3. You can always use WebMaBoSS if it does not work on local: https://webmaboss.vincent-noel.fr/

Hands On:

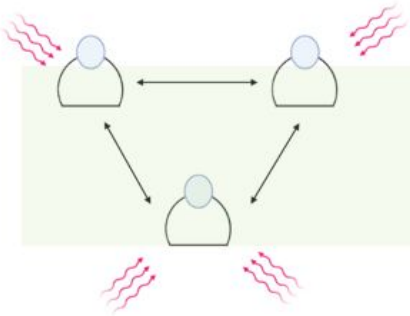
- Navigate in the folder MaBoSS with: *cd MaBoSS* (on Linux)
- Open a Jupyter Notebook: *jupyter notebook*
- **Hands On!**

Don’t miss the seminar by Dr. Laurence Calzone (Institut Curie, Paris) on March !

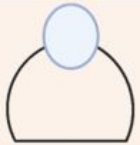
Part 2 - Introducing the PhysiCell ecosystem

Multi Agent-based Systems

Agent-oriented Programming

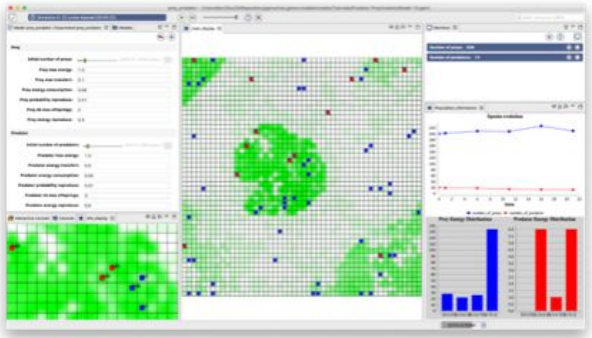


- **Autonomous**: makes independent decisions without any human intervention

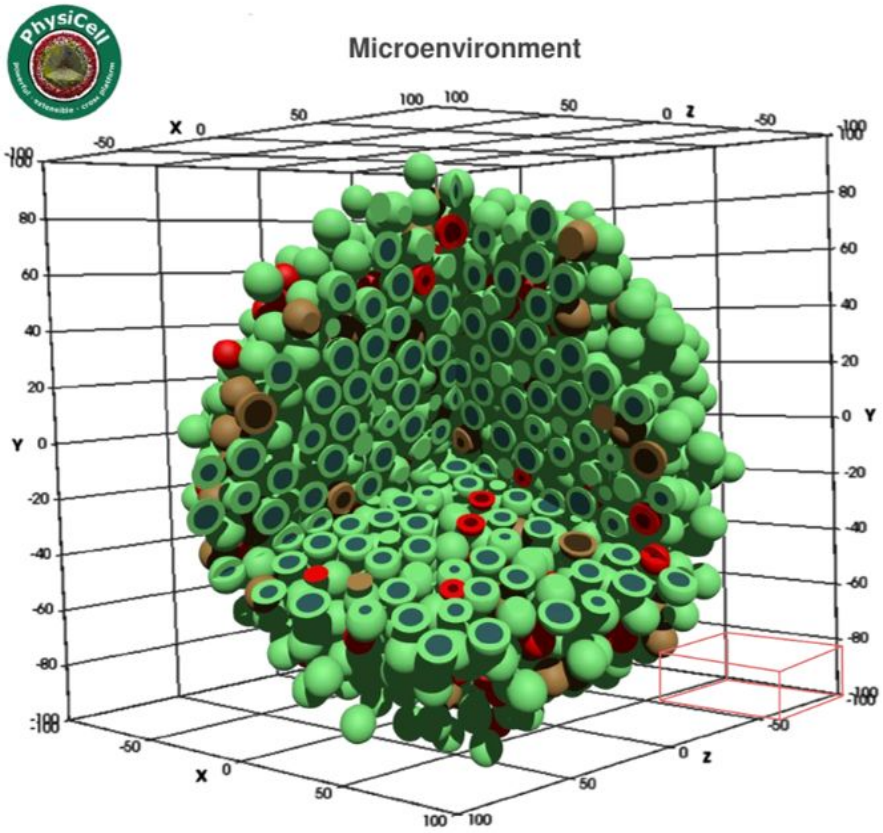


- **Social**: communication with other agents
- **Reactive**: perceives the environment and can modify its surrounding

Capture non-intuitive collective behaviors from simple agent-agent interactions



Introducing PhysiCell



Diffusion equations

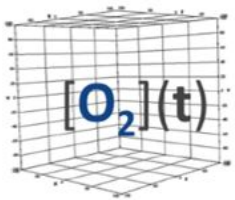
$$\frac{\partial \boldsymbol{\rho}}{\partial t} = \overbrace{\mathbf{D}\nabla^2\boldsymbol{\rho}}^{\text{diffusion}} - \overbrace{\boldsymbol{\lambda}\boldsymbol{\rho}}^{\text{decay}} + \overbrace{\mathbf{S}(\boldsymbol{\rho}^* - \boldsymbol{\rho})}^{\text{bulk source}} - \overbrace{\mathbf{U}\boldsymbol{\rho}}^{\text{bulk uptake}}$$

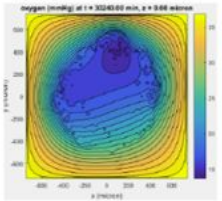
$$+ \overbrace{\sum_{\text{cells } k} \delta(\mathbf{x} - \mathbf{x}_k) W_k [\mathbf{S}_k(\boldsymbol{\rho}_k^* - \boldsymbol{\rho}) - \mathbf{U}_k \boldsymbol{\rho}]}^{\text{sources and uptake by cells}} \text{ in } \Omega$$

System of PDEs for each molecule:

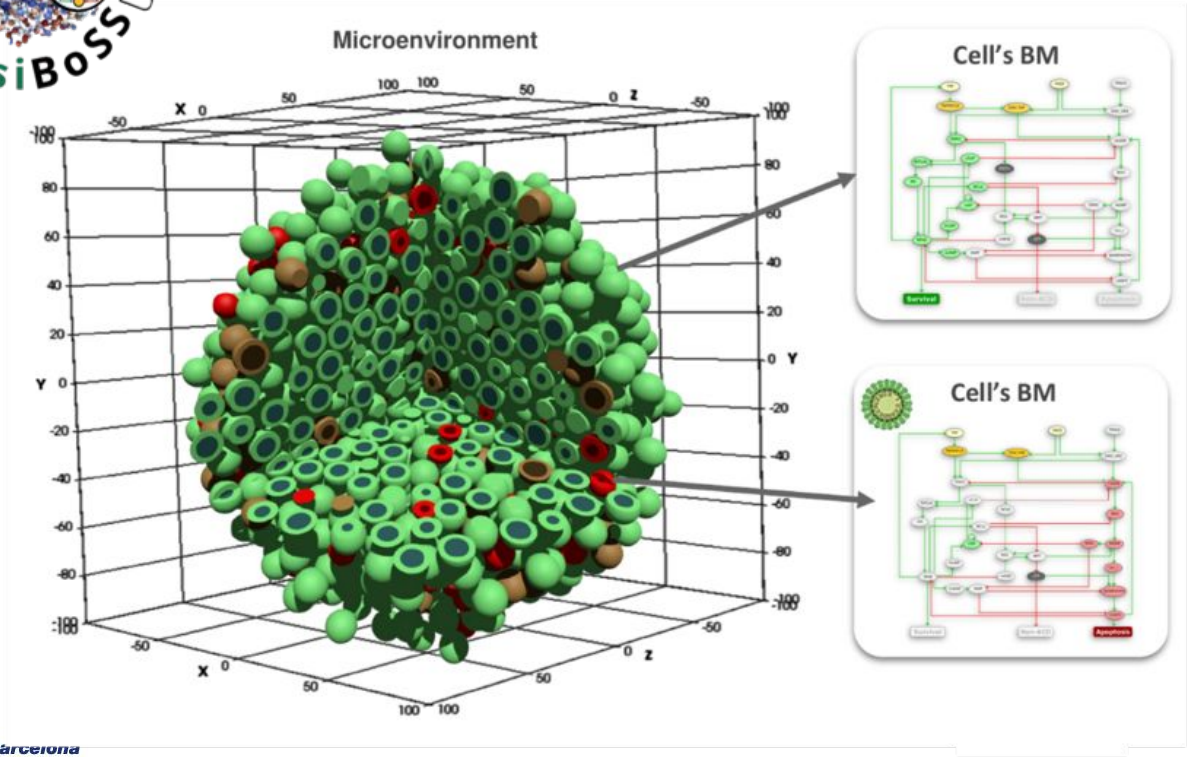
- Diffusion term
- Decay
- Uptake/Production

PDEs are solved using BioFVM





Introducing PhysiBoSS



MaBoSS

Markovian **Bo**olean
Stochastic **S**imulator

$$\rho_{(\mathbf{S}\to\mathbf{S}')} = R_i^{\text{up}}(\mathbf{S}) \text{ if } S_i = 0$$
$$\rho_{(\mathbf{S}\to\mathbf{S}')} = R_i^{\text{down}}(\mathbf{S}) \text{ if } S_i = 1$$

PhysiBoSS, what do you need?

- A .bnd file containing your network model
- A .cfg file containing your network configuration (can be modified using Python)
- The official GUI of PhysiCell → PhysiCell Studio installed (next slide)
- Nothing else! PhysiCell provides us with a list of possible **signals** and behaviors!

PhysiBoSS Hands On

- Installation (if not done already)
- Opening the PhysiCell GUI
- Creating a simple model of cell growth controlled by a Boolean model
- Simulate the model
- Visualize the results
- Try some mutations

How to install

1. Clone PhysiCell on your machine:
   git clone https://github.com/MathCancer/PhysiCell.git

2. Clone the PhysiCell-Studio on your machine in a separate folder:
   git clone https://github.com/PhysiCell-Tools/PhysiCell-Studio.git
   download the requirements in you env:
   pip install -r requirements.txt
3. Navigate in your PhysiCell folder and compile the template project:
   cd ~/your/path/to/PhysiCell
   make template_BM
   make -j
4. Now open the studio from the PhysiCell folder:
   python ~/path/to/PhysiCell-Studio/bin/studio.py -e project -c config/PhysiCell_settings.xml

Hands On:

- Interesting Boolean models can be found in the **MaBoSS** folder of the **Hands On material** → Sizek Cell Cycle model and Corral Differentiation model

- Ready-to-use config file for PhysiCell are available in the **PhysiCell** folder of the **Hands On material** → Cell Cycle model mutations, Corral Differentiation model mutations

- **Hands On!**

References

- Ruscone, Marco, Eirini Tsirvouli, Andrea Checcoli, Denes Turei, Emmanuel Barillot, Julio Saez-Rodriguez, Loredana Martignetti, Åsmund Flobak, and Laurence Calzone. “**NeKo: A Tool for Automatic Network Construction from Prior Knowledge.**” bioRxiv, October 15, 2024. https://doi.org/10.1101/2024.10.14.618311.
- Stoll, Gautier, Barthélémy Caron, Eric Viara, Aurélien Dugourd, Andrei Zinovyev, Aurélien Naldi, Guido Kroemer, Emmanuel Barillot, and Laurence Calzone. “**MaBoSS2.0: An Environment for Stochastic Boolean Modeling.**” Bioinformatics 33, no. 14 (July2017): 2226–28. https://doi.org/10.1093/bioinformatics/btx123.
- Ghaffarizadeh, A., Heiland, R., Friedman, S. H., Mumenthaler, S. M., & Macklin, P. (2018). “**PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems.”** In PLoS Computational Biology (Vol. 14, Issue 2).
- Letort, Gaelle, Arnau Montagud, Gautier Stoll, Randy Heiland, Emmanuel Barillot, Paul Macklin, Andrei Zinovyev, and Laurence Calzone. “**PhysiBoSS: A Multi-Scale Agent-Based Modelling Framework Integrating Physical Dimension and Cell Signalling.**” Bioinformatics, https://doi.org/10.1093/bioinformatics/bty766.
- Marco Ruscone, Andrea Checcoli, Randy Heiland, Emmanuel Barillot, Paul Macklin, Laurence Calzone, Vincent Noël, “**Building multiscale models with PhysiBoSS, an agent-based modeling tool**”, *Briefings in Bioinformatics*, Volume 25, Issue 6, November 2024, bbae509, https://doi.org/10.1093/bib/bbae509
- Heiland R, Bergman D, Lyons B, Waldow G, Cass J, Lima da Rocha H, Ruscone M, Noël V, Macklin P. “**PhysiCell Studio: a graphical tool to make agent-based modeling more accessible.**” GigaByte. 2024 Jun 19;2024:gigabyte128. doi: 10.46471/gigabyte.128. PMID: 38948511; PMCID: PMC11211762.

Acknowledgements

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Alfonso Valencia

Arnau Montagud

Adam Smelko

Miguel Ponce de Leon

Jose Carbonell Caballero

. . .

institutCurie

Computational Systems Biology of Cancer

Emmanuel Barillot

Laurence Calzone

Vincent Noël

Andrei Zinovyev

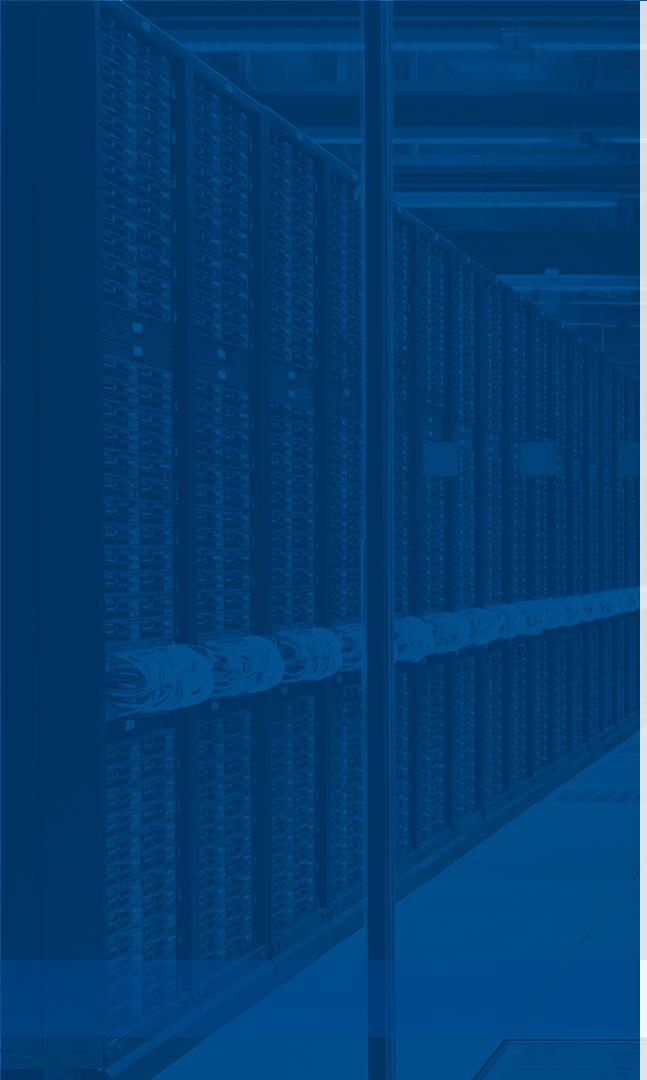
Gaëlle Letort

INDIANA UNIVERSITY

Paul Macklin

Randy Heiland

Barcelona Supercomputing Center
Centro Nacional de Supercomputación



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you for your participation!

Questions?

Marco Ruscone, PhD