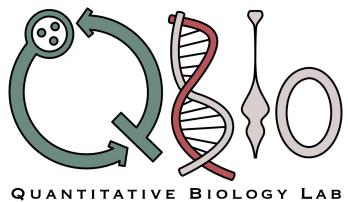




UNIVERSITÀ
DI TORINO | di.unito.it
DIPARTIMENTO
DI INFORMATICA



IEEE - International Conference on
Healthcare Informatics
Università della Calabria, Italy
June 18th-21st, 2025

Epidemic Modelling Frameworks for Studying Infectious Disease Dynamics

Daniele Baccega, Marco Beccuti, Simone Pernice, Irene Terrone

13TH IEEE International Conference on Healthcare Informatics
University of Calabria, Rende, Italy
18th-21st June, 2025

Outline

- Who are we?
- Overview of computational models in epidemiology (≈ 15 min).
- Macro-level models (≈ 1 hour).
 - Case study: [GreatMod](#).
- Micro-level models (≈ 1 hour).
 - Case study: [Forge4Flame](#).
- From macro to micro (≈ 15 min).
- Conclusion (≈ 5 min).

 SCAN
ME!

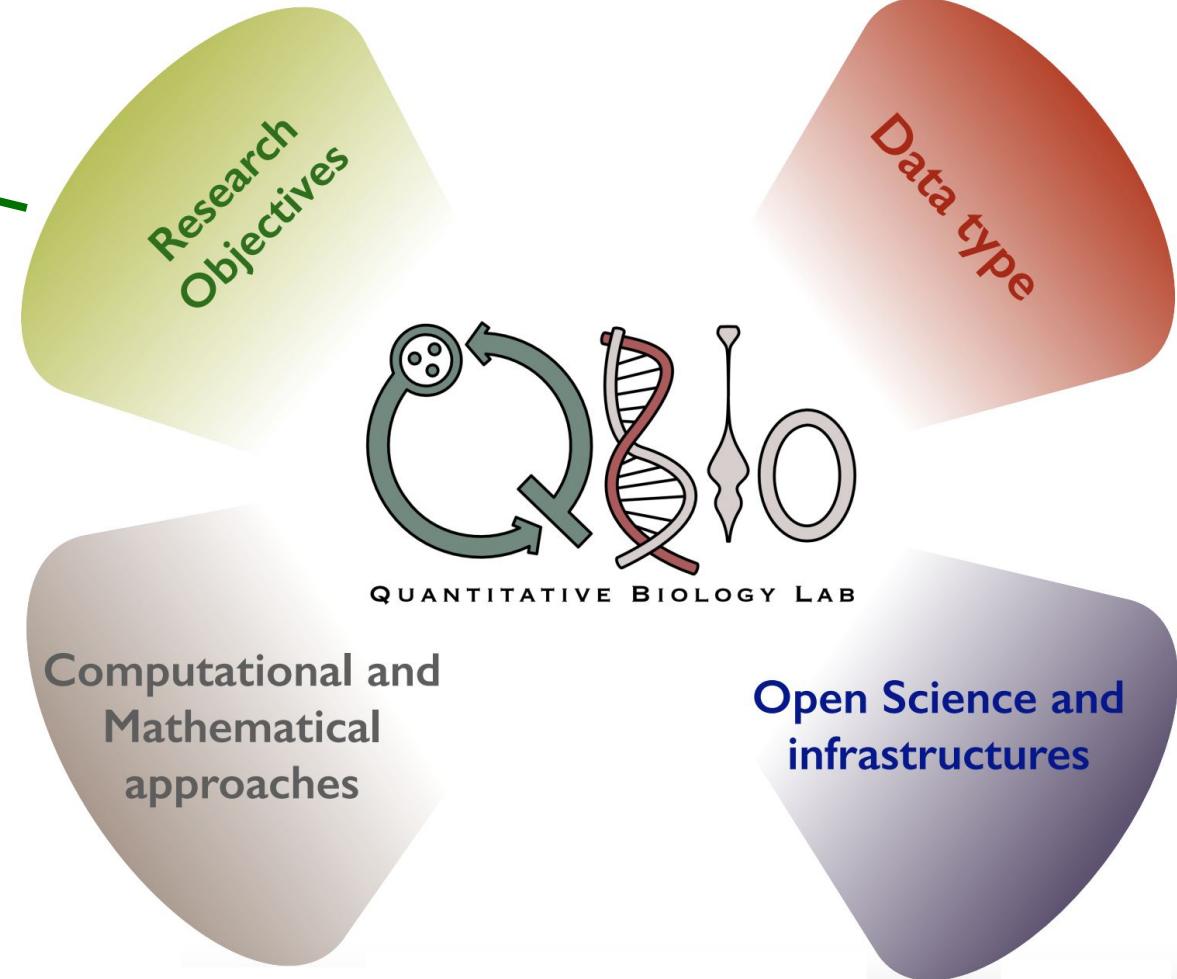
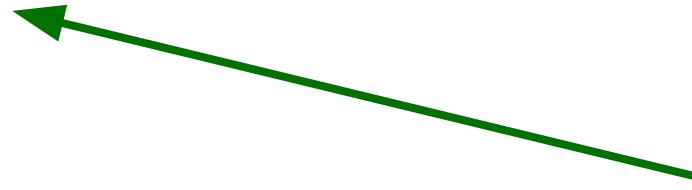


<http://github.com/qBioTurin/IEEEICHI2025>

Who are we?

Research Objectives:

- Patient stratification
- Identification of diagnostic and prognostic biomarkers
- Epidemiological models
- Mathematical models for cancer progression and treatment response
- Evolutionary Genomics and Modelling
- Basic Research: Cancer Genomics and Epigenomics



<https://qbio.di.unito.it/>



UNIVERSITÀ
DI TORINO

d1.unito.it
DIPARTIMENTO
DI INFORMATICA



Epidemic Modelling Frameworks for Studying Infectious Disease Dynamics

Who are we?



Marco Beccuti
Associate Professor

Marco Beccuti is currently associate professor at the Department of Computer Science of the University of Turin, scientific coordinator of InfoLife National Laboratory of CINI, scientific coordinator of the laboratory “HPC for biomed and AI” in ICxT, Chair of the Scientific Committee of the “HPC4AI” laboratory” at the University of Turin.



Simone Pernice
RTDA

Simone Pernice is currently assistant professor at the Department of Computer Science of the University of Turin, coordinator of Young InfoLife group of the Infolife National Laboratory of CINI. His current research activities mainly involve computational epidemiology, modeling of complex biological systems , and longitudinal data analysis.



Daniele Baccega
Post Doc

Daniele Baccega is a Post Doc researcher at the Department of Computer Science of the University of Turin. He holds a National PhD in Artificial Intelligence (Health and Life Sciences). His current research activities mainly involve computational epidemiology, deep learning, and data analysis.



Irene Terrone
Ph.D. Student

Irene Terrone is a PhD student at the Department of Computer Science of the University of Turin. Her research project focuses on the creation and study of biological models to analyze the spread and management of epidemics, integrating it with machine learning and generating models.

Who's presenting what?



Simone Pernice

RTDA

Introduction
Macro-level models

Micro-level models
From macro to micro level models



Daniele Baccega

Post Doc



Irene Terrone

Ph.D. Student

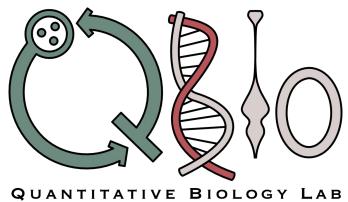
Case studies:
● GreatMod
● Forge4Flame



UNIVERSITÀ
DI TORINO

DIPARTIMENTO
DI INFORMATICA

di.unito.it



InfoLife
cini national lab



IEEE - International Conference on
Healthcare Informatics
Università della Calabria, Italy
June 18th-21st, 2025

Introduction



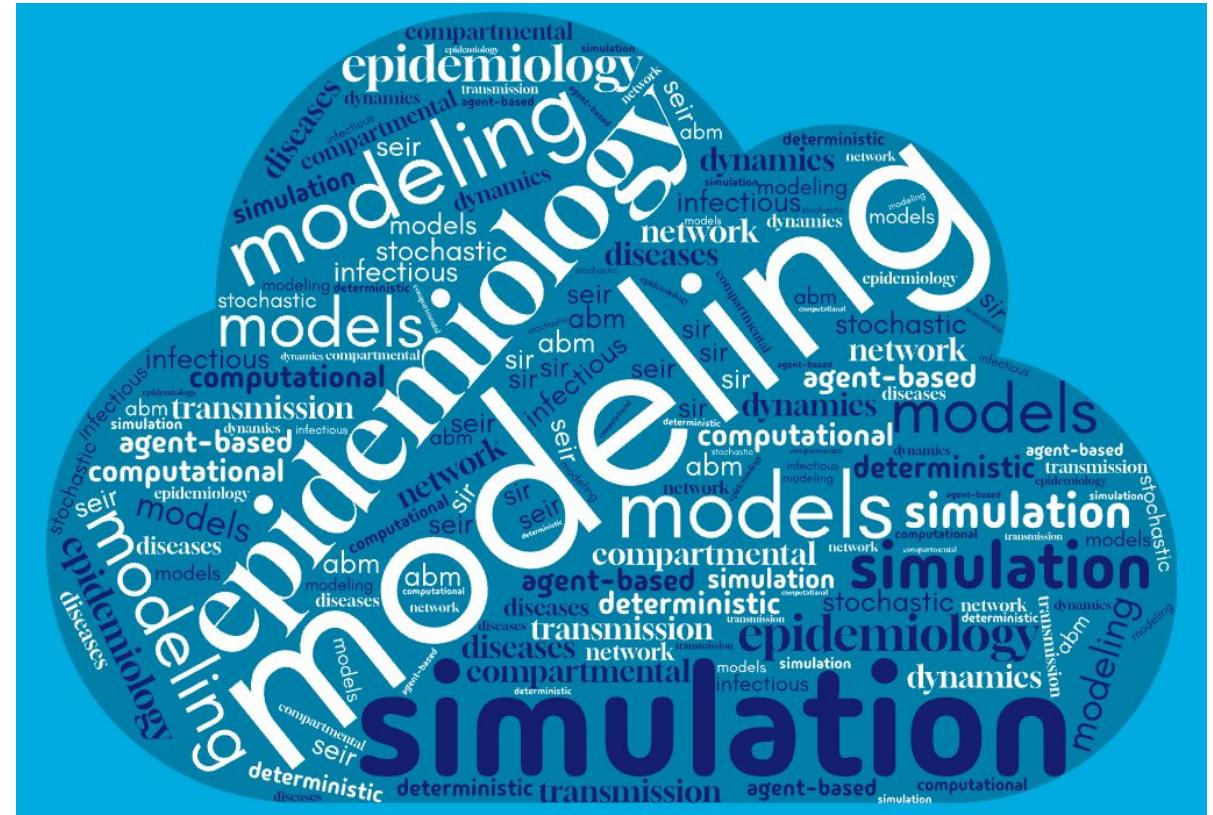
Computational modelling

Definition

The use of mathematics, statistics, physics and computer science to study the mechanism and behavior of complex systems by computer simulation.

Applications in different fields:

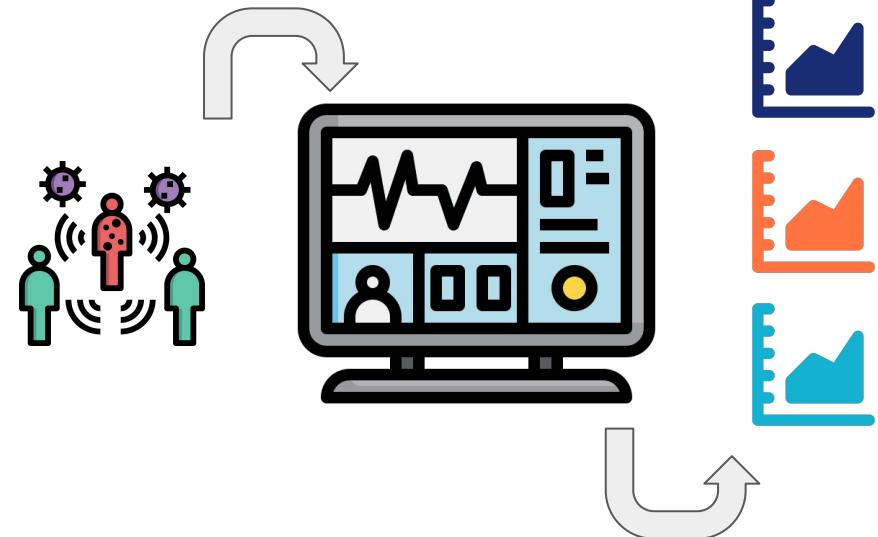
Network analysis, manufacturing systems, resource management, weather forecasting, clinical decision support, tracking infectious diseases, ...



Computational epidemiology

Definition

A multidisciplinary field that uses computational methods to simulate, analyze, and predict the spread of diseases within populations.



Key Components:

- **Epidemiological Models:** SIR, SEIR, ...
- **Computation:** Simulation approach, parameter estimation, data assimilation
- **Data Sources:** Case counts, mobility, demographics, contact networks

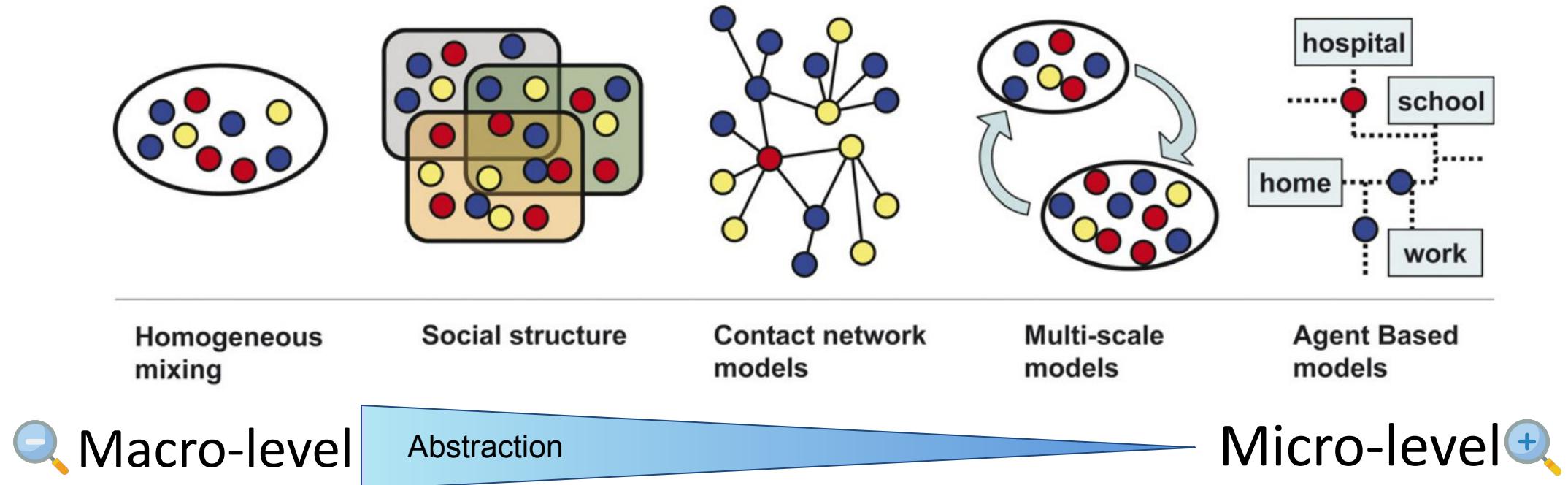


Handles **complex systems** with many interacting parts

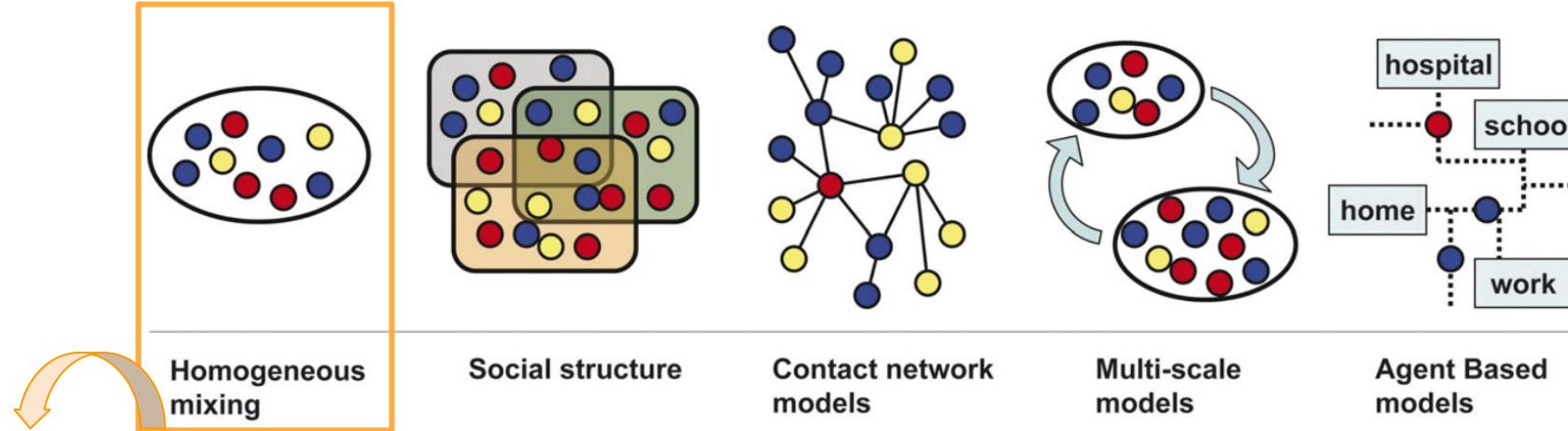
Enables **what-if scenarios** and policy testing

Allows for **rapid analysis** using real-time data

Different modelling approaches



Different modelling approaches



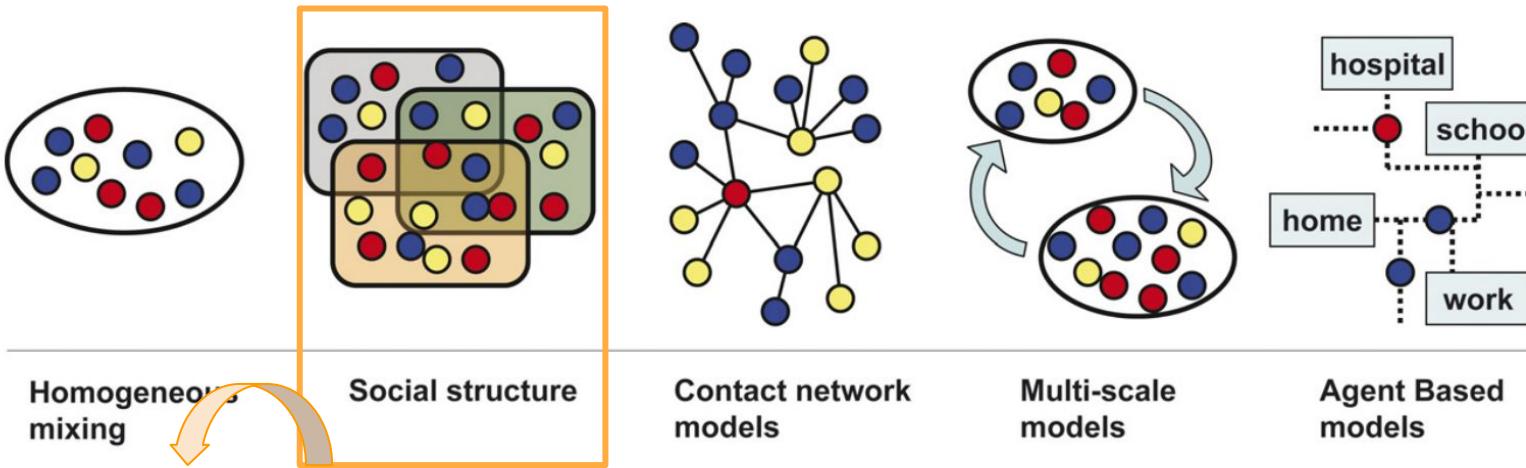
Homogeneous mixing: assumes that every individual in a population is equally likely to interact with every other individual and treats the population as a well-mixed group without any structure, where interactions happen randomly and uniformly.

Real-World Use Case: *Estimating how fast an infectious disease might spread in a new outbreak, by assuming homogeneous mixing*
to estimate:

- How many people might be infected in 30 days
- What the basic reproduction number R_0 is
- Whether the healthcare system will be overwhelmed

Model Type: SIR or SEIR compartmental model using **ordinary differential equations (ODEs)**, assuming population is like a "well-stirred soup"

Different modelling approaches



Social structure: accounts for the fact that individuals belong to different social groups or communities, based on factors such as age, economic status, region, and other demographic characteristics.

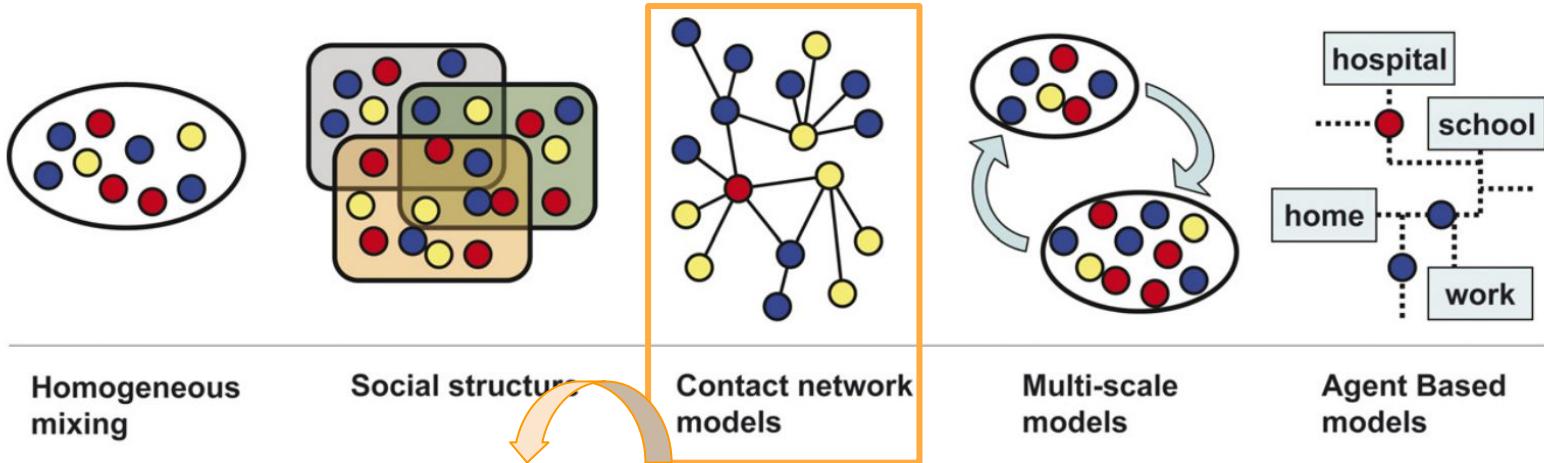
Real-World Use Case: *Evaluating the impact of closing schools vs. workplaces and adding and age structure, to simulate how transmission changes:*

- If schools are closed but workplaces stay open
- If elderly reduce contact but younger groups don't

Model Type:

- Structured compartmental models
- Age-stratified or region-based SIR/SEIR
- Matrix of contact rates between groups

Different modelling approaches



Contact network models: represent the population as a network, where nodes represent individuals, and edges represent the interactions or connections between them.

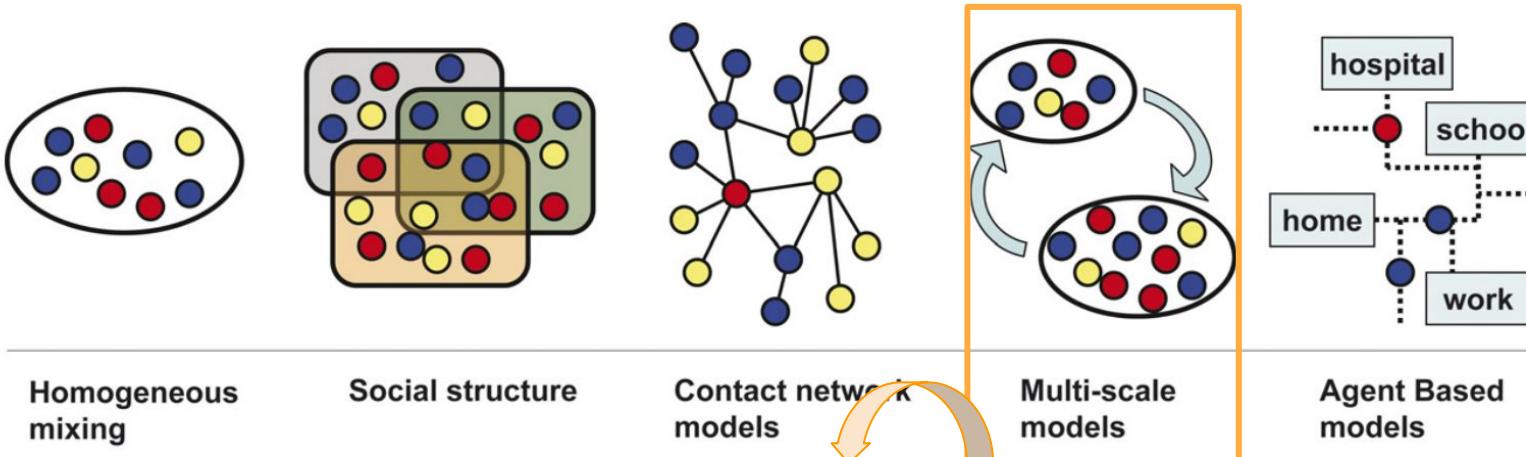
Real-World Use Case: *Identifying and managing super-spreader events.* The model identifies individuals with many connections (hubs) who could become super-spreaders. It helps simulate:

- What happens if these hubs isolate?
- How fast the outbreak could spread through friend networks

Model Type:

- Graph-based models
(nodes = individuals, edges = interactions)
- Network epidemiology

Different modelling approaches



Multi-scale models: integrate multiple levels of interaction, such as individual, group, and population levels, and consider the effects across these different scales.

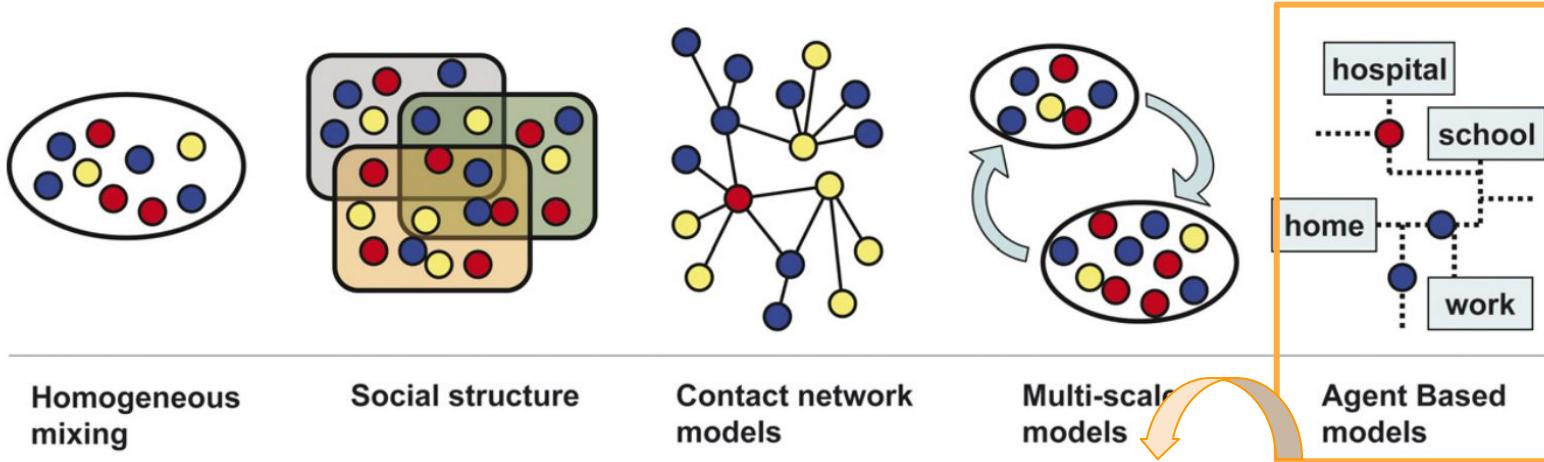
Real-World Use Case: *Understanding national-level spread and regional control strategies.*

- People interact at home, work, school
- Movement happens between cities (commuting, travel)
- Local healthcare capacity varies by region

Model Type:

- Hybrid models combining ODEs, networks, and spatial interactions
- Useful for modeling regional or national dynamics with local resolution

Different modelling approaches



Agent-based models: simulate the actions and interactions of individual agents (such as people or organizations) within specific environments. They consider different locations or settings (e.g., city, neighborhood, hospital, school, work, home) and model how agents move between these locations and interact based on defined rules.

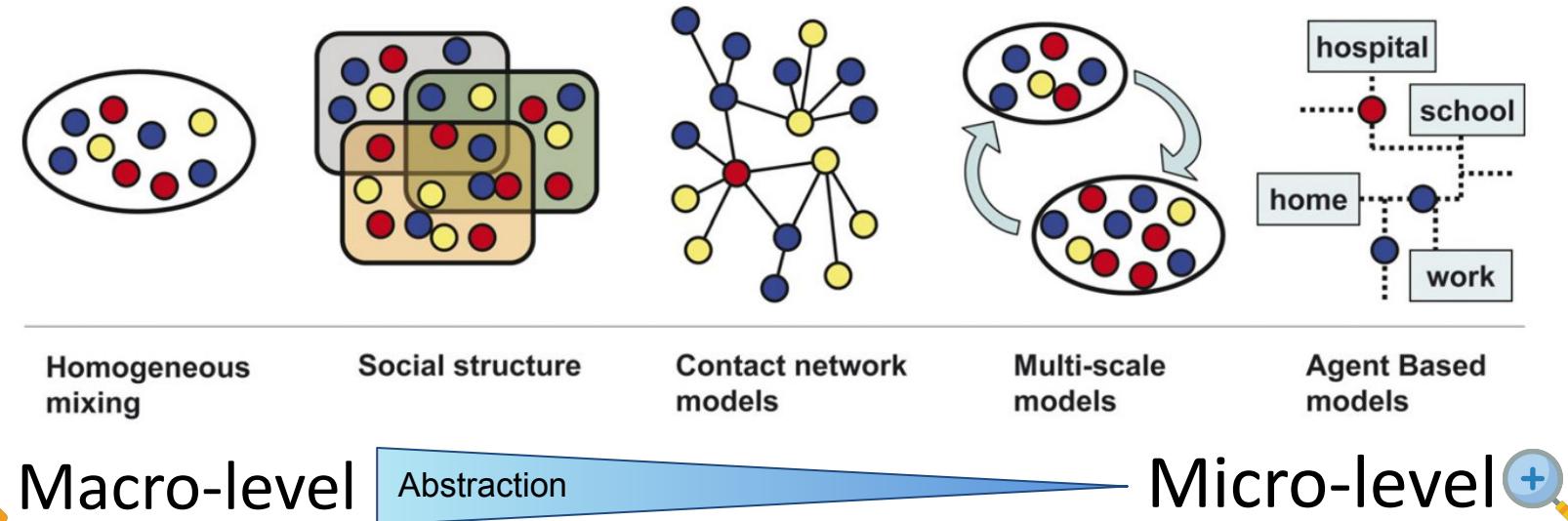
Real-World Use Case: *Testing the effectiveness of behavioral and spatial interventions.* The model explores:

- What if 60% of people wear masks?
- What if gyms close but offices remain open?
- How does work-from-home affect the spread?

Model Type:

- Individual-based simulation
- High computational cost but rich insights
- Allows for testing interventions like mask mandates

Macro- and micro-level models



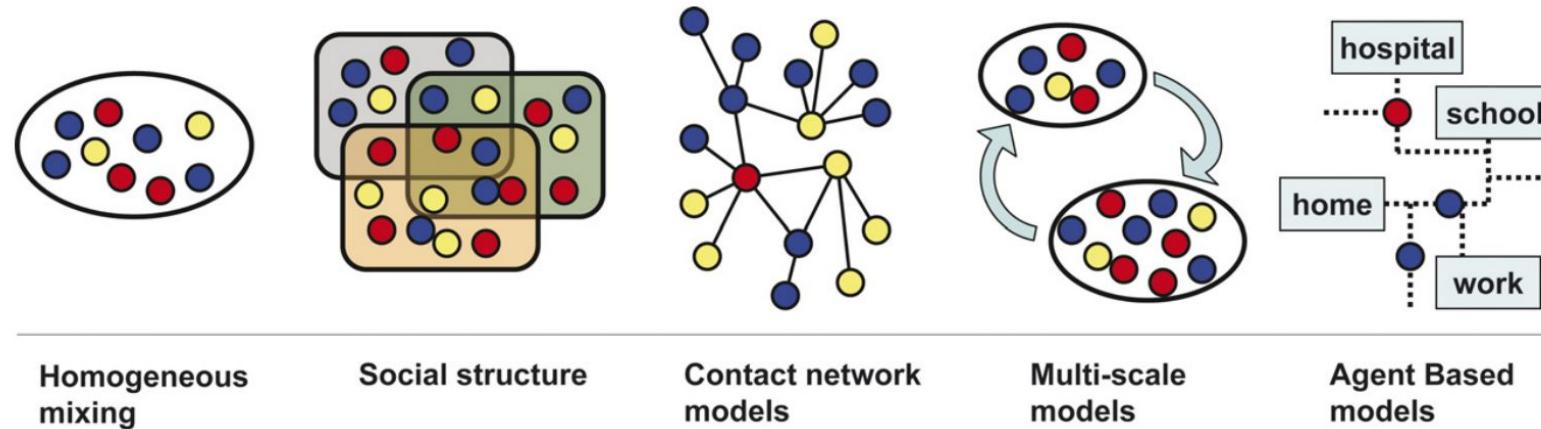
✓ Pros:

- Require **fewer data** and computational resources.
- Allow **fast simulations on large populations**.
- Useful for broad forecasts and general policy decisions.

✓ Pros:

- Capture **individual behaviors** and interactions.
- Allow detailed, **context-specific simulations**.
- Ideal for testing targeted interventions.

Macro- and micro-level models



Macro-level Abstraction Micro-level

✓ Pros:

- Require **fewer data** and computational resources.
- Allow **fast simulations on large populations**.
- Useful for broad forecasts and general policy decisions.

✗ Cons:

- **Do not capture individual behavior** or local dynamics.
- **Assume uniform** or simplified mixing **patterns**.
- Less accurate in heterogeneous or small-scale settings.

✓ Pros:

- Capture **individual behaviors** and interactions.
- Allow detailed, **context-specific simulations**.
- Ideal for testing targeted interventions.

✗ Cons:

- Require detailed, **high-resolution data**.
- **Computationally intensive**.
- More complex to calibrate and validate.

What do you expect from this tutorial?

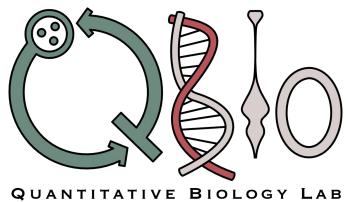
1. Can we estimate the impact of interventions like lockdowns or vaccines?
2. How do individual behaviors affect the epidemic?
3. What's the trade-off between simplicity and detail in a model?
4. When should we use an macro model, and when is a micro model better?



UNIVERSITÀ
DI TORINO

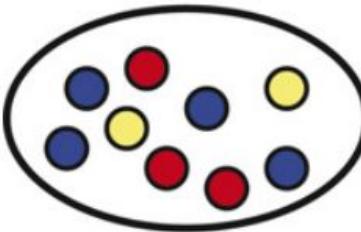
DIPARTIMENTO
DI INFORMATICA

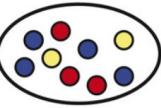
di.unito.it



IEEE - International Conference on
Healthcare Informatics
Università della Calabria, Italy
June 18th-21st, 2025

Macro-level models





Macro-level models

The first major class of modeling approaches: **Macro-Level Models**, often called **compartmental models**.

These models don't simulate each person individually. Instead, they divide the entire population into **groups**, for example:

- Susceptible
- Infected
- Recovered

The beauty of these models is in their **simplicity**. They help answer questions like:

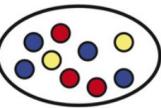
- *How fast will the outbreak grow?*
- *What's the peak number of cases?*
- *What happens if we vaccinate 50% of the population?*



We assume the population is well mixed (like stirring soup) where **everyone has an equal chance of interacting with everyone else**.



So while they don't give detailed answers about individual behavior, they're **powerful for seeing the big picture**.



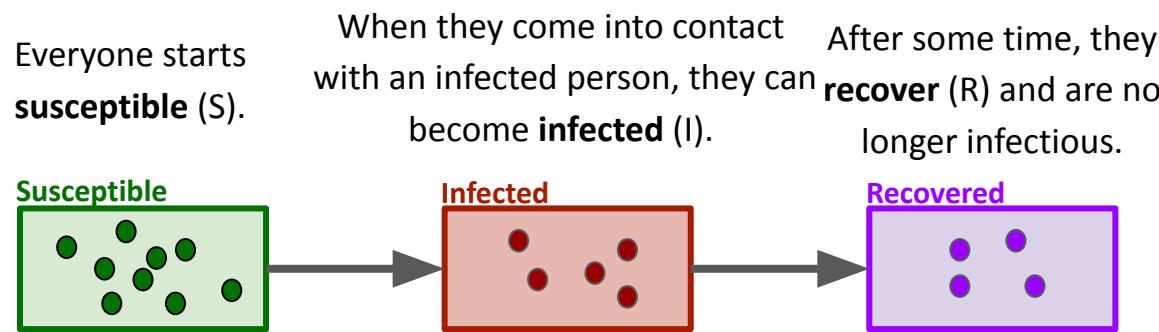
Compartmental model



We assume the population is well mixed (like stirring soup) where **everyone has an equal chance of interacting with everyone else.**



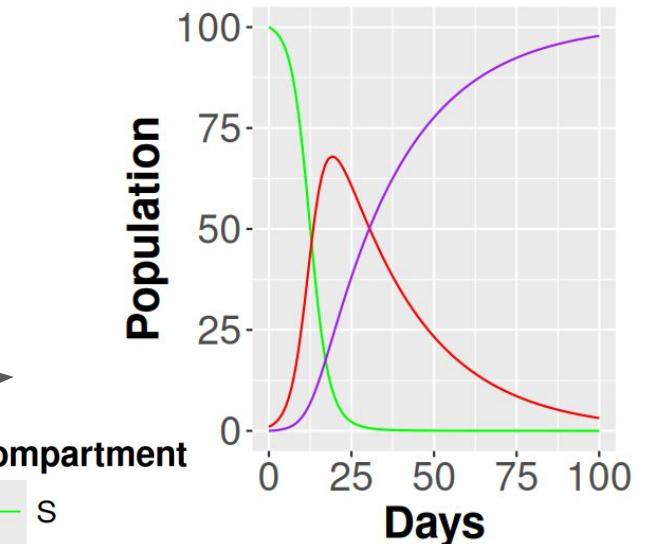
So while they don't give detailed answers about individual behavior, they're **powerful for seeing the big picture**.



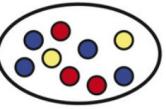
Individuals can move from the S compartment to the I compartment, but not back. They can also move from the I compartment to the R compartment, but not back. The rates at which these movements between compartments occur determine the **dynamics of the model**.

SIR compartmental model as a flow diagram

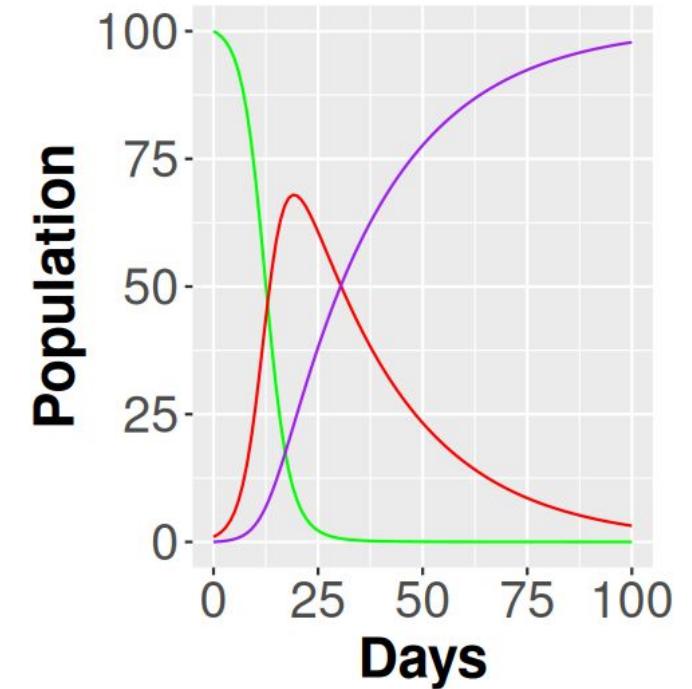
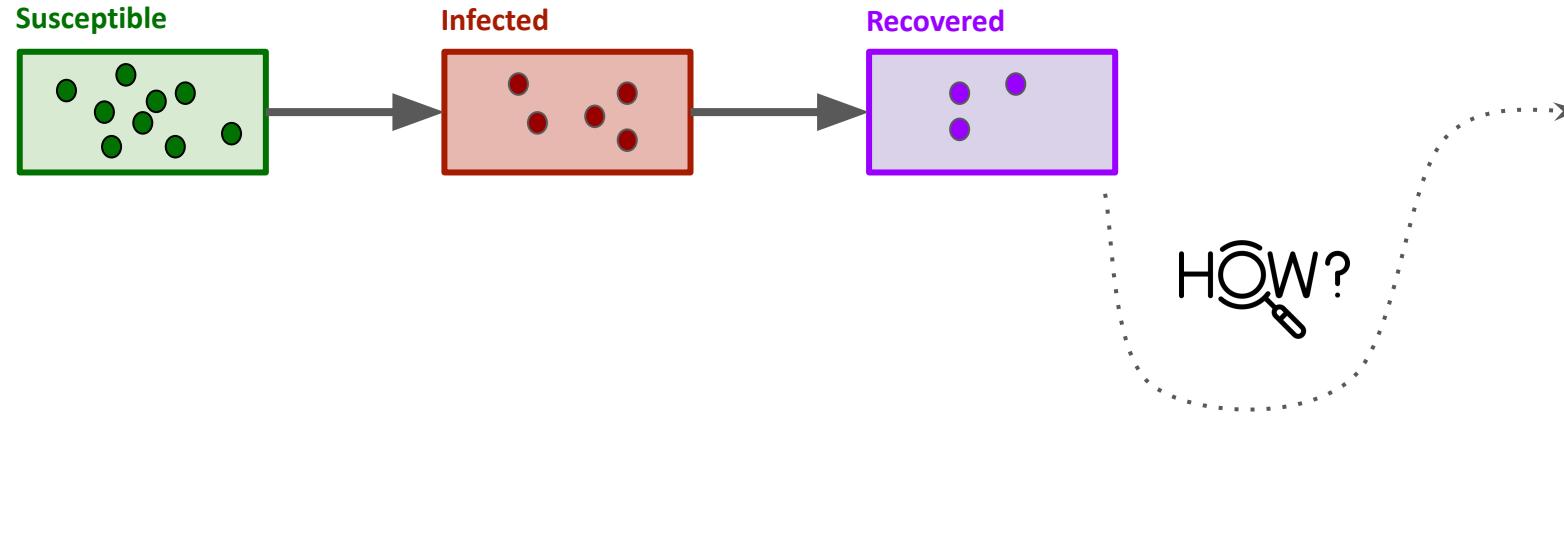
This diagram helps us visualize **what we're modeling**

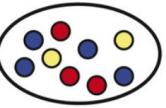


Compartment	Color
S	Green
I	Red
R	Purple

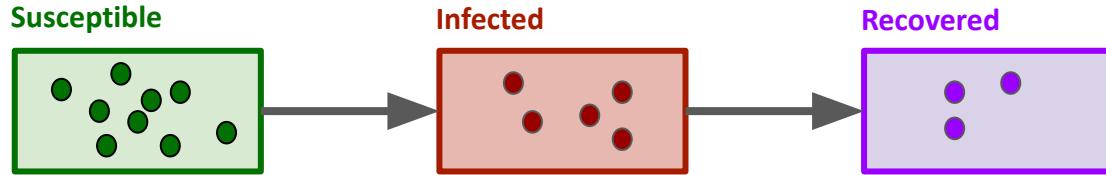


The mathematical processes underlying





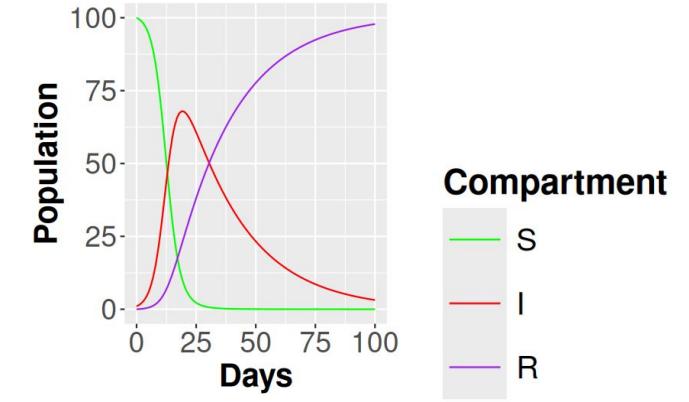
The mathematical processes underlying



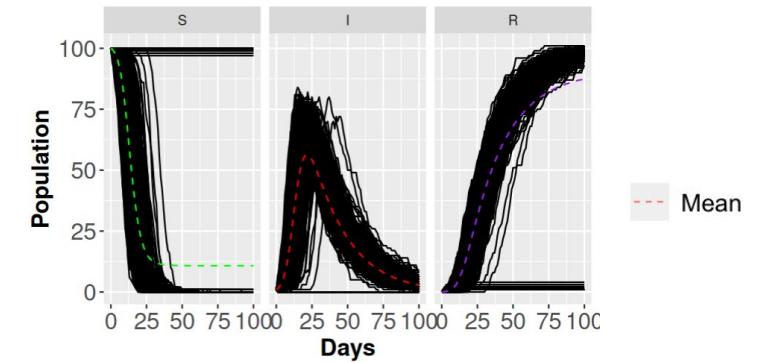
Once we have a flow diagram we can go in **two directions**:

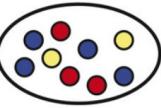


Deterministic approach: we assume the population is large enough that randomness averages out. This leads to **ordinary differential equations (ODEs)**



Stochastic approach: we recognize that especially in small populations or early outbreak stages, random events matter. This leads to **Continuous Time Markov Chain (CTMC)**.



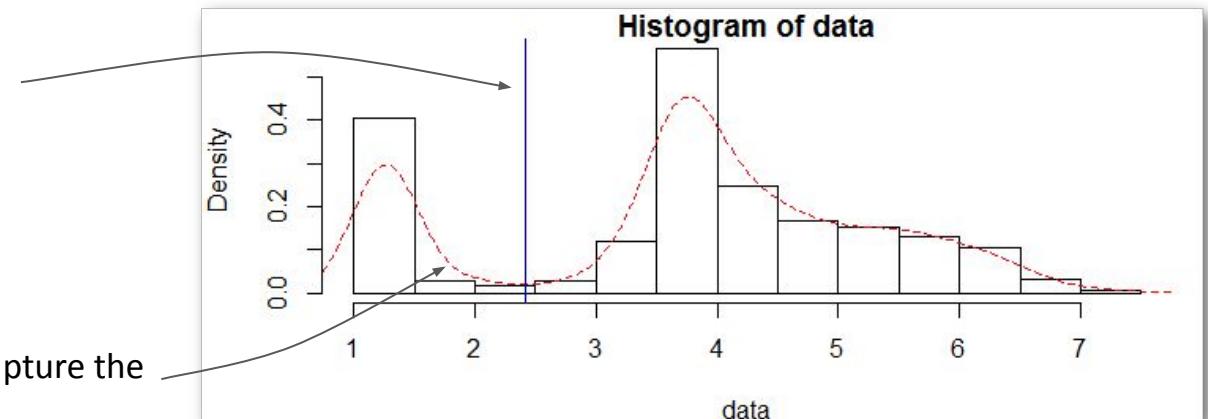


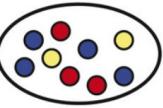
The mathematical processes underlying

Deterministic	Stochastic
Ordinary differential equations	Continuous-time Markov chains
Future is “predictable” given present knowledge	Includes randomness; every simulation is different
Wide range of techniques available for analysis	Not as many techniques for analysis; often rely on simulations
Good for large number of individuals; qualitative analysis	Better for simulating dynamics with small numbers of individuals
Represents population average	Represents population variability

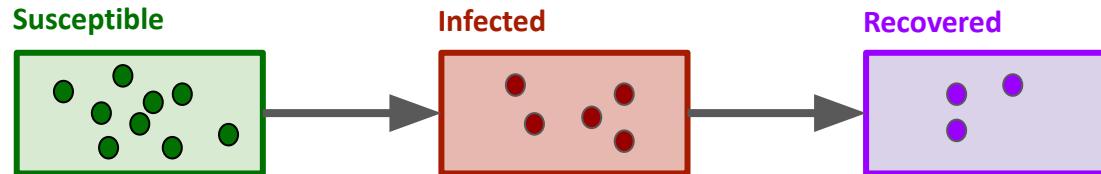
Deterministic models can be solved easier, but they provide only the average system behavior.

Stochastic models are more computationally demanding, but they can capture the stochastic nature of a biological process.





Stochastic approach



Two possible stochastic events:
Infection: $(S, I, R) \rightarrow (S-1, I+1, R)$
with rate β

Recovery: $(S, I, R) \rightarrow (S, I-1, R+1)$
with rate γ

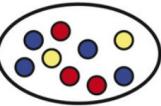
The system is in a discrete state:

$(\#S, \#I, \#R)$

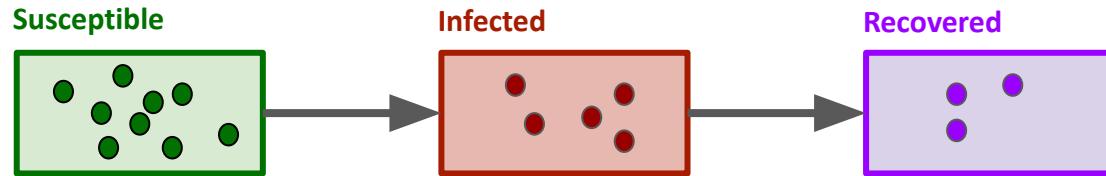
where $\#S$, $\#I$, and $\#R$ are **integer counts** of individuals in each compartment.

In the example, the state is:

$(\#S = 8, \#I = 5, \#R = 4)$



Stochastic approach



Two possible stochastic events:
Infection: $(S, I, R) \rightarrow (S-1, I+1, R)$
with rate β

Recovery:
 $(S, I, R) \rightarrow (S, I-1, R+1)$
with rate γ

The system is in a discrete state:

$(\#S, \#I, \#R)$

where $\#S$, $\#I$, and $\#R$ are **integer counts** of individuals in each compartment.

In the example, the state is:

$(\#S = 8, \#I = 5, \#R = 4)$

These Kolmogorov equations (also called **master equations**) describe how those probabilities to be in a specific state change over time:

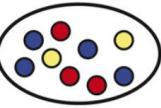
$$\frac{dP(S, I, R, t)}{dt} = \beta(S + 1)(I - 1)P(S + 1, I - 1, R, t) + \gamma(I + 1)P(S, I + 1, R - 1, t) - [\beta SI + \gamma I]P(S, I, R, t)$$

P(S, I, R, t) is the probability the system is in state (S, I, R) at time t

probability **into** state (S, I, R) from a transition **into** it (someone got infected in the previous state).

probability from someone recovering

subtracts probability, because once someone gets infected or recovers, we leave the current state.



Stochastic approach

Kolmogorov equations:

$$\frac{d\pi(s_i, \nu)}{d\nu} = \sum_{s_k} \pi(s_k, \nu) q_{k,i}$$

$\pi(s_i, \nu)$ = probability to be in the state s_i at time ν

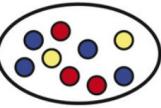
$q_{k,i}$ = rate from s_k to s_i



One Kolmogorov equation per state

State Space Explosion

- [1] Gillespie (1977) *The journal of physical chemistry*
- [2] Gillespie (2001) *The journal of physical chemistry*
- [3] T.G. Kurtz (1970) *Journal of Applied Probability*



Stochastic approach

Kolmogorov equations:

$$\frac{d\pi(s_i, \nu)}{d\nu} = \sum_{s_k} \pi(s_k, \nu) q_{k,i}$$

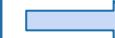
$\pi(s_i, \nu)$ = probability to be in the state s_i at time ν

$q_{k,i}$ = rate from s_k to s_i



One Kolmogorov equation per state

State Space Explosion

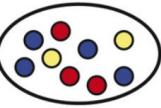


These equations are hard to solve directly for large systems, but they form the foundation of stochastic simulations like **Gillespie's algorithm**.



Stochastic Simulation Algorithm [1]:
exact algorithm to simulate the events that
might occur in the system

- [1] Gillespie (1977) *The journal of physical chemistry*
- [2] Gillespie (2001) *The journal of physical chemistry*
- [3] T.G. Kurtz (1970) *Journal of Applied Probability*



Stochastic approach

Kolmogorov equations:

$$\frac{d\pi(s_i, \nu)}{d\nu} = \sum_{s_k} \pi(s_k, \nu) q_{k,i}$$

$\pi(s_i, \nu)$ = probability to be in the state s_i at time ν

$q_{k,i}$ = rate from s_k to s_i



One Kolmogorov equation per state
State Space Explosion



These equations are hard to solve directly for large systems, but they form the foundation of stochastic simulations like **Gillespie's algorithm**.



Stochastic Simulation Algorithm [1]:
exact algorithm to simulate the events that might occur in the system

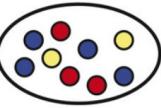


T -leaping algorithm [2] a Poisson approximation to leap over many fast events



Longer simulation with an increasing number of events!!

- [1] Gillespie (1977) *The journal of physical chemistry*
- [2] Gillespie (2001) *The journal of physical chemistry*
- [3] T.G. Kurtz (1970) *Journal of Applied Probability*



Stochastic approach

Kolmogorov equations:

$$\frac{d\pi(s_i, \nu)}{d\nu} = \sum_{s_k} \pi(s_k, \nu) q_{k,i}$$

$\pi(s_i, \nu)$ = probability to be in the state s_i at time ν

$q_{k,i}$ = rate from s_k to s_i



One Kolmogorov equation per state

State Space Explosion

These equations are hard to solve directly for large systems, but they form the foundation of stochastic simulations like **Gillespie's algorithm**.



Stochastic Simulation Algorithm [1]:
exact algorithm to simulate the events that
might occur in the system



Deterministic Approximation

Mean field analysis (or fluid approximation) [3]

τ -leaping algorithm [2] a Poisson
approximation to leap over many fast
events



Longer simulation with an increasing
number of events!!

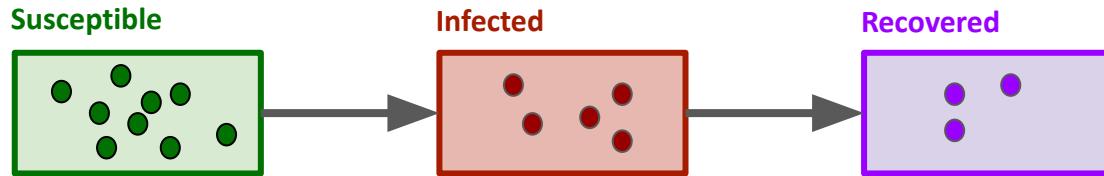
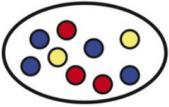


[1] Gillespie (1977) *The journal of physical chemistry*

[2] Gillespie (2001) *The journal of physical chemistry*

[3] T.G. Kurtz (1970) *Journal of Applied Probability*

Deterministic approximation



Kolmogorov equations:

$$\frac{d\pi(s_i, \nu)}{d\nu} = \sum_{s_k} \pi(s_k, \nu) q_{k,i}$$

$\pi(s_i, \nu)$ = probability to be in the state s_i at time ν
 $q_{k,i}$ = rate from s_k to s_i

Deterministic Approximation

$$\frac{dX_i^N(\nu)}{d\nu} = f(X^N(\nu), \nu)$$

$X_i^N(\nu)$ Number of elements in each variable i -th at time ν



Number of Kolmogorov equations

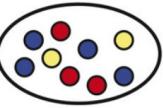
=

Number of all the possible states of the SIR model that can be reached from a specific initial state

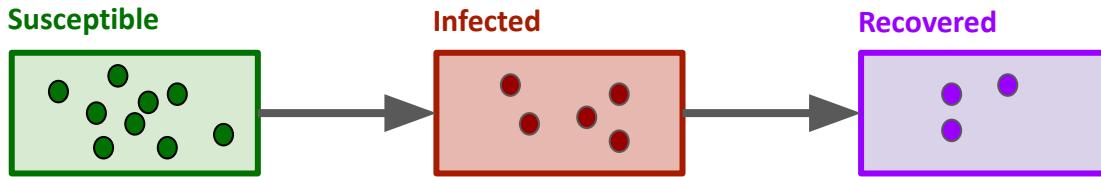
If the initial state is (#S = 8, #I = 5, #R = 4) we have more than 60 states

One Ordinary Differential equation (ODE) per variable that characterizes the state!

The state (#S, #I, #R): 3 Variables!



Deterministic approach



People are leaving the susceptible compartment

$$\frac{dS}{dt} = -\beta SI$$

People are recovering and moving to the R compartment

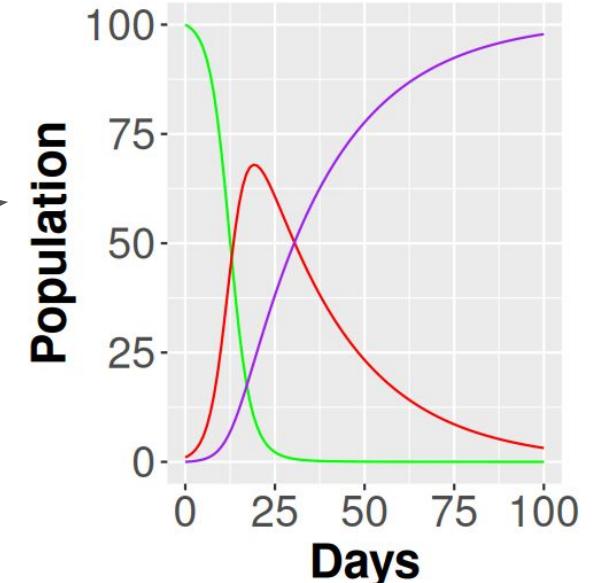
$$\frac{dI}{dt} = \beta SI - \gamma I$$

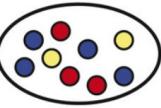
$$\frac{dR}{dt} = \gamma I$$

β : the transmission rate — how often a susceptible and infected person interact

γ : the recovery rate — how fast people leave the infected state

Solving (integrating)
the equations





Several compartment models



SEIQR Model (Susceptible-Exposed-Infected-Quarantined-Recovered)

- **Used for:** Diseases with **quarantine or isolation policies**

SIRV Model (Susceptible-Infected-Recovered-Vaccinated)

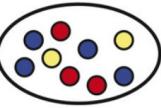
- **Used when** vaccination is implemented.

SAIR Model (Susceptible-Asymptomatic-Infected-Recovered)

- **Used for:** Diseases with **asymptomatic carriers**

SEIRD Model (Susceptible-Exposed-Infected-Recovered-Deceased)

- Combines latency (E) and death (D) aspects. Realistic for diseases like **Ebola or COVID-19**



Several compartment models



SEIQR Model (Susceptible-Exposed-Infected-Quarantined-Recovered)

- Used for: Diseases with **quarantine or isolation policies**

SIRV Model (Susceptible-Infected-Recovered-Vaccinated)

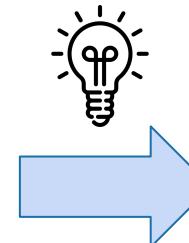
- Used when vaccination is implemented.

SAIR Model (Susceptible-Asymptomatic-Infected-Recovered)

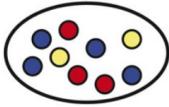
- Used for: Diseases with **asymptomatic carriers**

SEIRD Model (Susceptible-Exposed-Infected-Recovered-Deceased)

- Combines latency (E) and death (D) aspects. Realistic for diseases like **Ebola or COVID-19**



*It could be useful to have a **general modelling framework** to simulate the models easily!!!!*



Graphical User Interface:

- Interactive modeling via Petri Net formalism
- Construction and visualization of system dynamics
- User-friendly interface for dynamic system modeling

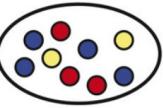
Epimod Library:

- Simulations of complex systems
- 'What-if' scenario analysis
- Detailed result checking (plots, data visualization)

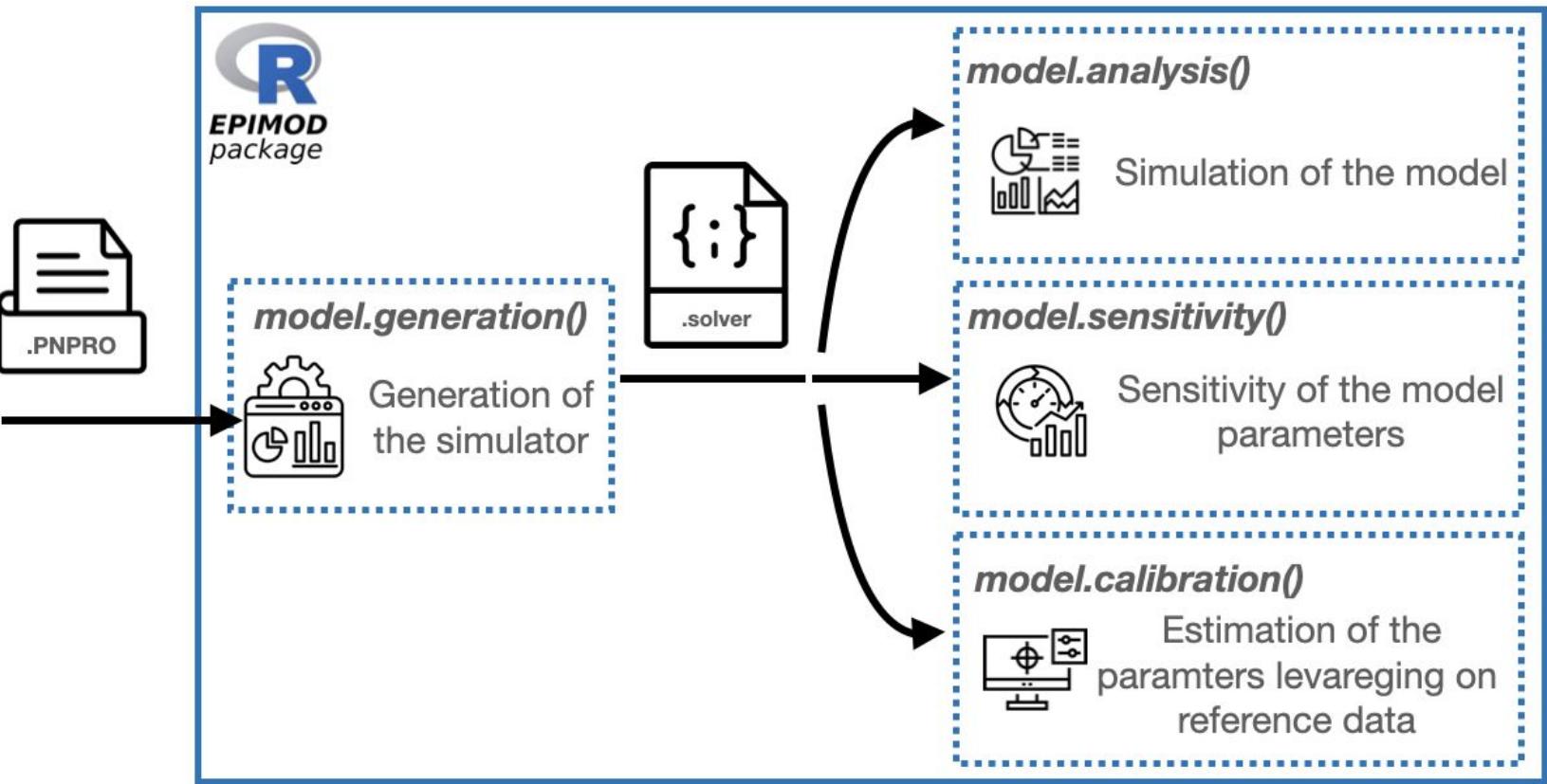
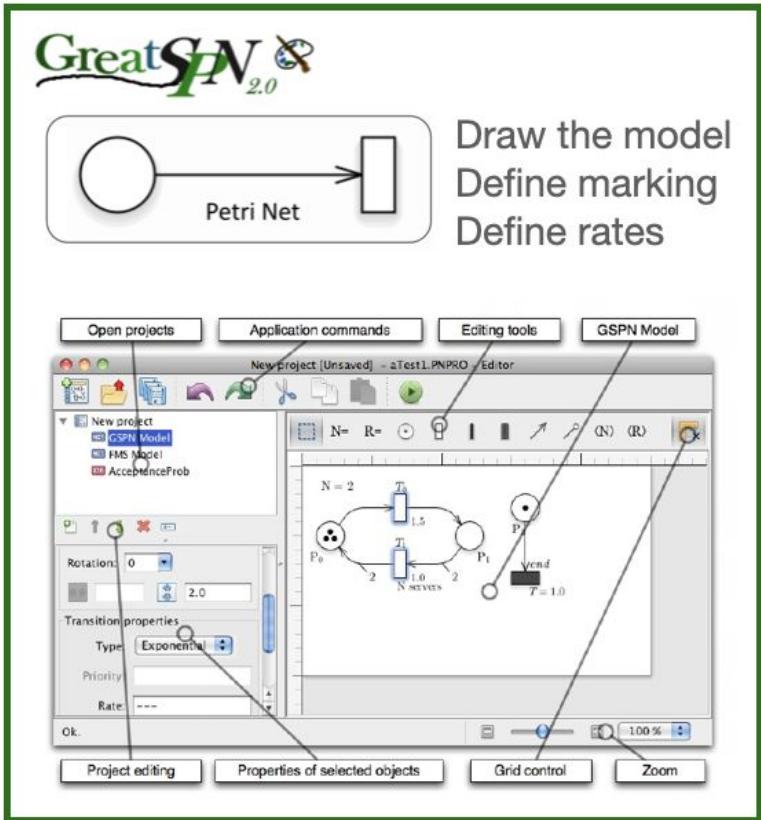
Docker Containers:

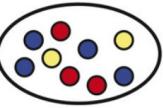
- Simulations reproducibility
- Simplifies deployment, updates, and maintenance

<https://qbioturin.github.io/epimod/>



GreatMod





Petri Net: a graphical formalism to draw model

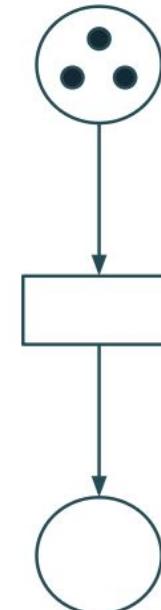
Petri Net (PN) is graphical formalism (bipartite graph), conveniently used for the analysis of complex models. It allows us to derive qualitative and quantitative properties of the system.

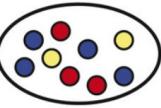
Places (circles): Represent *states* or *conditions* (e.g., a molecule present, a person infected).

Transitions (rectangles): Represent *events* or *actions* that change the state (e.g., infection, recovery).

Tokens (dots inside places): Indicate the *current state* (how many resources or individuals are in a place).

Arcs (arrows): Show how places and transitions are connected (direction matters).





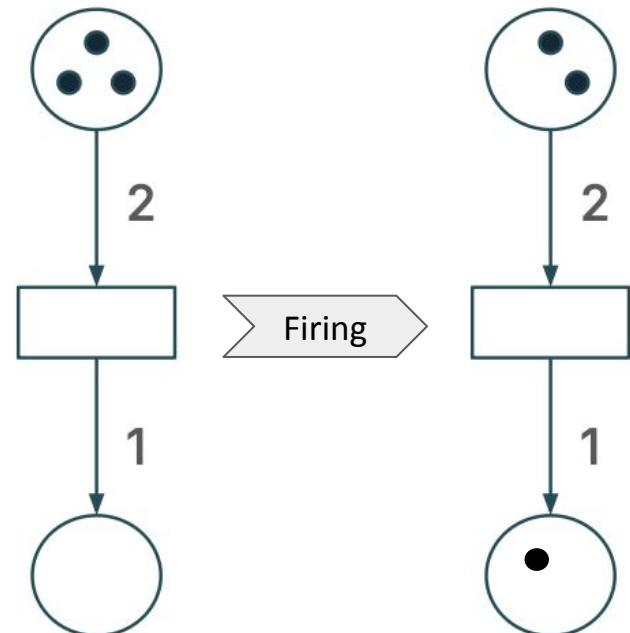
Stochastic Petri Net

Arcs Multiplicity:

- Arcs can have multiplicities, indicating the number of tokens transferred between places and transitions.

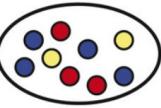
Transitions Firing:

- A transition 'fires' when the input places contain enough tokens based on the arc multiplicities.
- Firing consumes tokens from input places and produces tokens in output place



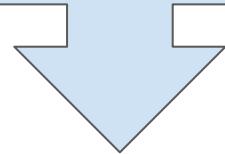
Firing Rate:

- Determines how quickly a transition converts input tokens into output tokens



Stochastic Petri Net

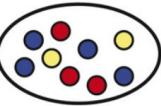
Exponentially distributed random delays are associated with transition firings (time to wait before the tokens are removed from/ added to places).



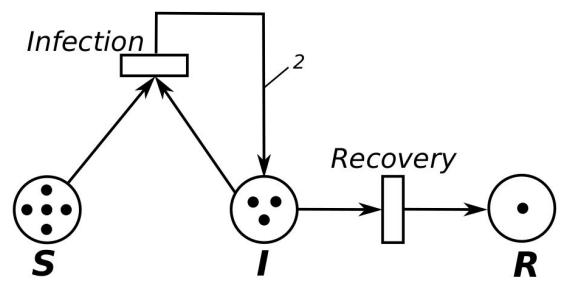
The **introduction of time** allows to model the *temporal dynamics* of the system

The process representing the dynamic of the SPN model is:

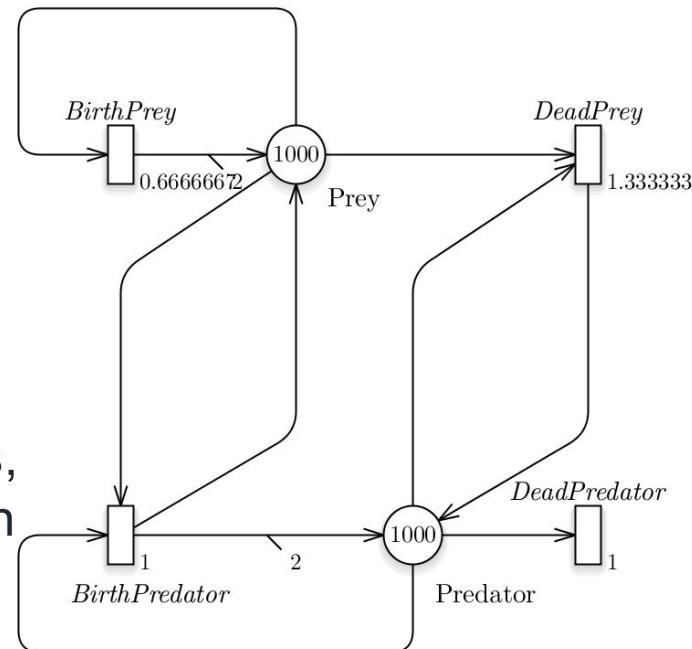
Continuous Time Markov Chain



Some examples

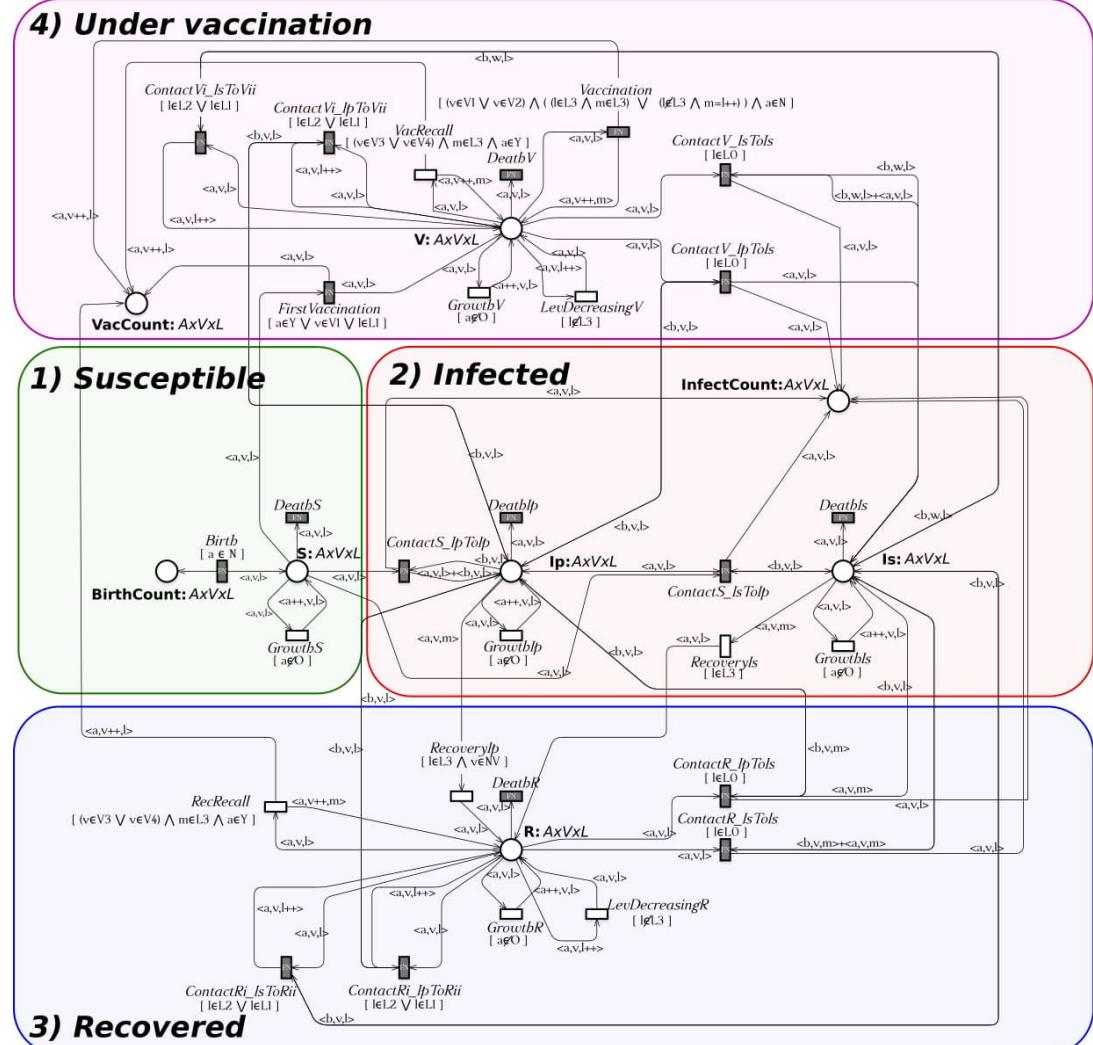


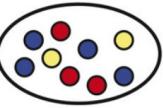
SIR model,
one of the simplest
compartmental models



Predator–prey equations,
biological systems in which
two species interact

Investigation of the
pertussis epidemiology
in Italy

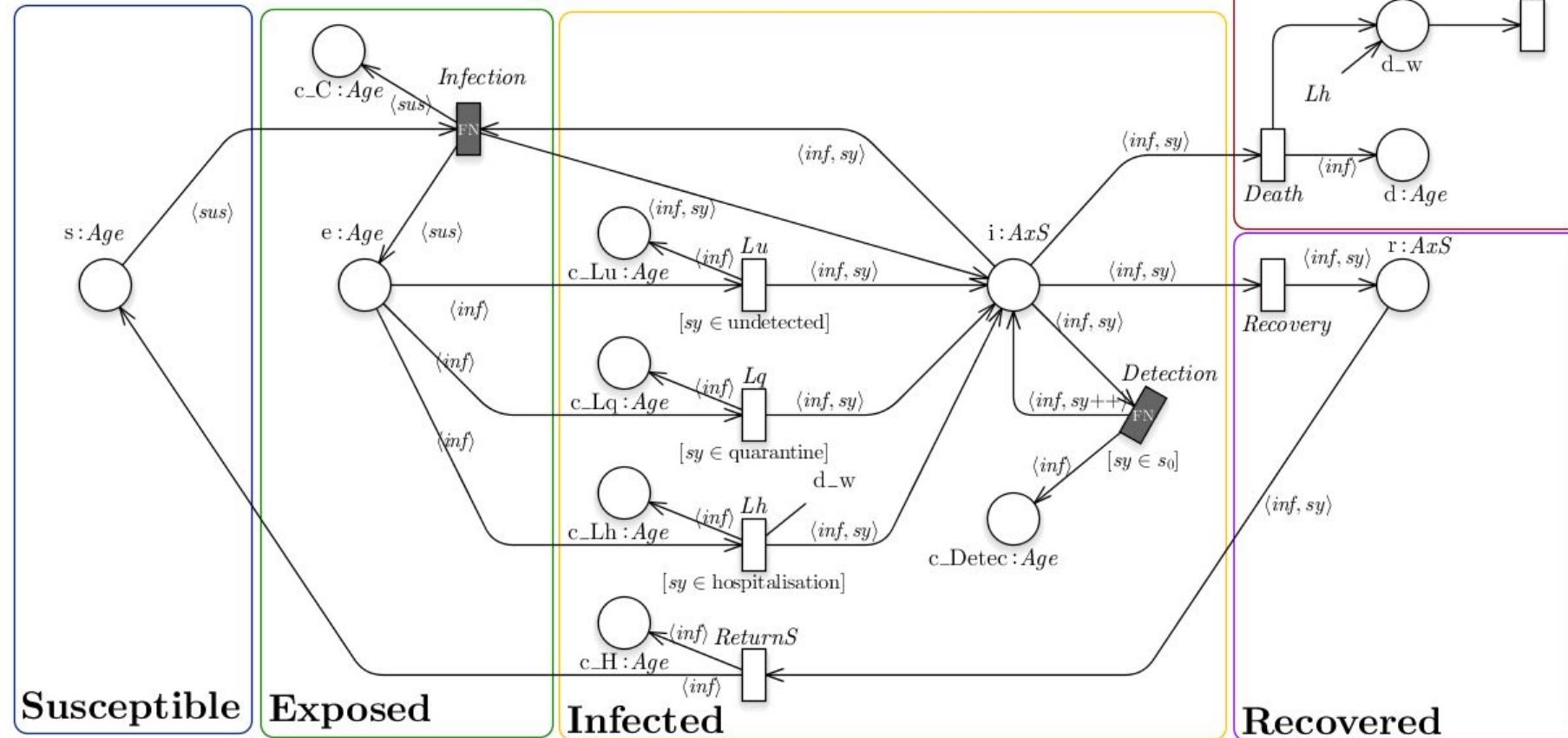




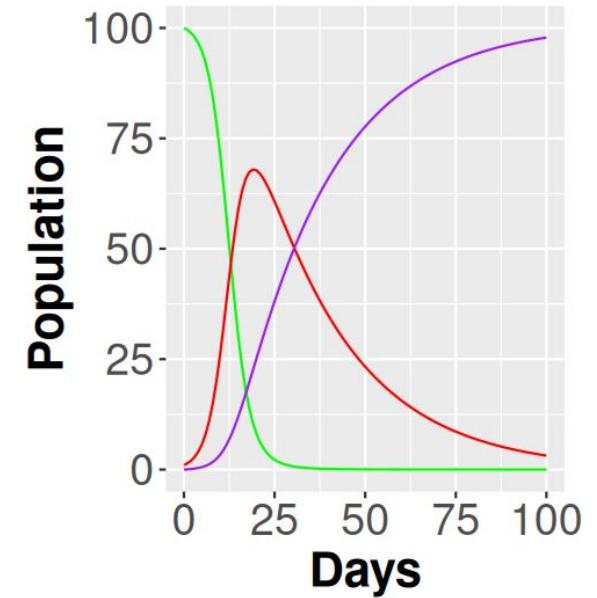
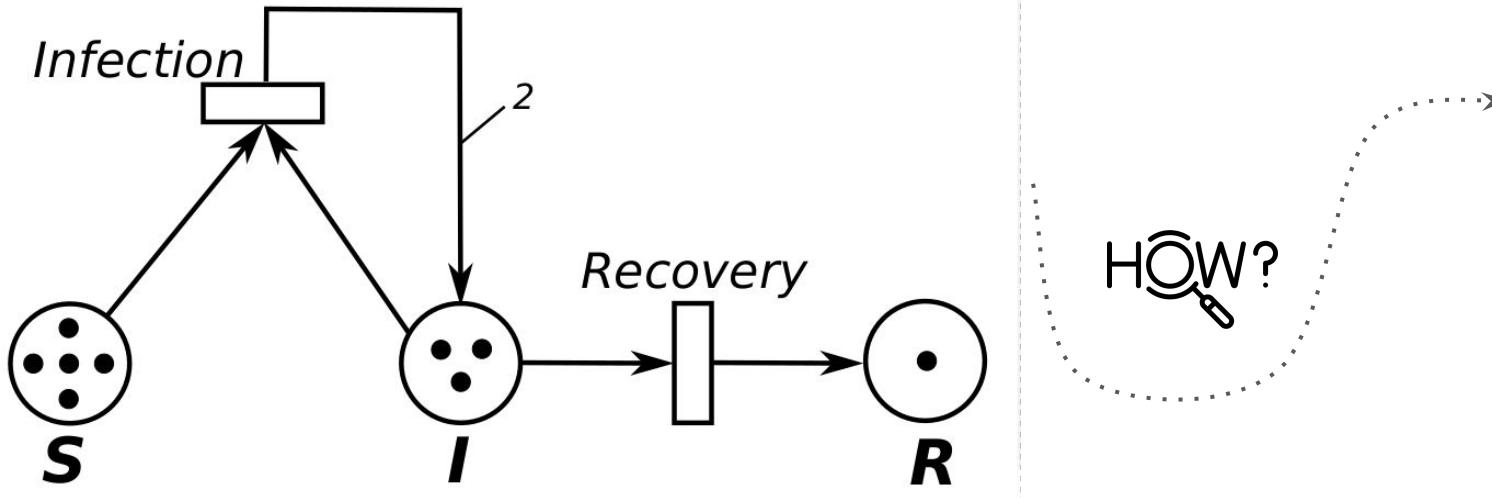
COVID-19 model

```

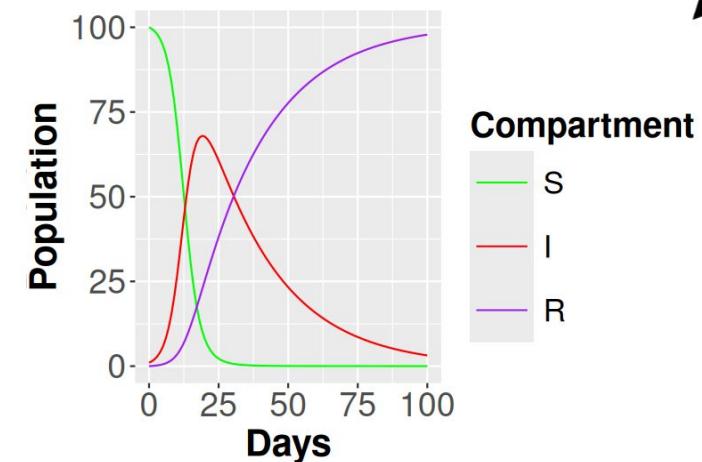
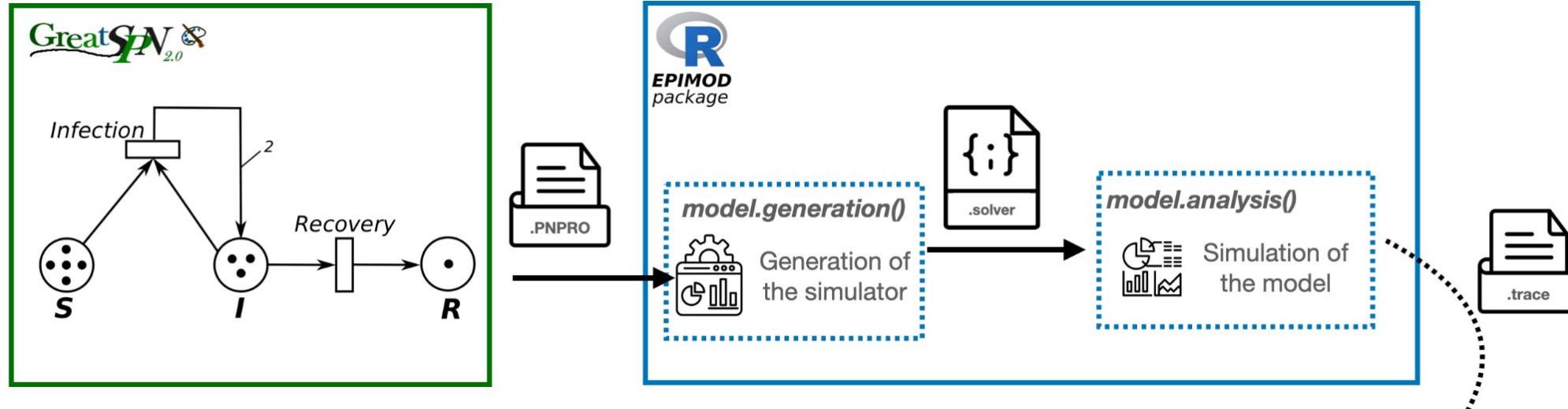
domain  $AxS = Age \times Symptom$ 
class Symptom = circular { $s_0$ } is undetected + { $s_1$ } is quarantine + { $s_2$ } is hospitalisation
class Age = { $a_0$ } is A0+{ $a_1$ } is A1+{ $a_2$ } is A2
  
```



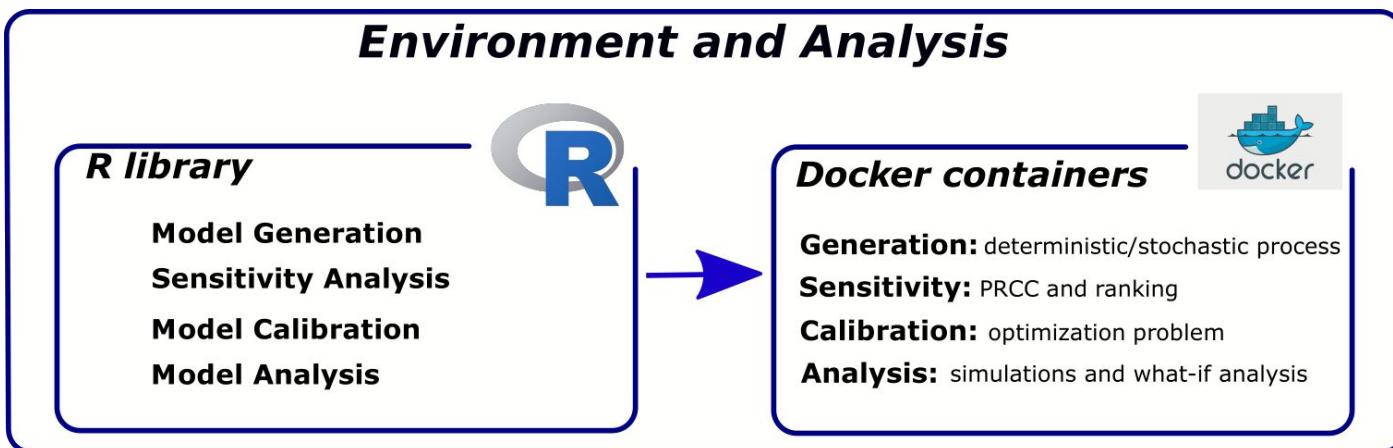
From graphics to dynamics



From graphics to dynamics



- General modeling framework for the analysis of epidemiological/biological systems;
- Tool easily accessible by any researcher even without advanced mathematical and computational skills.



Main features:

- **Graphical formalism:** you don't have to write equations, you draw 'em!
- **Modular:** you can shuffle the components to get what you need
- **R package:** no configuration is needed, you just install it
- **Virtualization with Docker:** the framework runs on your laptop and rocks on your data-center

Docker containerization

- Reproducibility is guaranteed by providing **Docker containers** packing all the required software and libraries frozen at a given version.
- Docker images are equivalent to Virtual Machines but **Docker container provide an operating-system-level virtualization** by abstracting the user space rather than the hardware, so that all the running docker images share the same kernel.
- Abstracting the user space guarantees better performance at the cost of minor portability (i.e., docker images built on Linux will not run on Windows systems.)



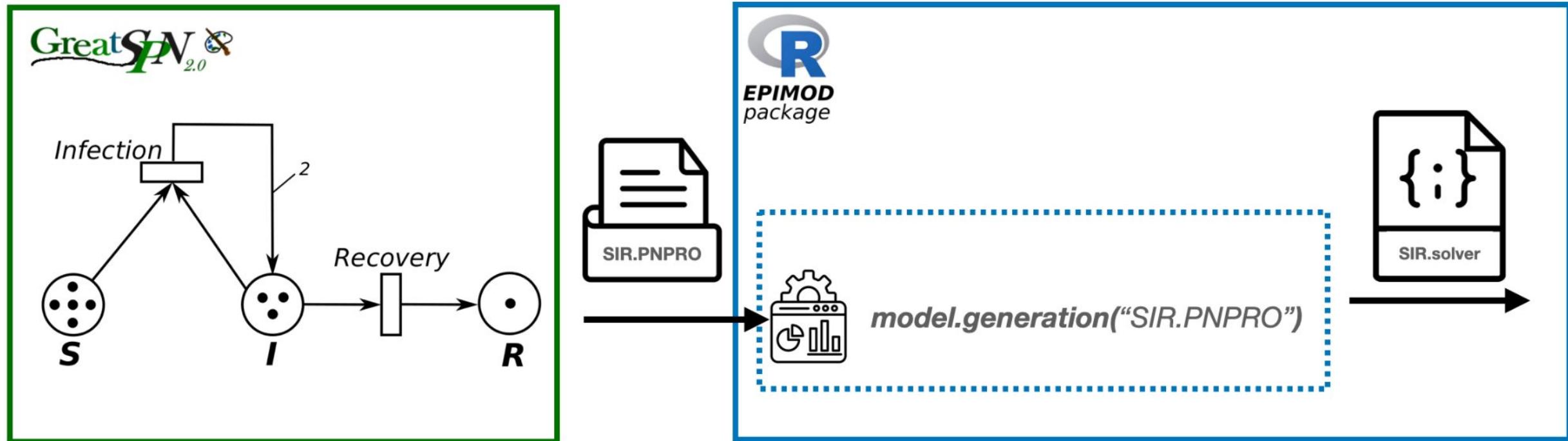
Ten Simple Rules for Reproducible Computational Research

1. For Every Result, Keep Track of How It Was Produced
2. **Avoid Manual Data Manipulation Steps**
3. Archive the Exact Versions of All External Programs Used
4. Version Control All Custom Scripts
5. Record All Intermediate Results, When Possible in Standardized Formats
6. For Analyses That Include Randomness, Note Underlying Random Seeds
7. Always Store Raw Data behind Plots
8. Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
9. Connect Textual Statements to Underlying Results
10. Provide Public Access to Scripts, Runs, and Results

Sandve et al. PLoS Comp Biol. 2013

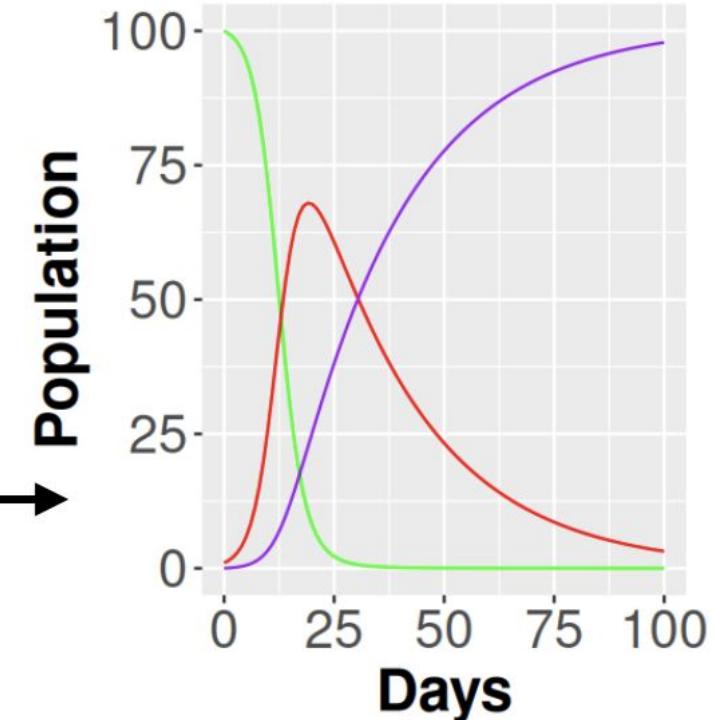
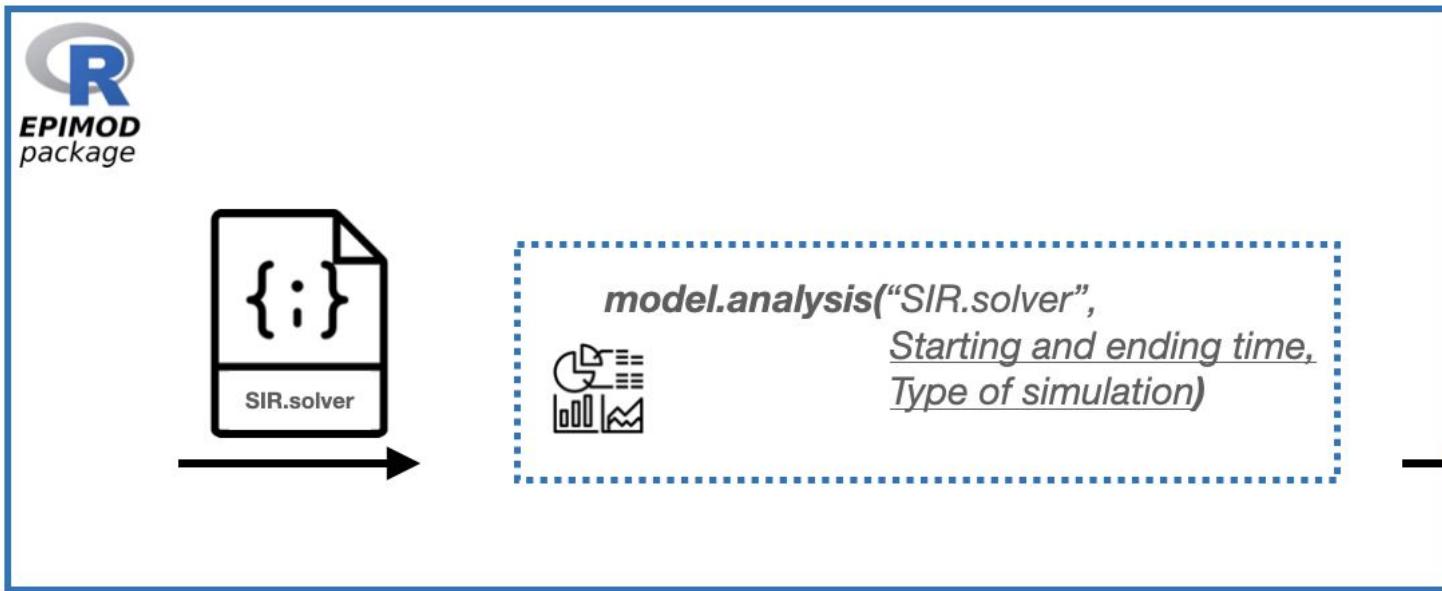
Model Generation

model_generation(): generate the implementation of the model (both deterministic and stochastic) from its graphical representation



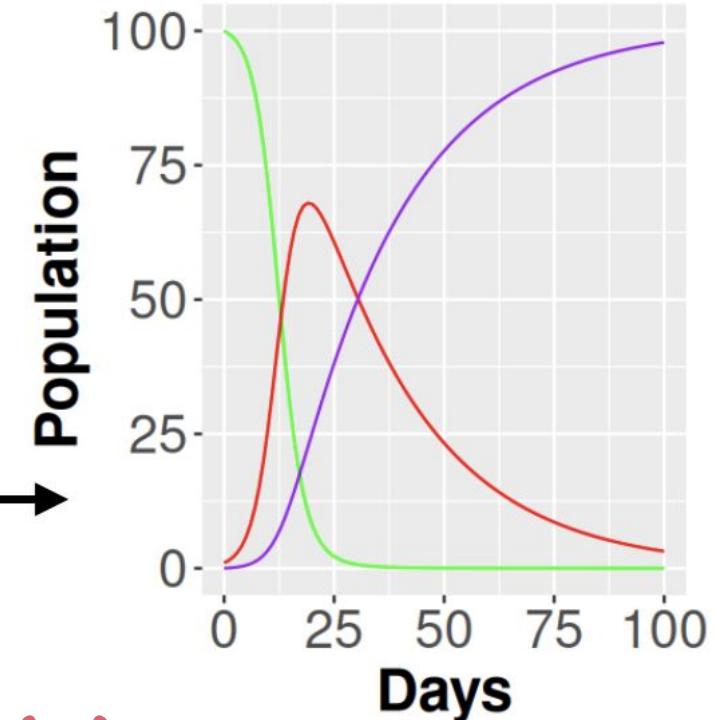
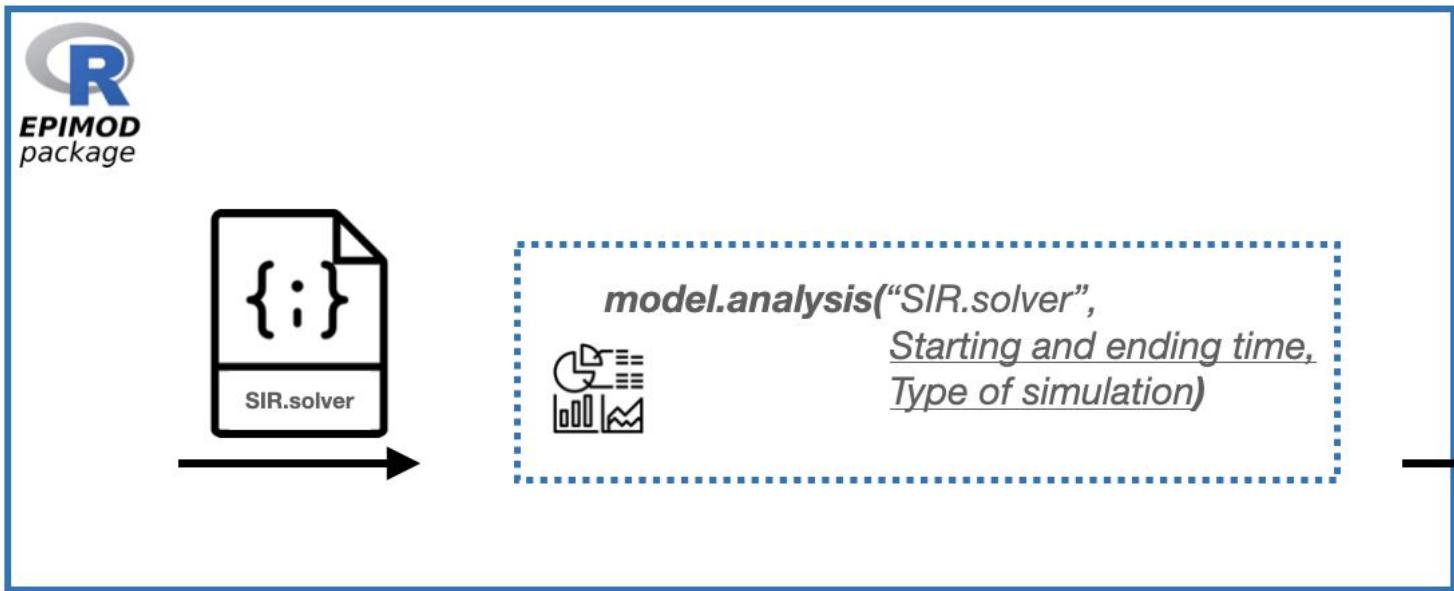
Model Analysis

model_analysis(): play with your brand new model



Model Analysis

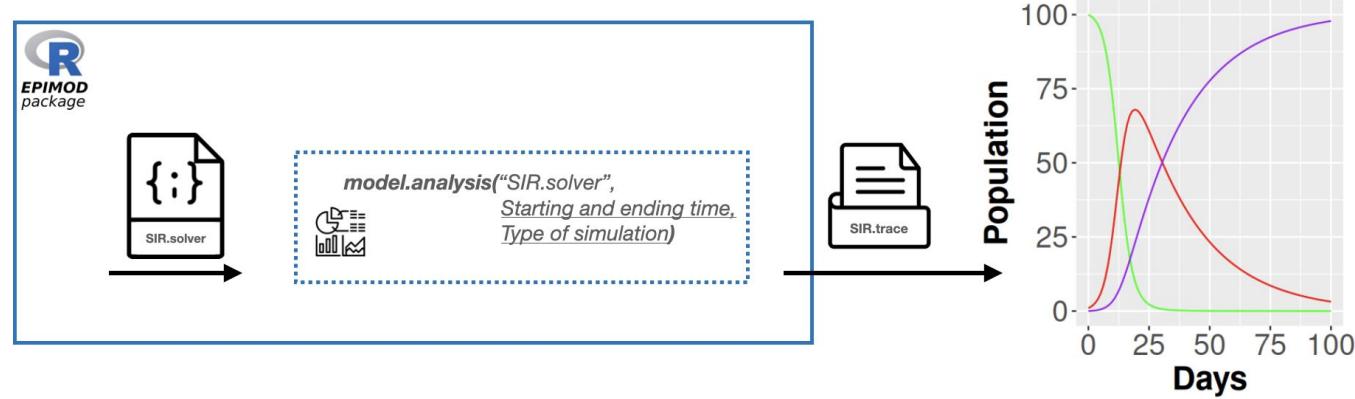
model_analysis(): play with your brand new model



The simulations depend on the parameters (initial state and transition rates) set from GreatSPN



Changing Model Parameters



The simulations depend on the parameters (initial state and transition rates) set from GreatSPN



Change parameters from GreatSPN

Change parameters from `model_analysis()`

Changing Model Parameters

The simulations depend on the parameters (initial state and transition rates) set from GreatSPN



Change parameters from *model_analysis()*

parameters_fname.csv

Describes what parameters vary without modifying the PNPRO file and how to define them

Tag & Meaning	Name	Value	Further Parameters
i= Complete initial marking	Usually <code>init</code>	Vector of numbers equal to the number of places or the name of a R function returning that vector	
m= Initial marking of a specific place	Name of the <code>place</code>	An integer number or the name of a R function returning a integer number	
c= Constant rate for a transition	Name of the <code>transition</code>	An real number or the name of a R function returning a real number	
g= Rate for a general transition	Name from <code>file in which will be saved the value</code>	An real number or the name of a R function returning a real number	input parameters needed by the R function, which can be an R built-in function or a <u>user-defined function</u>

functions_fname.R

Contains user-defined R functions used in `parameters_fname` and for analysis

Example	Purpose	Purpose	Notes
<code>init_generation</code>	Generates the initial marking values	<code>parameters_fname</code> (with tag i)	Initial conditions
<code>target</code>	Extracts the target values from model output (for PRCC calculation)	<code>target_value</code>	Selects one or more places to track over time.
<code>mse</code>	Computes the error/distance between model output and reference data	<code>distance_measure</code>	For instance Mean Squared Error

Changing Model Parameters



parameters_fname.csv

Describes what parameters vary without modifying the PNPRO file and how to define them

Tag & Meaning	Name	Value	Further Parameters
i= Complete initial marking	Usually init	Vector of numbers equal to the number of places or the name of a R function returning that vector	
m= Initial marking of a specific place	Name of the place	An integer number or the name of a R function returning a integer number	
c= Constant rate for a transition	Name of the transition	An real number or the name of a R function returning a real number	
g= Rate for a general transition	Name from file in which will be saved the value	An real number or the name of a R function returning a real number	input parameters needed by the R function, which can be an R built-in function or a user-defined function

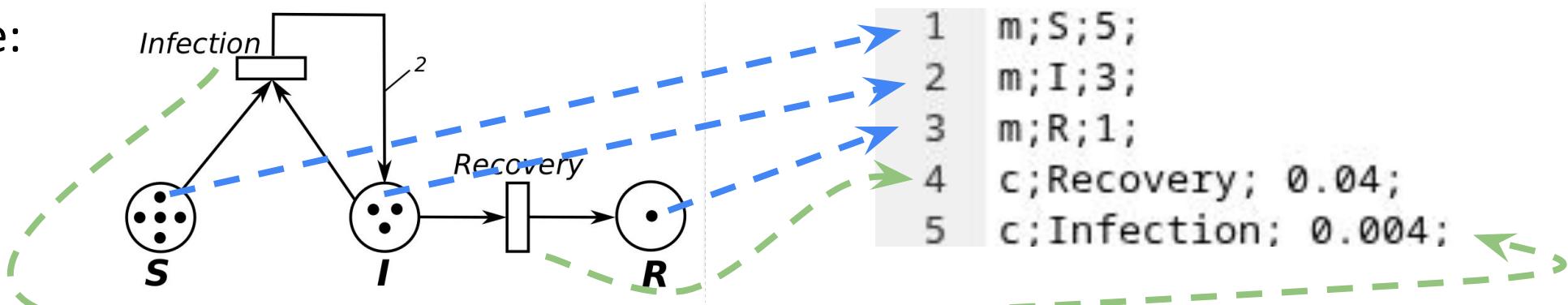


functions_fname.R

Contains user-defined R functions used in parameters_fname and for analysis

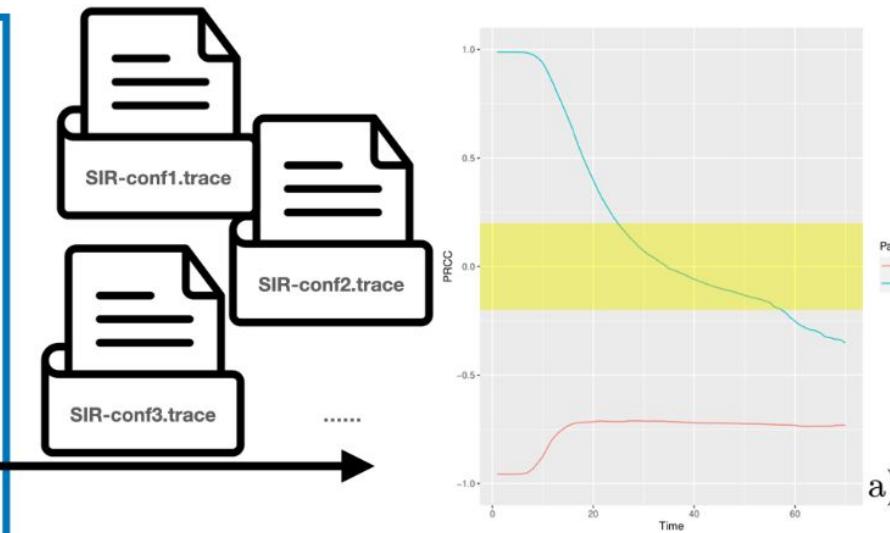
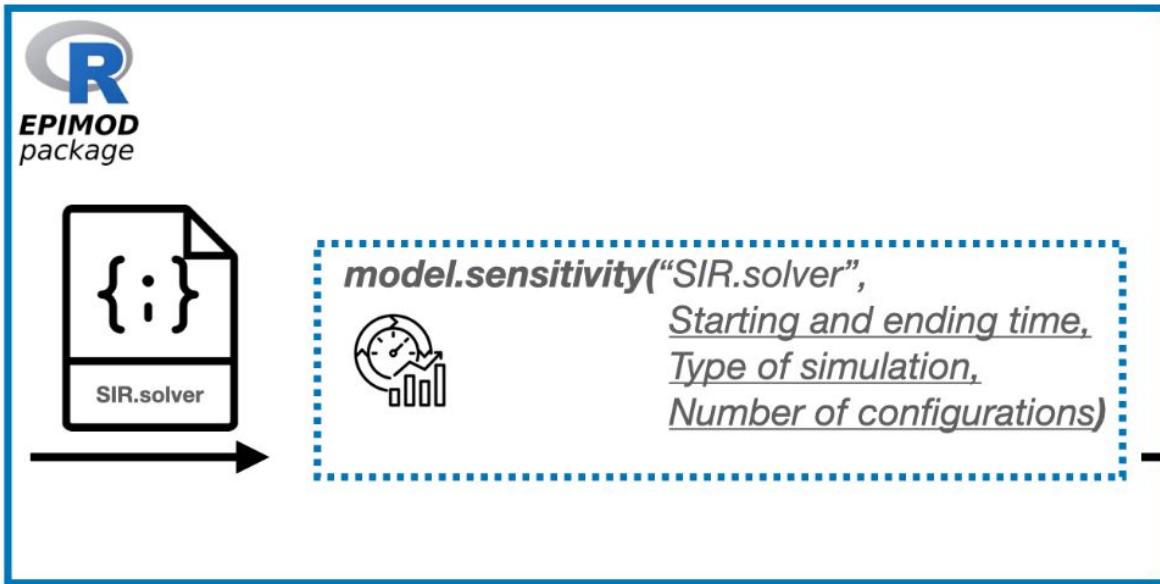
Example	Purpose	Purpose	Notes
init_generation	Generates the initial marking values	parameters_fname (with tag i)	Initial conditions
target	Extracts the target values from model output (for PRCC calculation)	target_value	Selects one or more places to track over time.
mse	Computes the error/ distance between model output and reference data	distance_measure	For instance Mean Squared Error

Simple example:

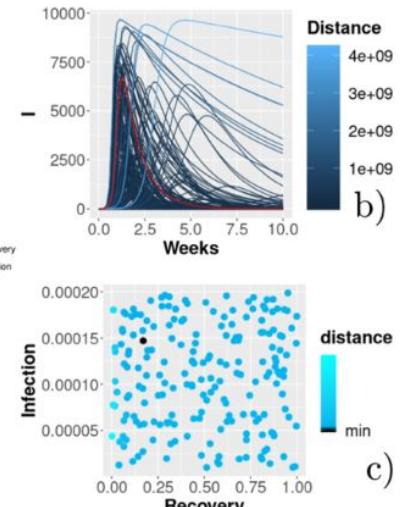


Model Sensitivity

model_sensitivity(): identify the set of parameter that impact the most on the model outcome (dynamics) with the Partial Rank Correlation Coefficients



a) Study the input parameters through the PRCC.

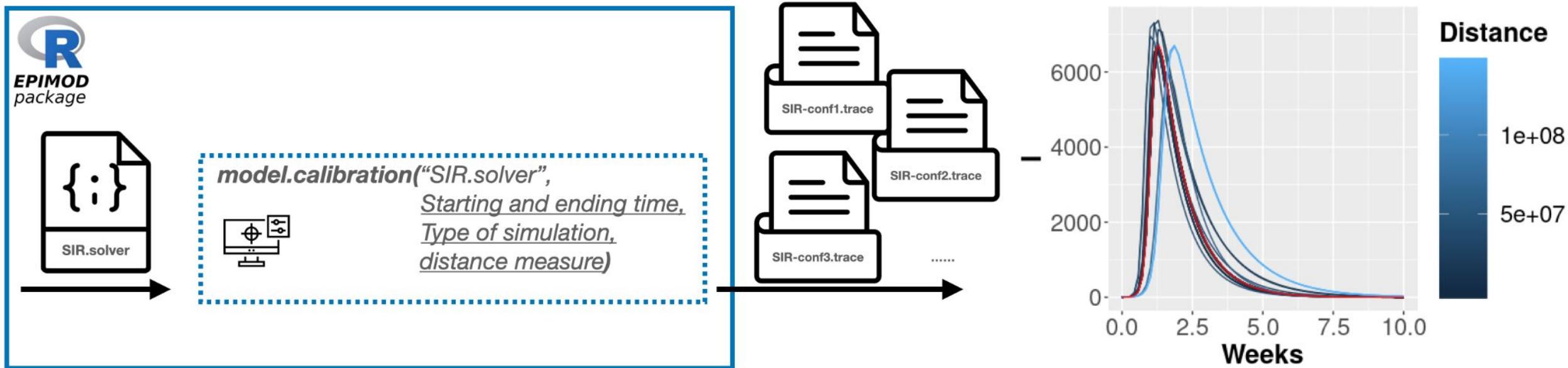


b-c) Study the parameter search space with respect to the reference data

Model Calibration

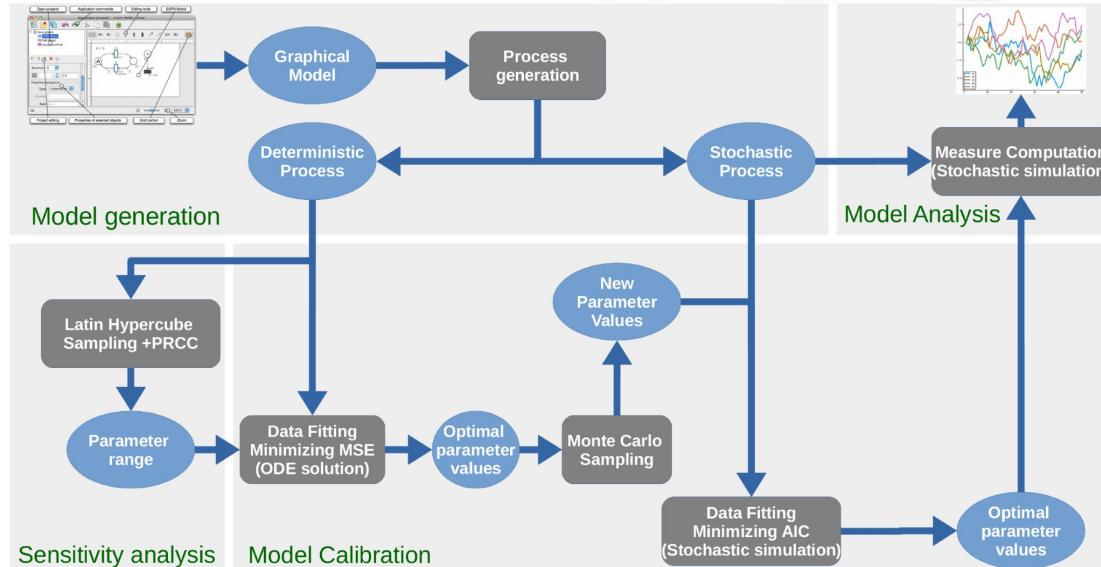
model_calibration(): estimate the parameters configuration which fit the best a given reference data.

Minimise the distance between the reference data and the model's output according to a given (user provided) distance metric.

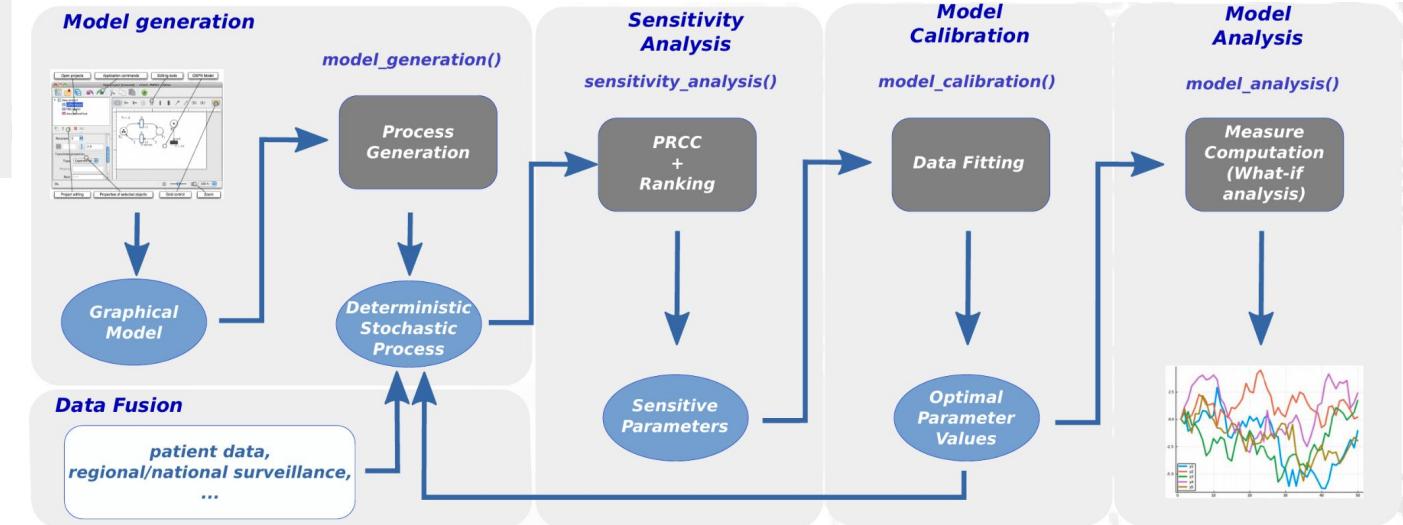


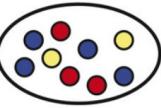
Epimod is modular

Analysis pipeline for Pertussis model

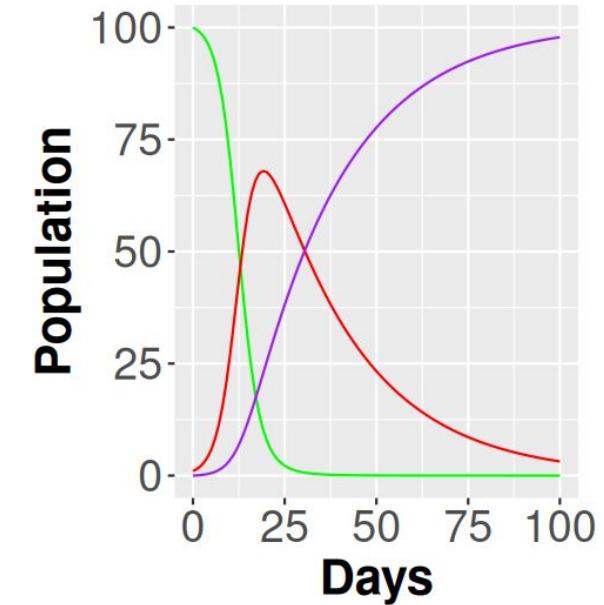
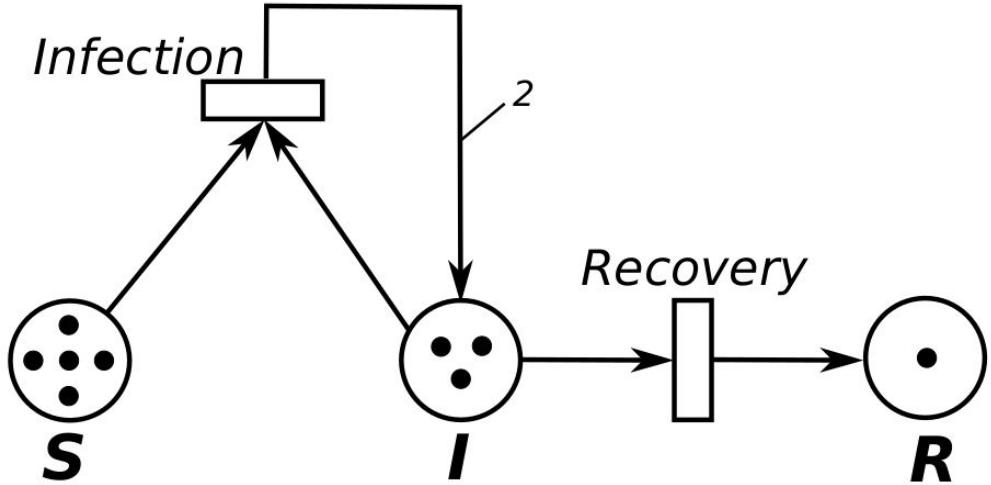


Analysis pipeline for COVID-19 model





Case study: SIR model

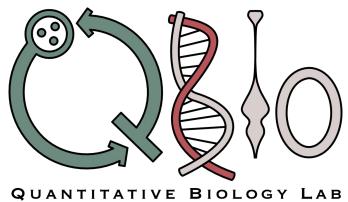




UNIVERSITÀ
DI TORINO

DIPARTIMENTO
DI INFORMATICA

di.unito.it

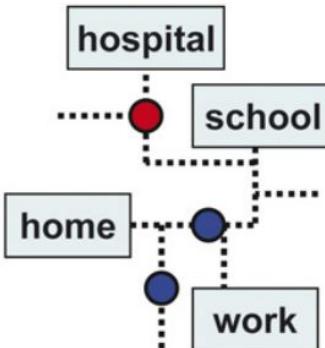


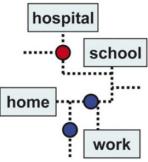
Young
InfoLife
cini national lab

IEEE
I_HI
2025

IEEE - International Conference on
Healthcare Informatics
Università della Calabria, Italy
June 18th-21st, 2025

Micro-level models





Micro-level models

The second major class of modeling approaches: **Micro-Level Models**, often called **agent-based models (ABMs)**.

ABMs have two main components: agents and the environment. **Agents** represent individual people, each following their own rules, behaviors, and decision-making processes rather than being grouped. The **environment** provides the specific space or setting where these agents interact and carry out their decisions.

Some key features of these models include:

- Each agent can have **different** characteristics.
- Interactions depend on **who meets whom, when, and where**.
- Agents can move, change behavior, and react to their **environment**.
- From these individual interactions, **complex global patterns**, like waves of infection or clustering, can emerge over time.

The strength of these models lies in their **realism**. They help explore questions like:

- *What if some people are super-spreaders?*
- *How do contact patterns affect the outbreak?*
- *Can we target specific groups or locations to slow the spread?*



We simulate interactions in detail, like a crowd of individuals, each making **their own decisions**.



While they're **more complex and slower to run**, they're excellent for understanding detailed dynamics.

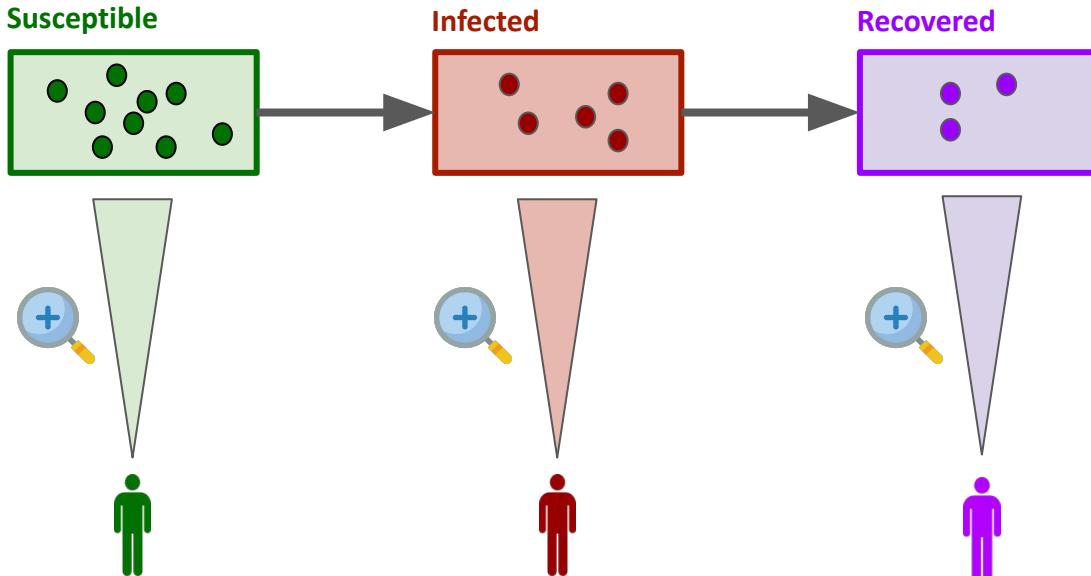
Agent-based models in infectious disease modeling



We simulate interactions in detail, like a crowd of individuals, each making **their own decisions**.



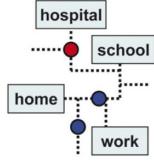
While they're **more complex and slower to run**, they're **excellent for understanding detailed dynamics**.

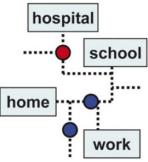


Agents can be **Susceptible (S)**, **Infected (I)**, or **Recovered (R)**, just like in the SIR model, but **interactions are local**, not global.

Unlike compartment models...

- **No assumption of perfect mixing.**
- Infection spreads through **local contact and shared spaces**, e.g., in neighborhoods, households, or workplaces.
- The outcomes are **stochastic** (random), and we often run simulations **many times** to explore possibilities.

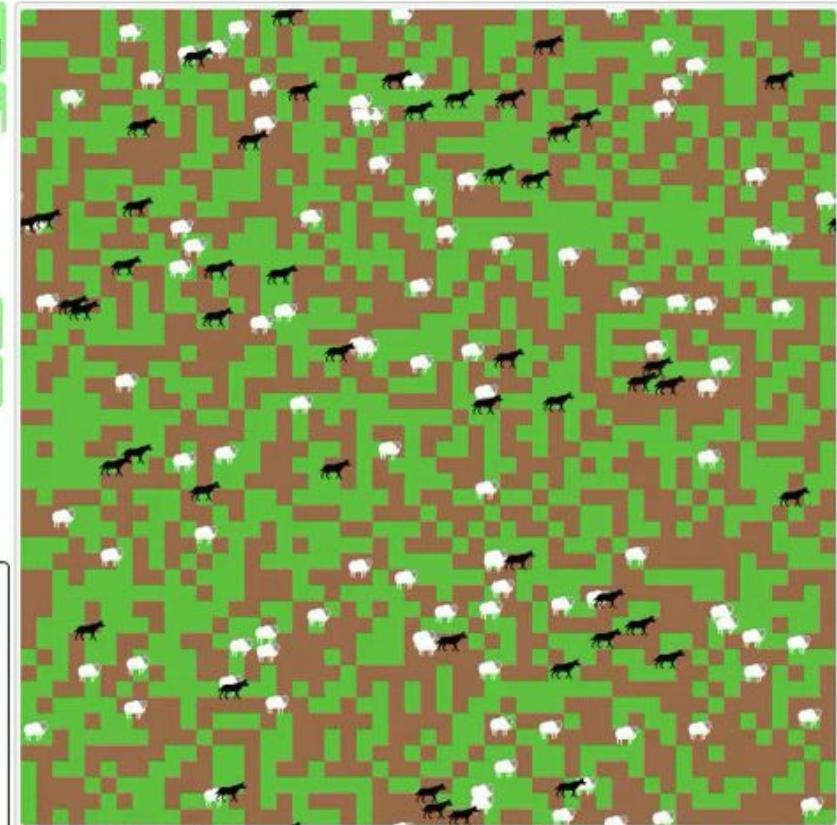
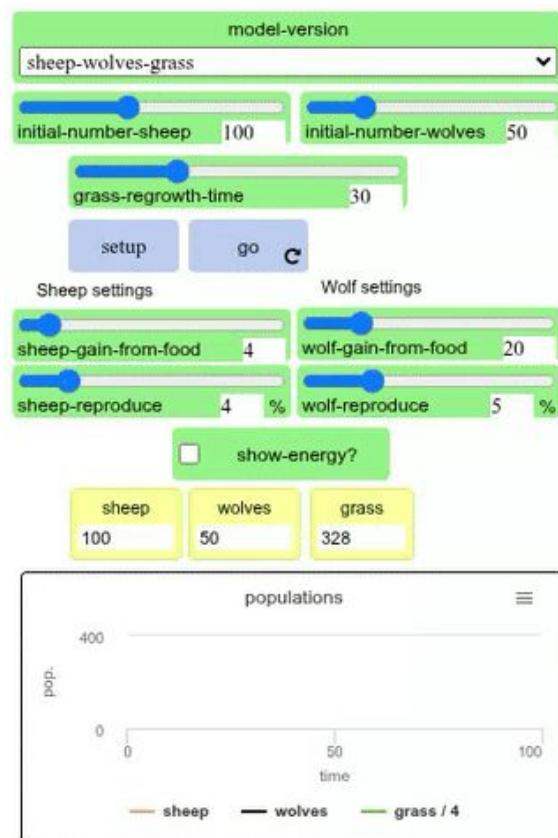




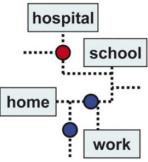
ABM environment

Spatial (grid or continuous space):

- Agents move across physical spaces (e.g., cities, maps).
- Supports location-based interactions.
- Examples: predator-prey models, traffic flow, school.

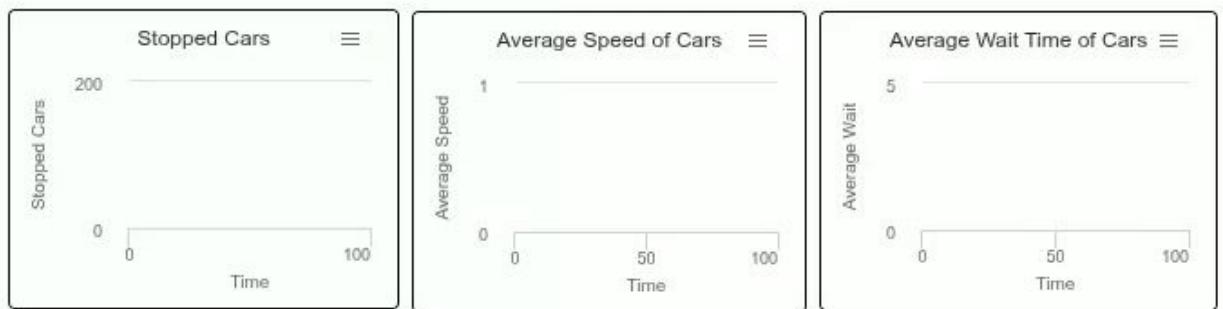
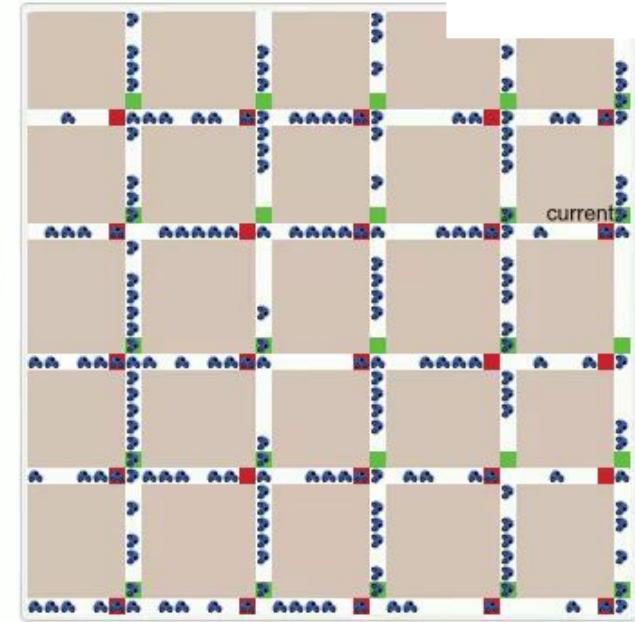
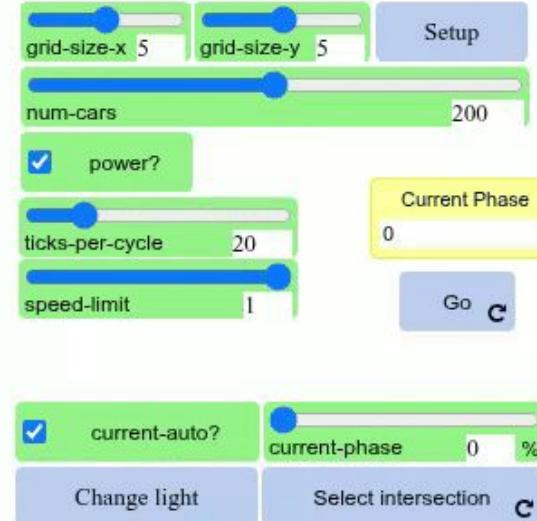


From <https://ccl.northwestern.edu/netlogo/models/>

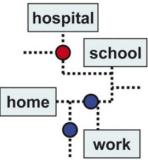


ABM environment

Traffic Grid

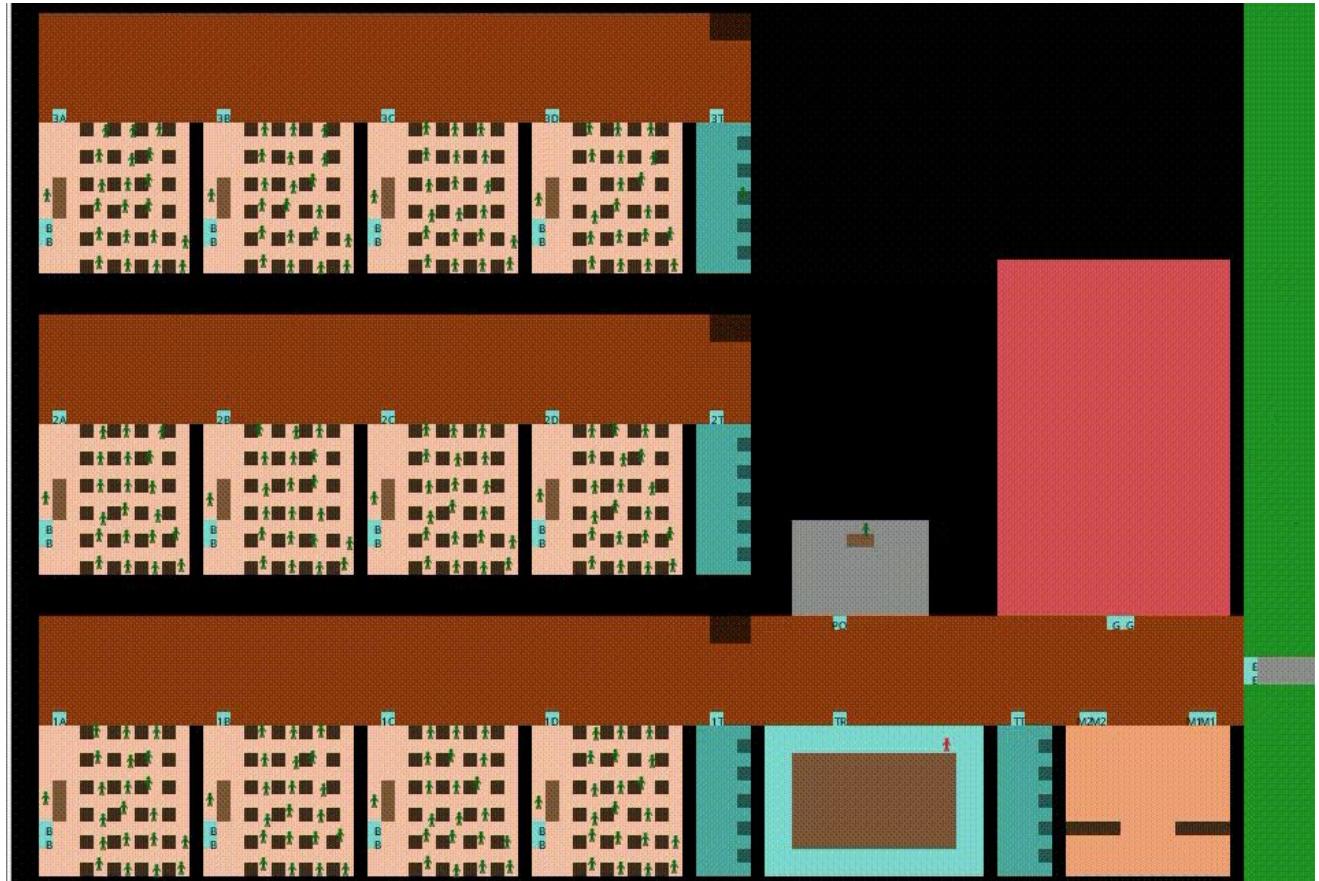


From <https://ccl.northwestern.edu/netlogo/models/>



ABM environment

School model

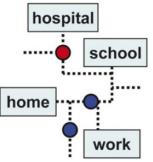


Spatial (grid or continuous space):

- Agents move across physical spaces (e.g., cities, maps).
- Supports location-based interactions.
- Examples: predator-prey models, traffic flow, school.

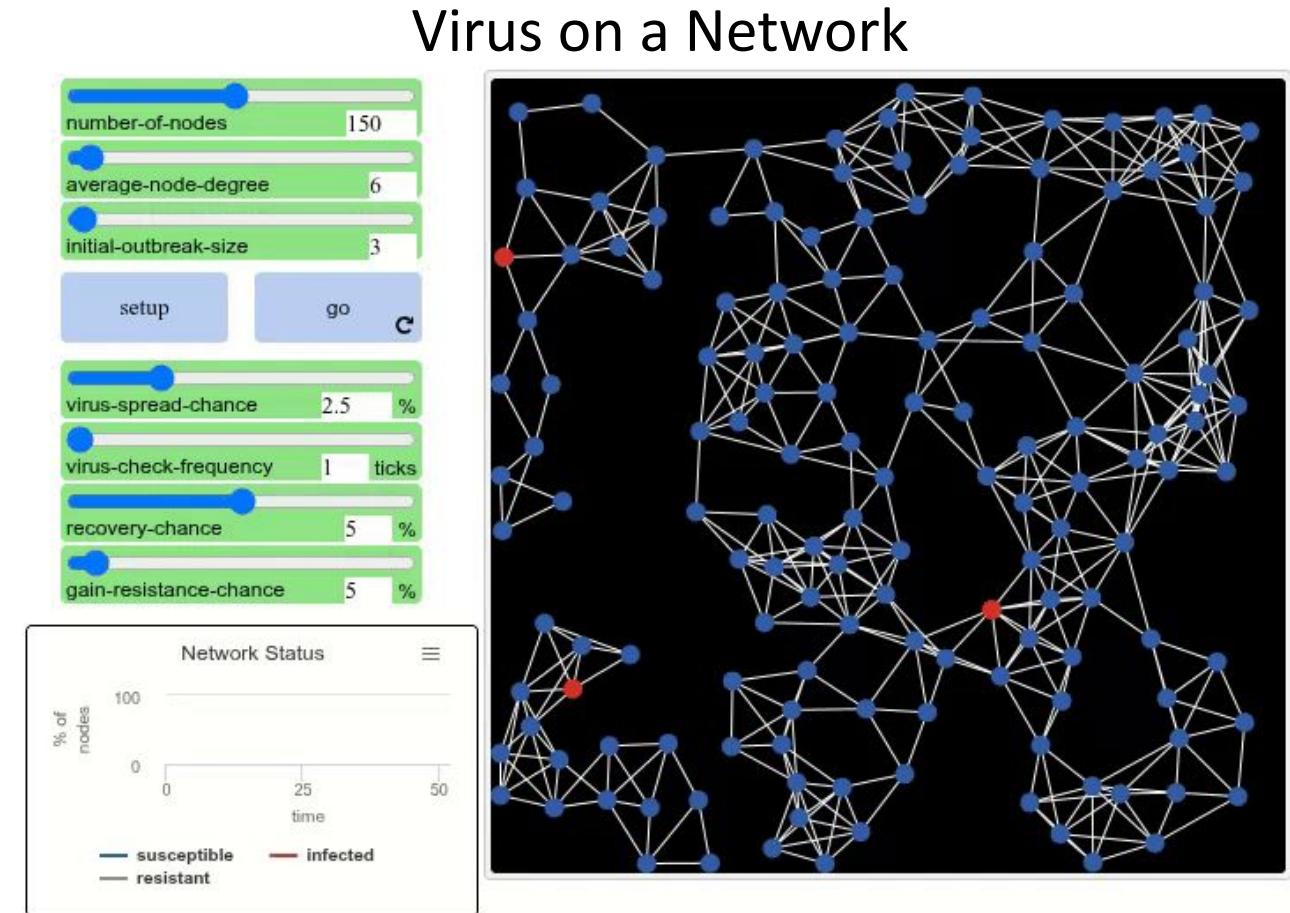
Baccega, Daniele, et al. "An agent-based model to support infection control strategies at school." *JASSS* 25.3 (2022): 1-15.
 Baccega, Daniele, et al. "Living along COVID-19: assessing contention policies through Agent-Based Models." *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer, Cham, 2025.

ABM environment

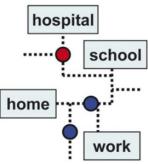


Network-based:

- Agents are nodes connected by relationships (edges).
- Useful for modeling social influence, disease spread, communication.
- Examples: social networks, power grids.



From <https://ccl.northwestern.edu/netlogo/models/>



ABM environment

Abstract or conceptual:

- Represents non-physical dimensions (e.g., markets, belief systems).
- Agents interact based on logical conditions or data.
- Examples: bidding market, voting.

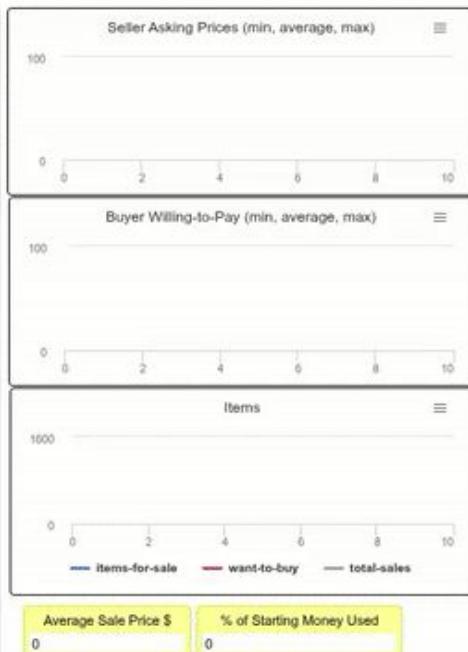
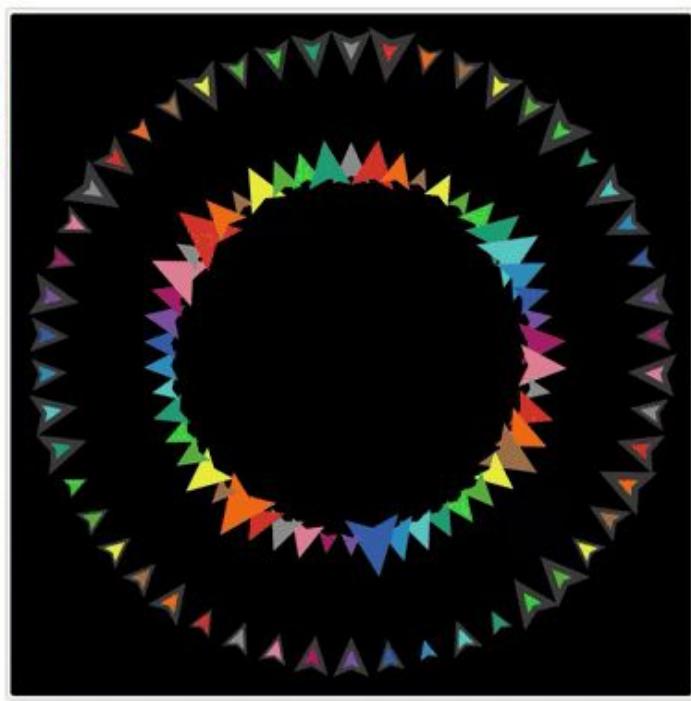
starting-money 500

supply-amount high supply-distribution even

demand-amount high demand-distribution even

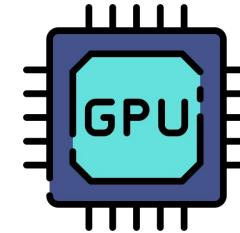
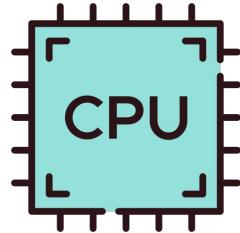
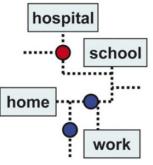
seller-behavior random buyer-behavior random

sellers-ignore-full-buyers? stars?



From <https://ccl.northwestern.edu/netlogo/models/>

ABM tools



[NetLogo](#)



{
[Repast Simphony](#)
[Repast HPC](#)

[Mason](#)



[FLAME](#)

...



[FLAME GPU 2](#)



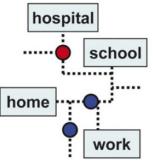
[BioDynaMo](#)

[GEMSim](#)

[SIMCoV-GPU](#)

...

NetLogo

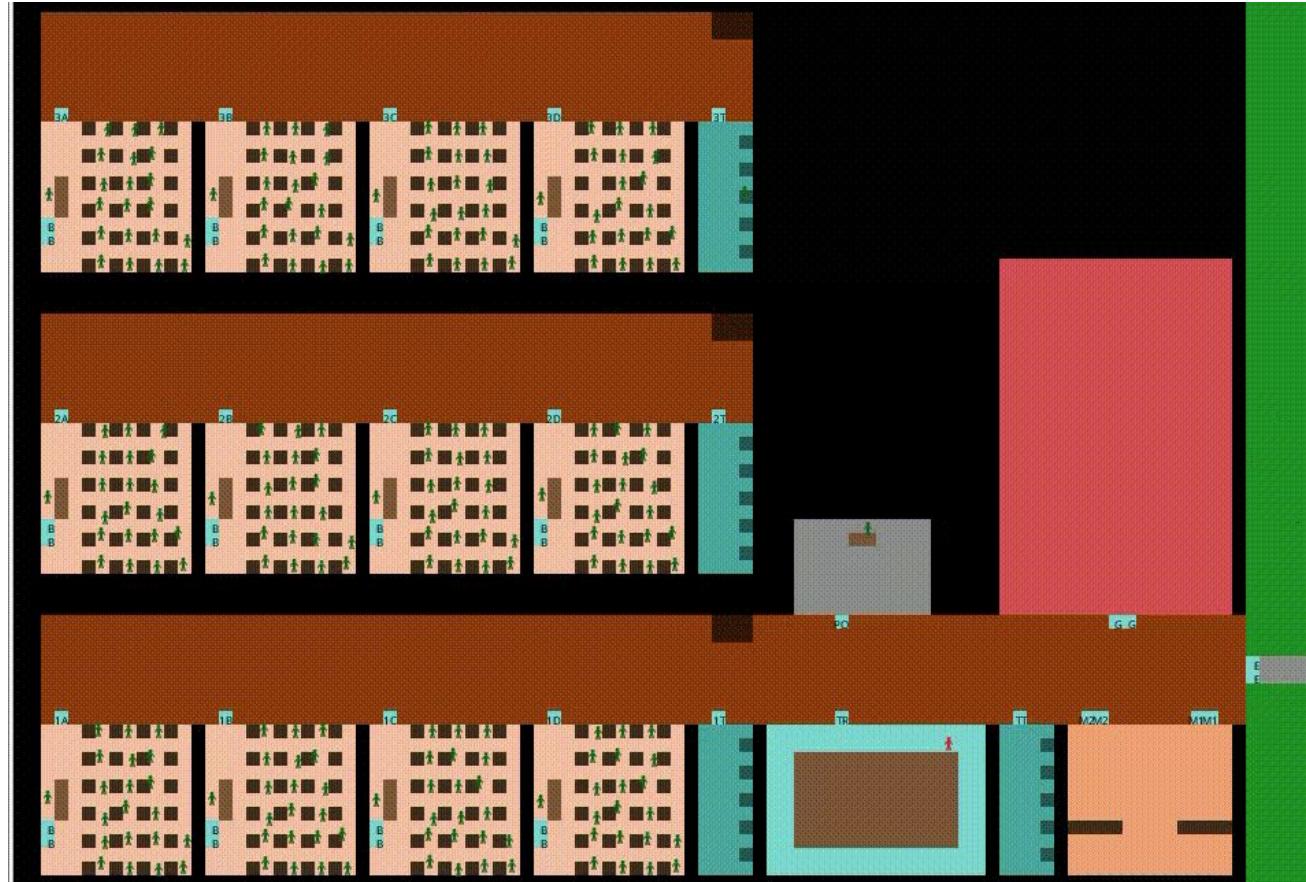


What is NetLogo?

NetLogo is a simulation environment for building and exploring agent-based models, designed to run on the CPU.

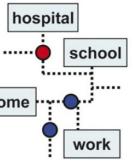
Why use it?

- Simulate real-world systems (e.g., ecosystems, traffic, markets).
- Visual + interactive = great for learning & experimentation.
- Easy to start, powerful to scale up.



Baccega, Daniele, et al. "An agent-based model to support infection control strategies at school." JASSS 25.3 (2022): 1-15.
Baccega, Daniele, et al. "Living along COVID-19: assessing contention policies through Agent-Based Models." International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics. Springer, Cham, 2025.

NetLogo

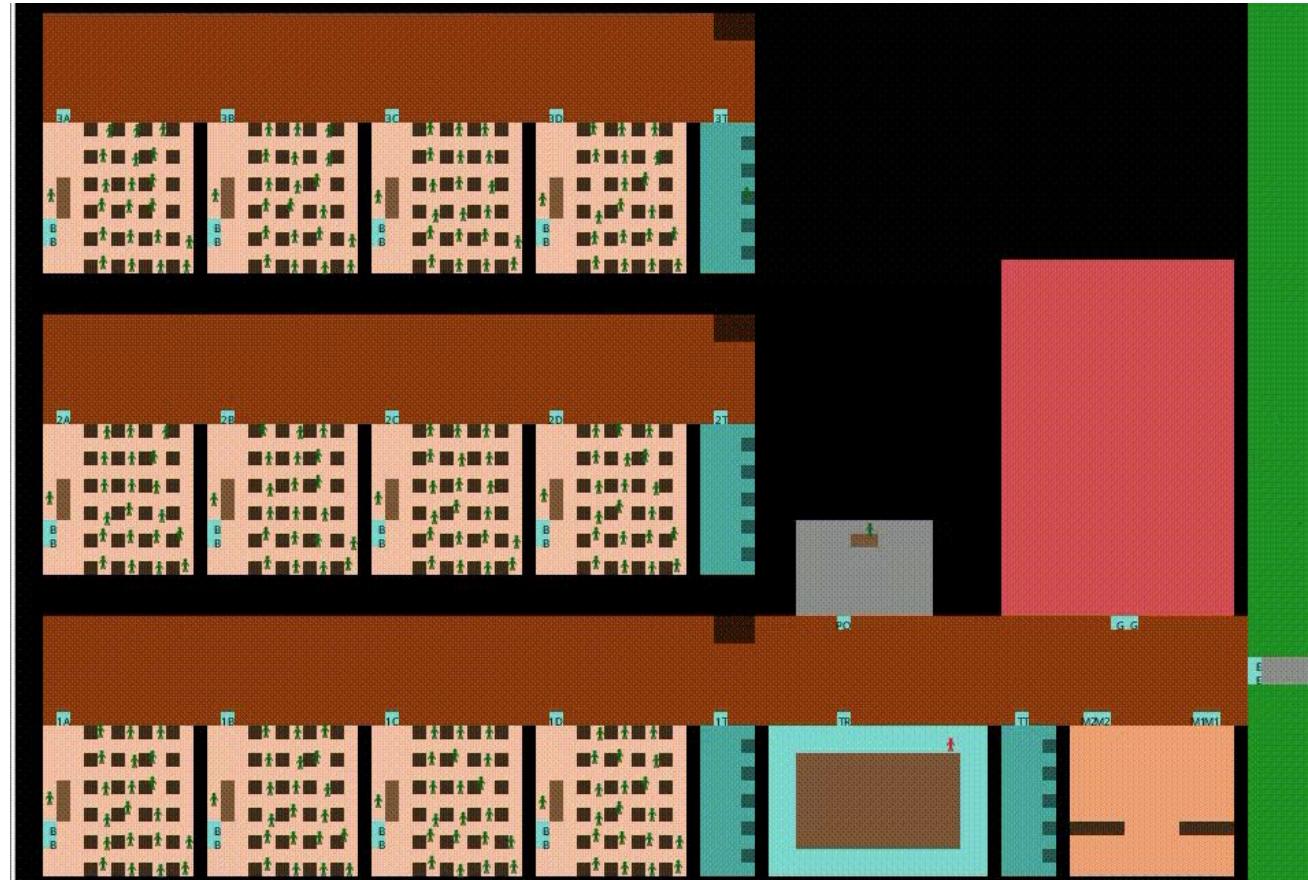


✓ Pros

- Easy to learn, beginner-friendly.
- Visual feedback for real-time learning.
- Huge built-in model library.
- Supports complex behavior with simple code.
- Cross-platform and free.

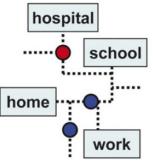
✗ Cons

- Slower with very large agent counts.
- No parallelization.
- Not ideal for heavy numerical computations.
- Scripting language unique to NetLogo (not general-purpose).



Baccega, Daniele, et al. "An agent-based model to support infection control strategies at school." JASSS 25.3 (2022): 1-15.

Baccega, Daniele, et al. "Living along COVID-19: assessing contention policies through Agent-Based Models." International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics. Springer, Cham, 2025.

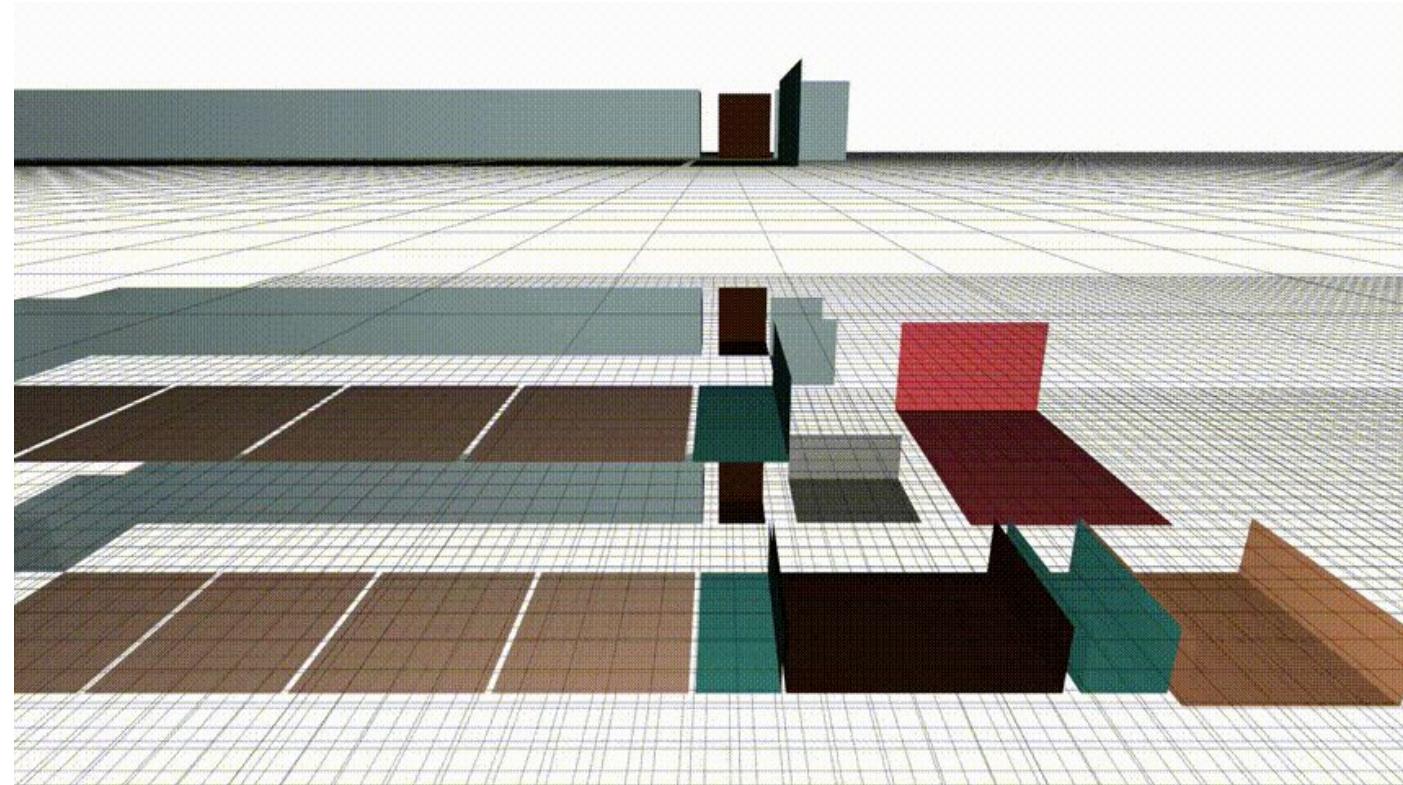


What is FLAME GPU 2?

FLAME GPU 2 is a high-performance agent-based modeling framework designed for GPU acceleration using NVIDIA CUDA.

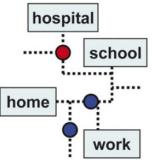
Why use it?

- Ideal for large-scale, computationally intensive models.
- Massively parallel: leverages the GPU to simulate millions of agents.
- Great for fields like biology, transportation, crowd simulation, and epidemiology.



Baccega, D., et al. "Forge4Flame: An Intuitive Dashboard for Designing GPU Agent-Based Models to Simulate Infectious Disease Spread." Available at SSRN 5194584.

FLAME GPU 2

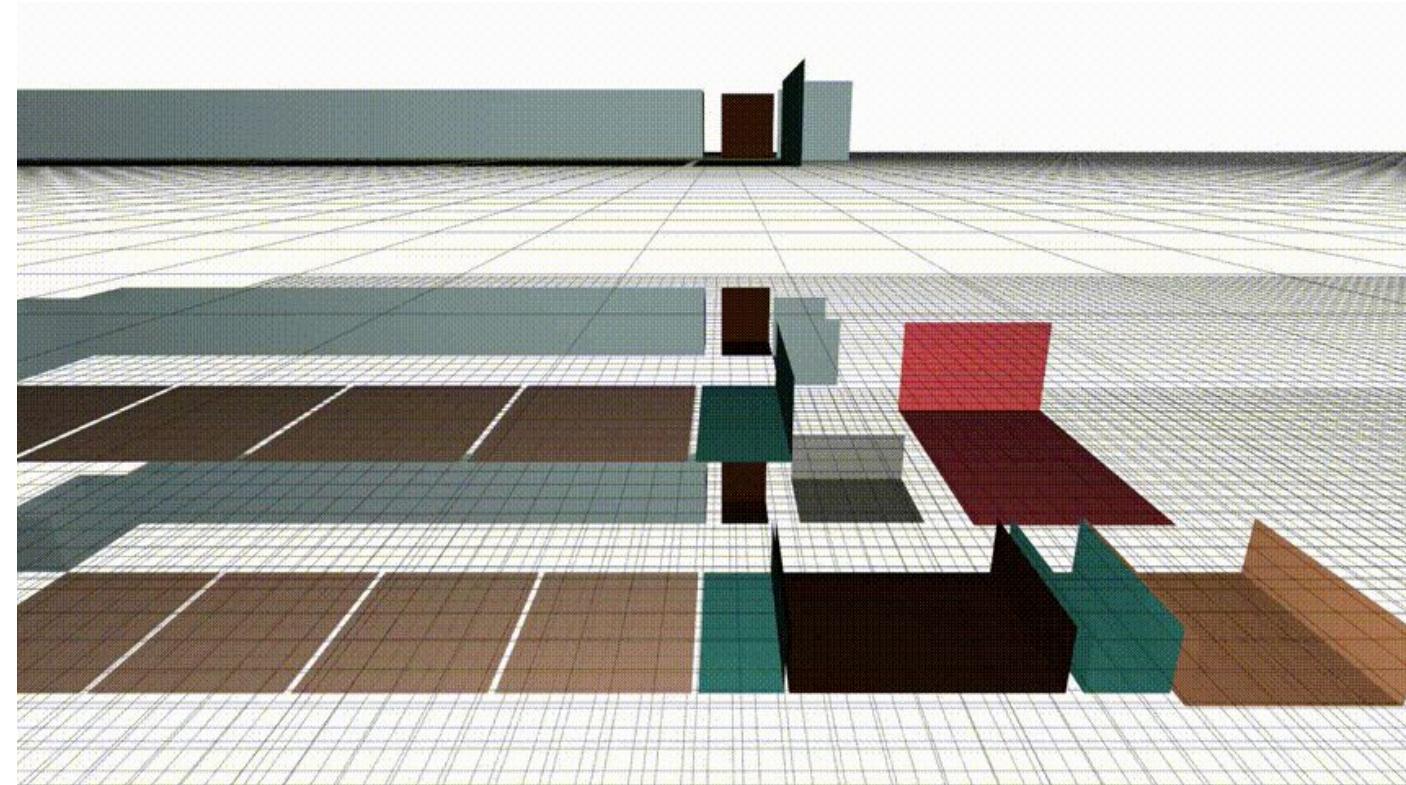


✓ Pros

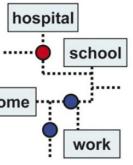
- Extreme scalability: handles millions of agents.
- GPU acceleration = faster simulations.
- Custom C++ agent logic.
- Integration with Python.

✗ Cons

- Steeper learning curve than GUI-based tools.
- Requires CUDA-capable GPU and some C++ knowledge.
- Less beginner-friendly for quick prototyping.
- Limited built-in visualization (compared to NetLogo).

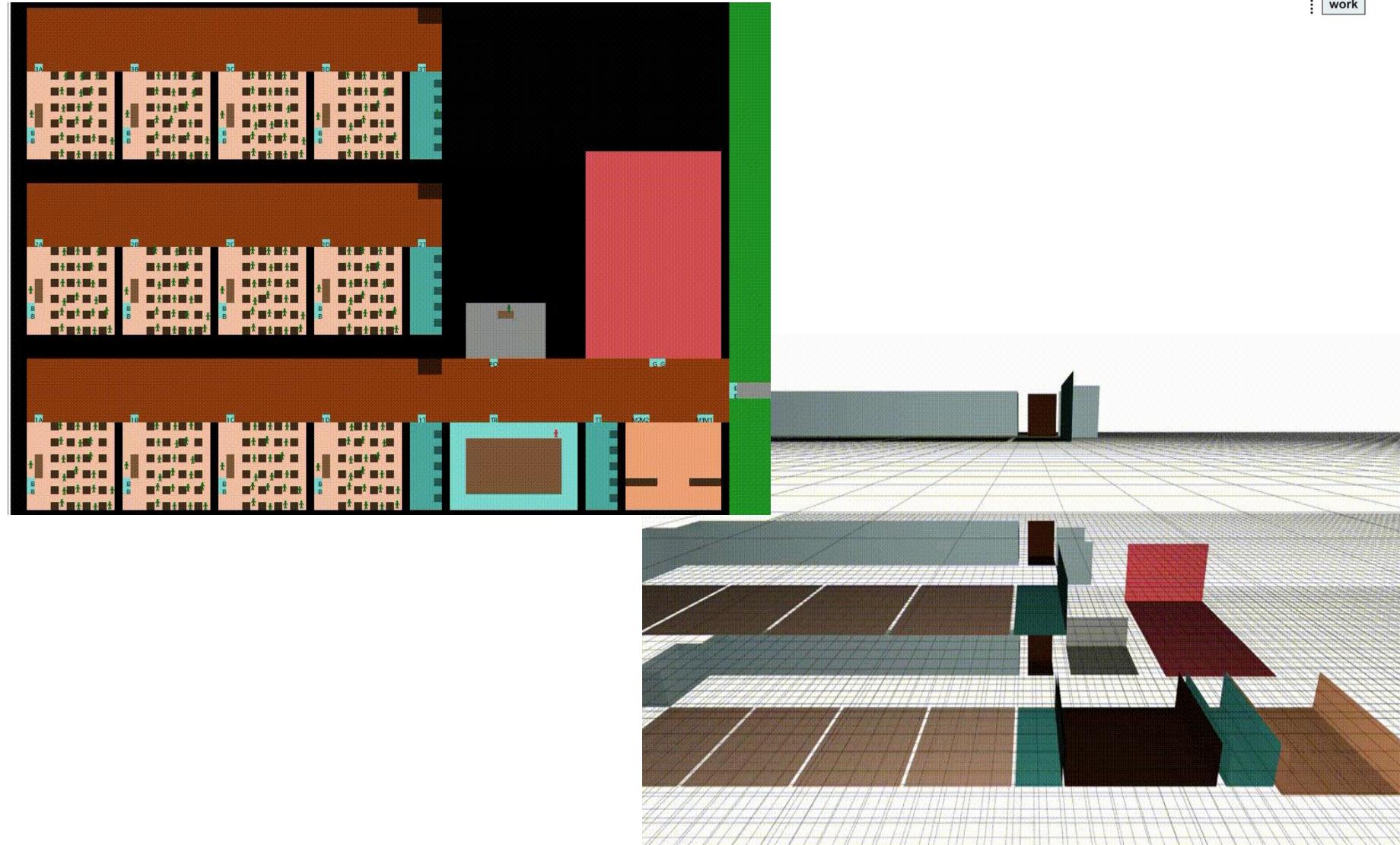


Baccega, D., et al. "Forge4Flame: An Intuitive Dashboard for Designing GPU Agent-Based Models to Simulate Infectious Disease Spread." Available at SSRN 5194584.



Well-defined closed environment

- These scenarios feature detailed simulation of agents' movements within a confined and structured space.
- **Very complex** to define.
- Examples: school, hospital, supermarket, office.



Forge4Flame: An Intuitive Dashboard for Designing GPU Agent-Based Models to Simulate Infectious Disease Spread



Baccega, D., et al. "Forge4Flame: An Intuitive Dashboard for Designing GPU Agent-Based Models to Simulate Infectious Disease Spread." Available at SSRN 5194584.

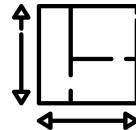


Forge4Flame: introduction

User-friendly dashboard designed to simplify the definition of an **Agent-Based Model environment** without the need to write code.

It enables to define:

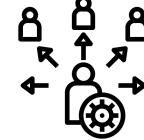
1. model's environment



2. the agents interacting within it



3. the disease model



4. other components relevant to an agent-based simulation of disease spread



5. Simple visualisations



User-friendly

Approachable by scientists without specific computational knowledge



Automatic

Automatic generation of 3D model environment for FLAME GPU 2



Reproducibility

Scientific results can be consistently replicated by independent researchers using the same methodology and data



How it started...



TrustAlert

Platform for the analysis of near real-time stream of healthcare data

Founded by Compagnia
San Paolo

1M Euro Project
7 Collaborators



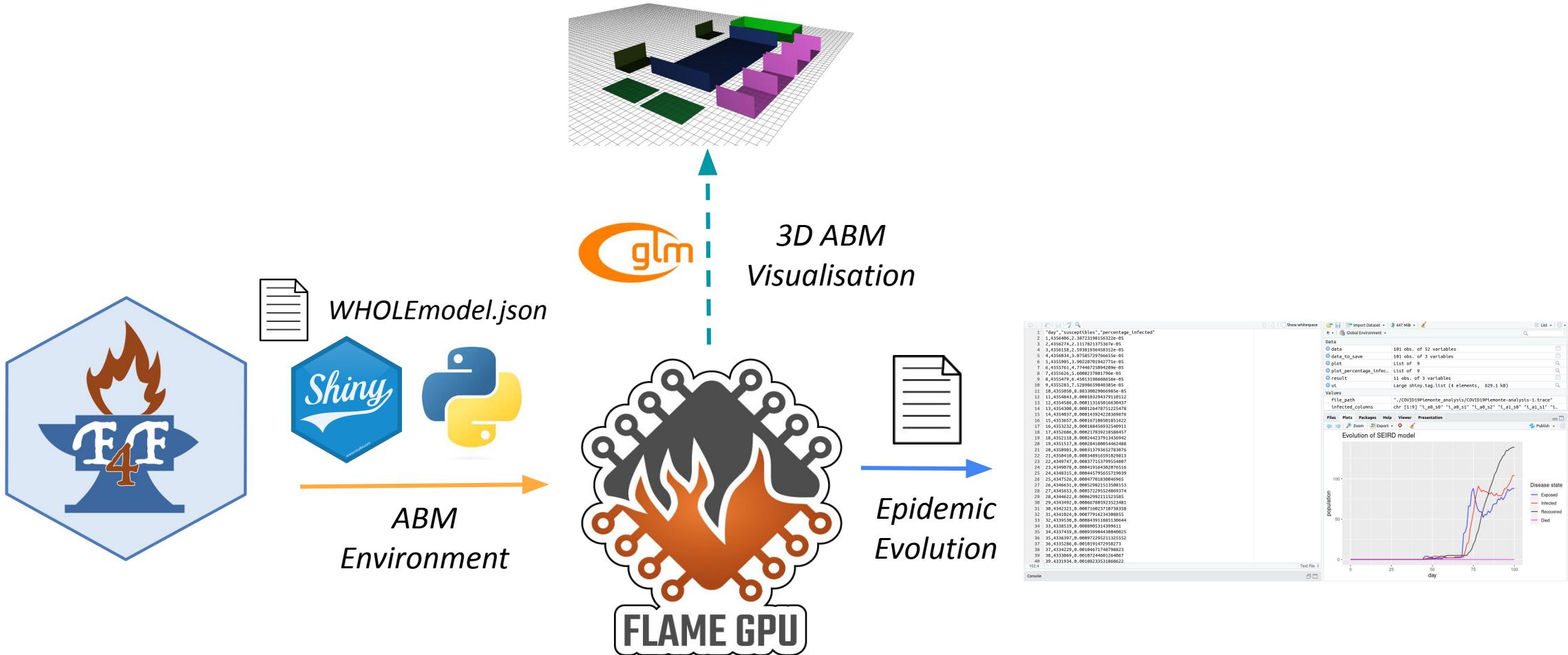
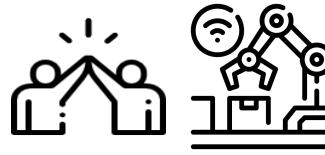
Our objectives in this project:

Evaluate the potential impacts of a health emergency in a real case scenario (Cottolengo Hospital in Turin) at different levels of detail, spanning from **different simulation paradigms** (e.g., ABM).

Expected results:

Being able to simulate the behavior of the "Hospital" system, for what happens outside, will allow Cottolengo Hospital to be able, in the medium term, not only to be effective but to improve the efficiency (cost / result ratio) and response capacity to ensure the delivery of health services.

Pipeline

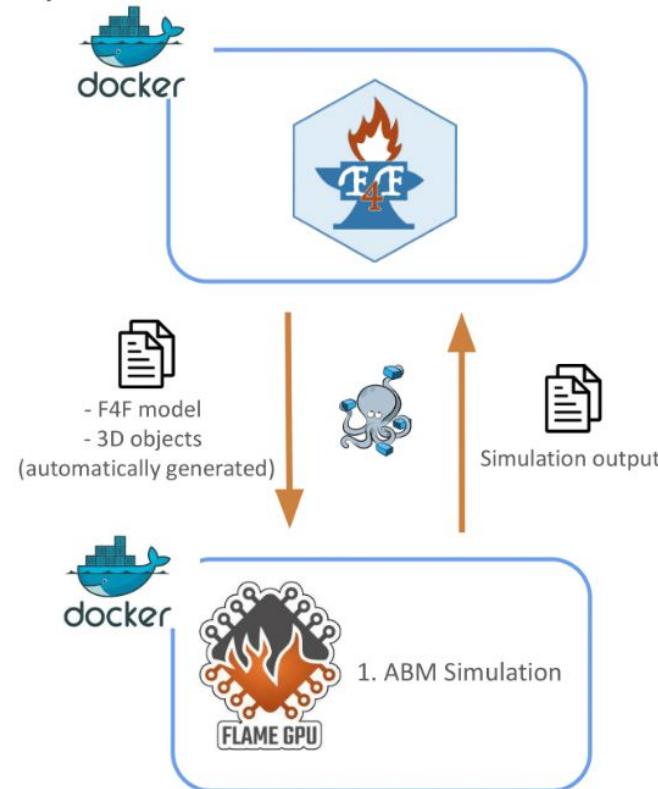




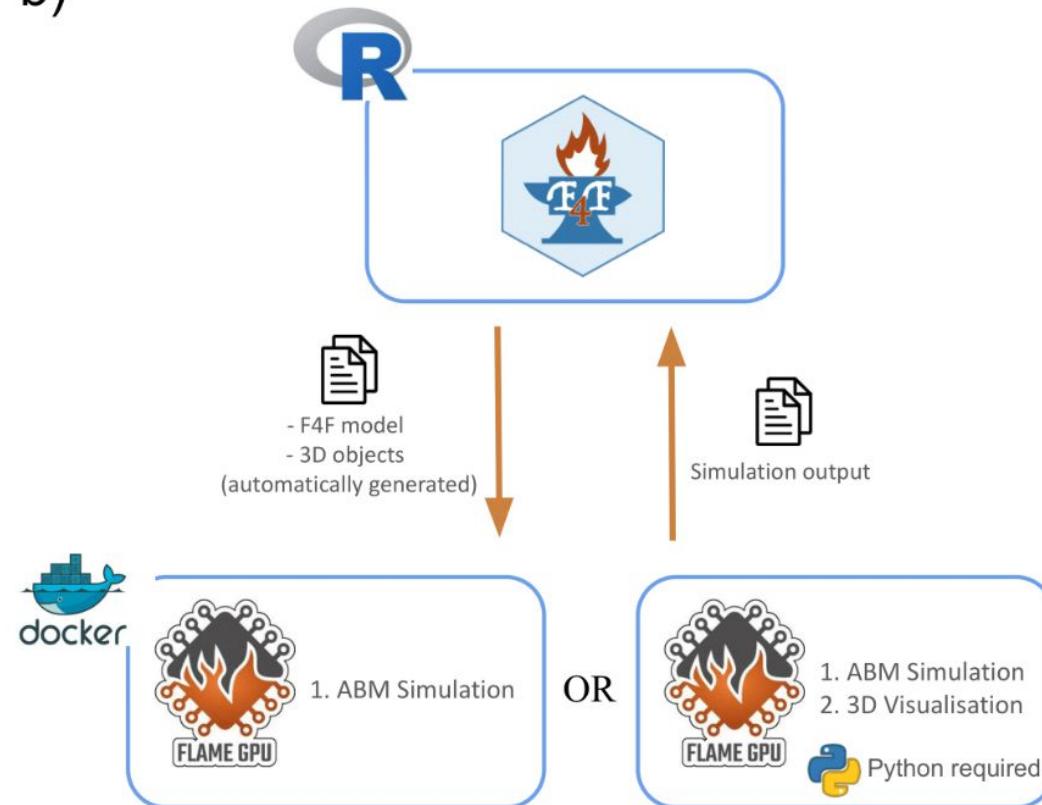
Docker pipeline



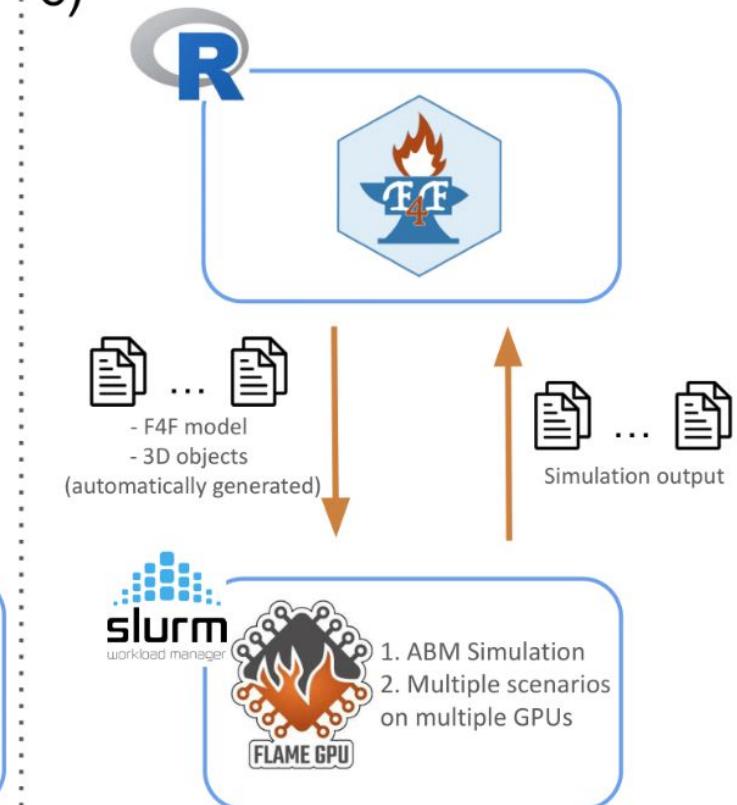
a)



b)

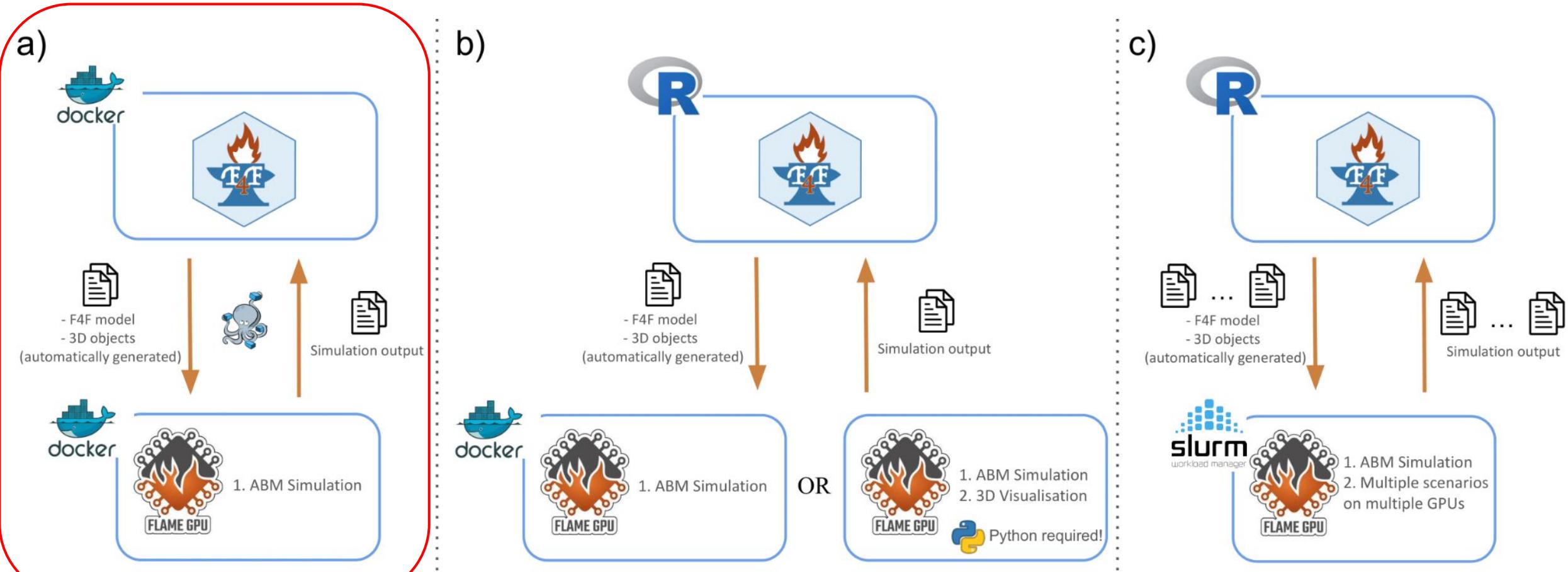


c)





Docker pipeline





Homepage

Build your ABM



Home

Canvas

Rooms

Agents

Resources

Infection

What-if

Configuration

Run

Settings

Post Processing



Forge4Flame (F4F)

F4F is a user-friendly dashboard (developed in R Shiny) designed to simplify the definition of an ABM environment for FLAME GPU 2 [5, 6] agent-based models, automatically generating the necessary code. It enables users to define the model's environment, the agents interacting within it, the disease model, and other components relevant to an ABM simulation. F4F is constituted by the following components (the images refer to the school model defined in [7, 8]):

- **Canvas:** define the model's environment using a drag-and-drop interface for rooms.

Define floor

Define>Select floor

floor0

Delete floor

Drag the floors in the desired order

floor0

floor1

floor2

Add elements

Draw rooms

Select the room:

Length of selected room (length refers to the wall with the door):

Width of selected room:

Height of selected room:

Door position:

Remove rooms

Select the room to remove:

Remove room

Clear floor

Agents management





How to install and run



Forge4Flame (F4F)

Forge4Flame (F4F) is a user-friendly R Shiny dashboard designed to streamline the creation of FLAME GPU 2 [1, 2] ABMs, automatically generating the necessary code. With its intuitive interface, users can easily define key components, including the model's environment, the agents interacting within it, the disease model, and other essential elements for agent-based simulations.

See [user guide](#) for more information.

How to download

Repository

Cloning the repository is optional, especially for users intending to use the F4F package, dockers, or Slurm--see the following sections for more details. To clone it, use the following Bash command:

```
git clone --recurse-submodules https://github.com/qBioTurin/FORGE4FLAME.git
```

The user can run F4F:

- Installing and running the R package.
- Downloading and running the provided Docker image.

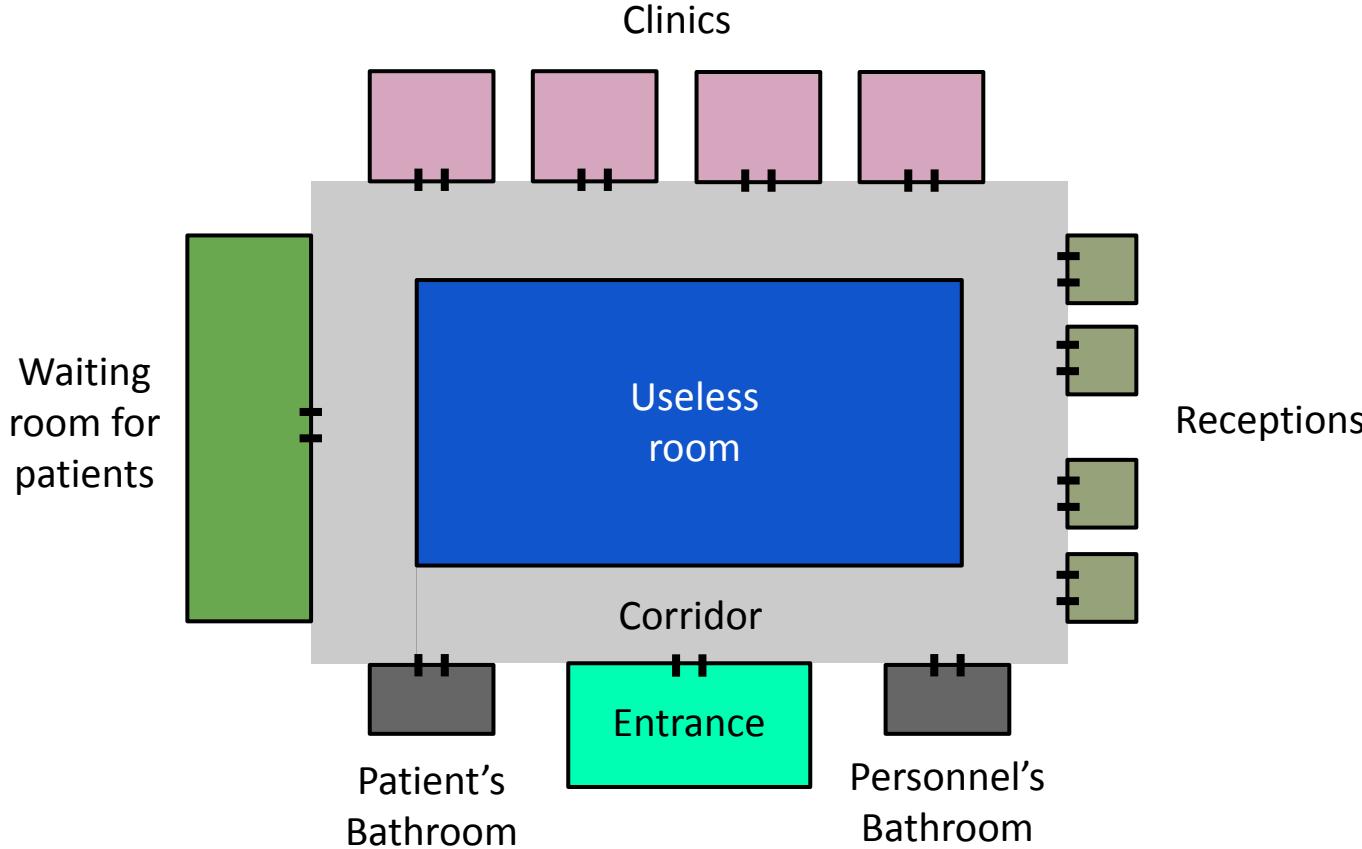
See the [GitHub page](#) for more details.



<http://github.com/qBioTurin/IEEEICHI2025>

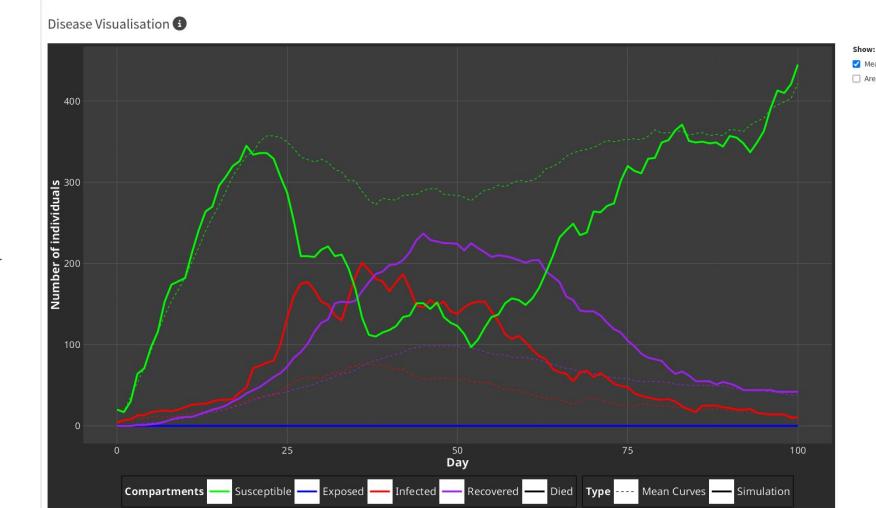
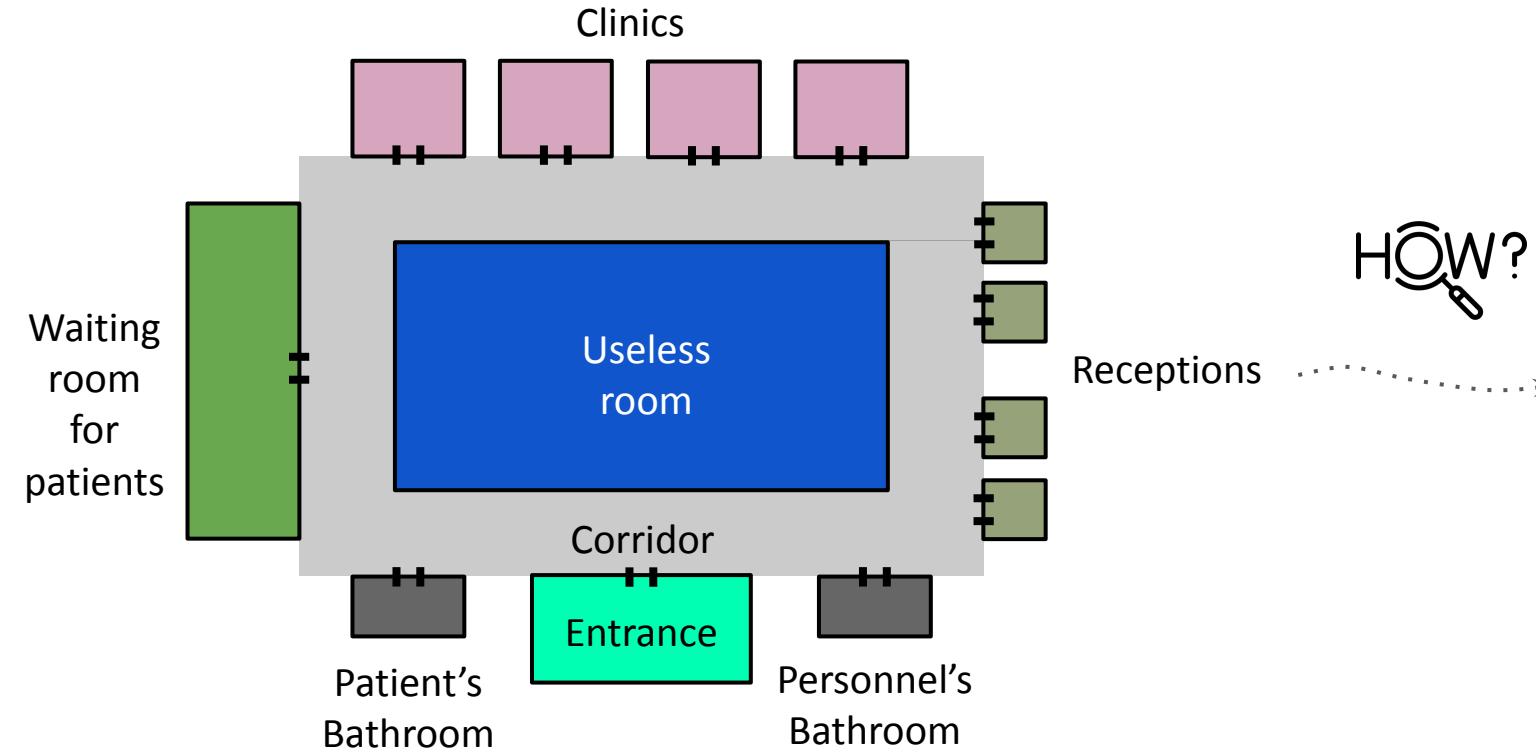
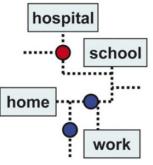


Case study: small ambulatory



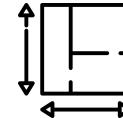
- 12 doctors and 12 receptionists are divided across three shifts (6 AM–2 PM, 2 PM–10 PM, and 10 PM–6 AM). Doctors remain in the clinic during their shifts, while receptionists stay at the reception.
- On average, 20 patients visit the clinic per day, each required to attend four weekly appointments. They check in at the reception (≈ 5 min) and proceed to the clinic (≈ 40 min), waiting their turn in the respective waiting rooms.
- Each individual may use the bathroom with a predefined probability: $1.4e^{-4}$ for both receptionists and doctors, and $2.8e^{-4}$ for patients.

Case study: small ambulatory





Canvas page with empty canvas



Build your ABM

- [Home](#)
- [Canvas](#)
- [Rooms](#)
- [Agents](#)
- [Resources](#)
- [Infection](#)
- [What-if](#)
- [Configuration](#)
- [Run](#)
- [Settings](#)
- [Post Processing](#)

Define floor

Define/Select floor

Define or select floor

Delete floor

Add elements

Draw rooms

Select the room:

Length of selected room (length refers to the wall with the door):

Width of selected room:

Height of selected room:

Door position:

right

Area:

None

Add room

Remove rooms

Select the room to remove:

Remove room

Clear floor

Agents management

Add graph point

Remove last graph point

Visualise graph

Colour fill by:

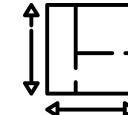
Room

The canvas represents a plane (or floor) where predefined rooms can be added and positioned using drag and drop. At least one floor has to be defined.

Before building the environment, rooms must be defined.



Rooms page



Build your ABM

Define a new room

Name: Define a room

Type: Normal

Length (meter): Room length

Width (meter): Room width

Height (meter): Room height

Save room

Set colours legend

Select room: room0

Select type: Normal

Select area: None

#5FFB07

Select colour for Normal

#FF0000

Select colour for None

#000000

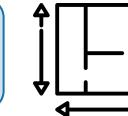
A room is defined by:

- 3 dimensions (*Length, Width, Height*),
- name (*room's id*), and
- type (*to group rooms into clusters*)

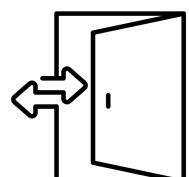
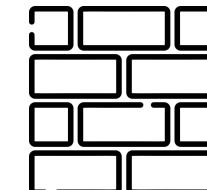
It is possible to define and change the color of rooms, types and areas displayed in the *Canvas*.



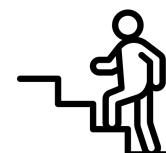
Rooms page: predefined rooms' type



Filling-rooms: room type utilized by agents to simulate walls or useless rooms.



Spawn-rooms: room type utilized by agents to enter (and exit) the environment.



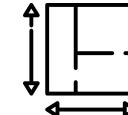
Stairs: room type utilized by agents to move among floors.



Waiting-rooms: room type utilized by agents to wait to enter in another room.



Canvas page



Build your ABM



Home

Canvas

Rooms

Agents

Resources

Infection

What-if

Configuration

Run

Settings

Post Processing

Define floor

Define/Select floor

Define or select floor

Delete floor

Add elements

Draw rooms

Select the room:

Length of selected room (length refers to the wall with the door):
Width of selected room:
Height of selected room:

Door position:

right

Area:

None

Add room

Remove rooms

Select the room to remove:

Remove room

Clear floor

Agents management

Add graph point

Remove last graph point

Visualise graph

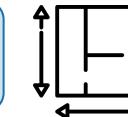
Colour fill by:

Room

Each room may be assigned to a particular area (e.g., a specific area of a hospital). This is useful restrict the areas in which agents can go.



Canvas page: floor0



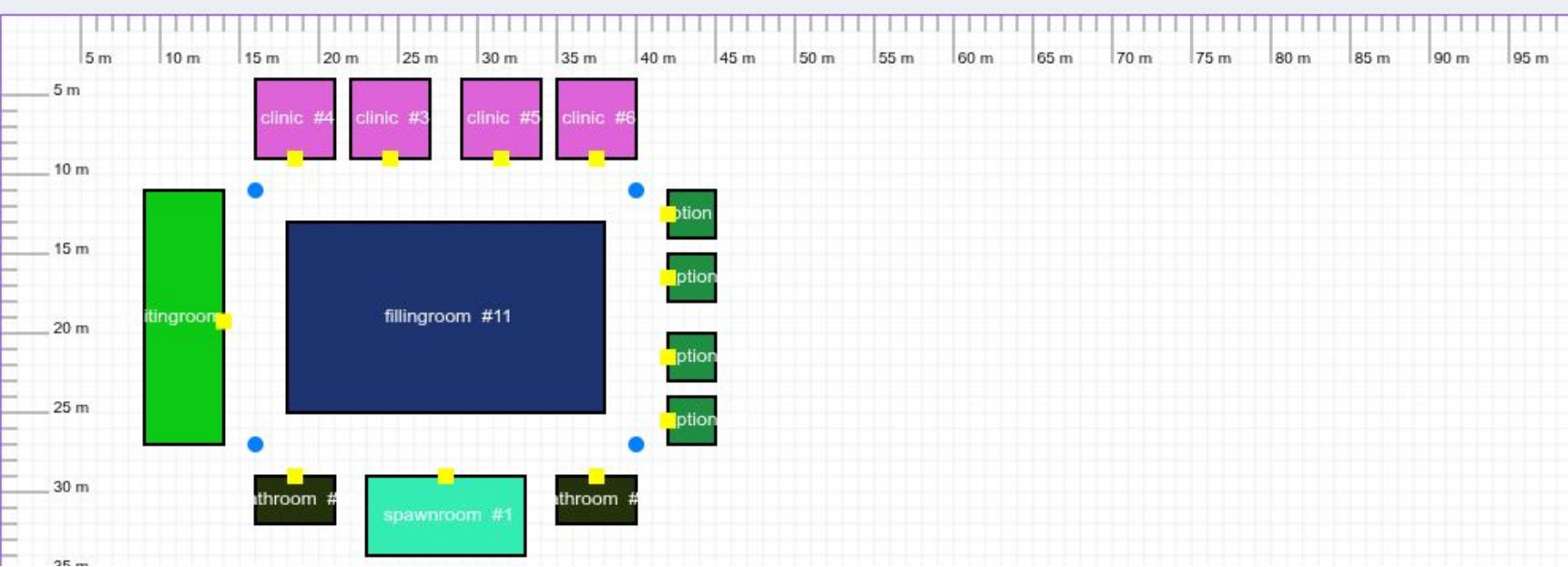
Define floor

Define>Select floor

floor0

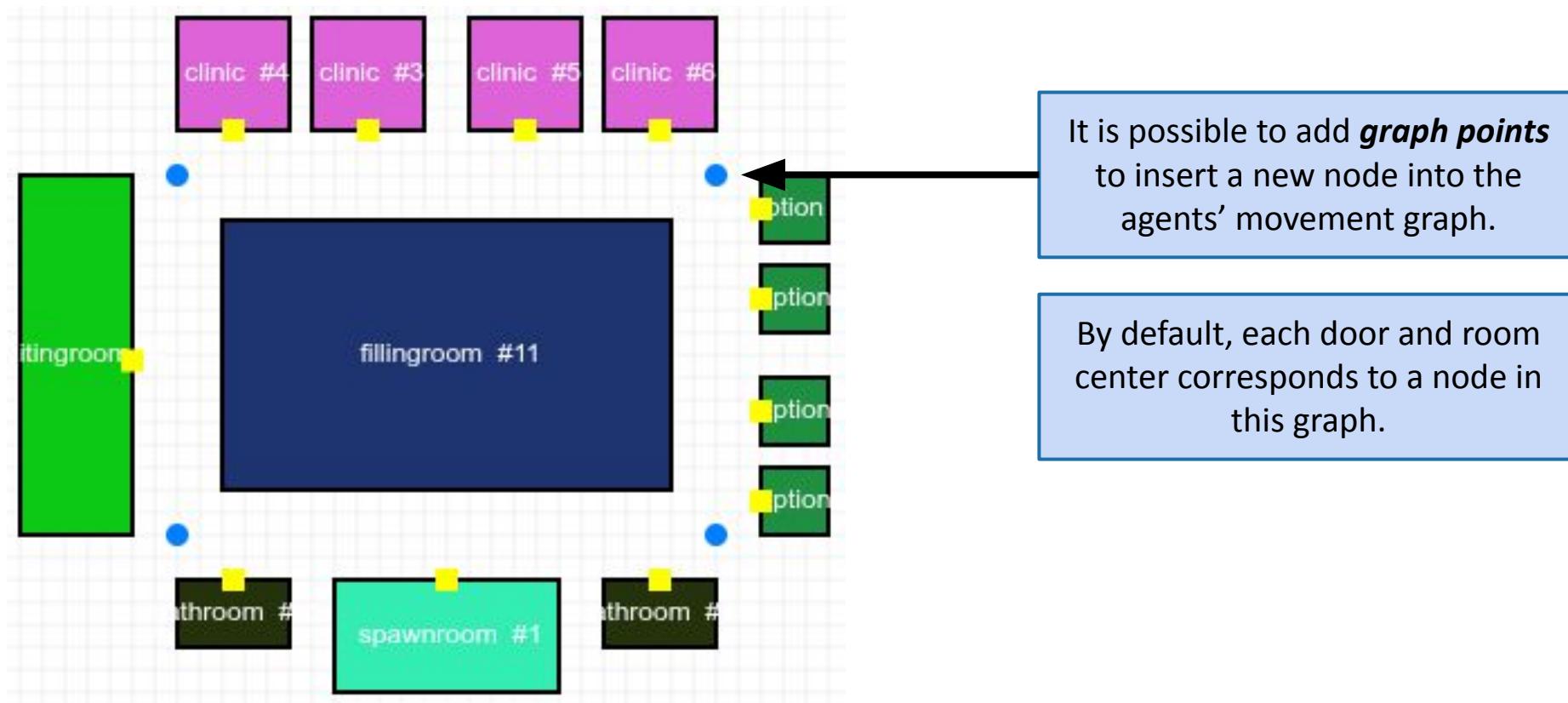
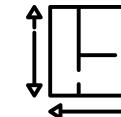
Delete floor

Add elements



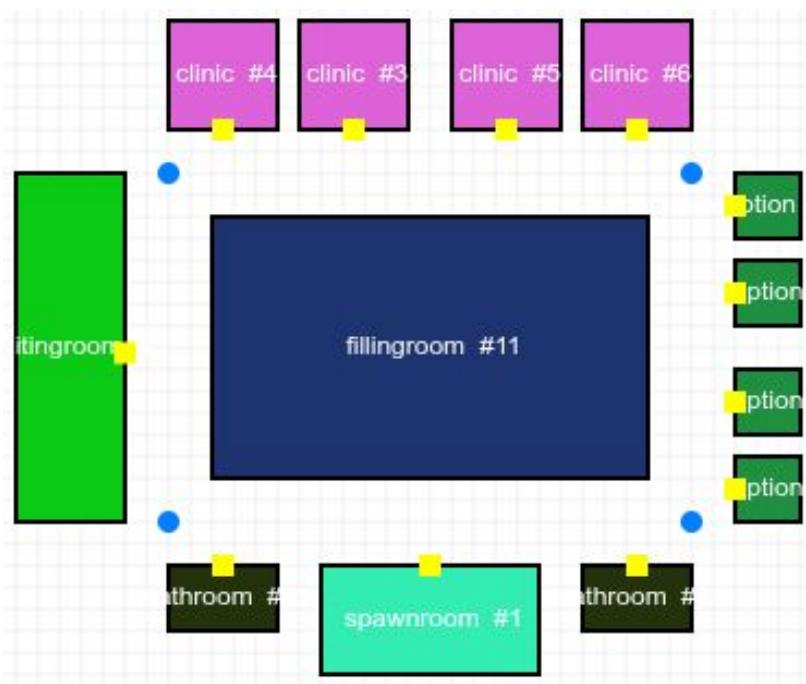
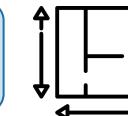


Canvas page: graph points





Canvas page: graph points



It is possible to generate and visualise all the possible paths that the agents might follow.





Agents

Build your ABM

- Home
- Canvas
- Rooms
- Agents**
- Resources
- Infection
- What-if
- Configuration
- Run
- Settings
- Post Processing

Agent definition

Agent name: Remove agent

Copy information from: Copy

Agent class: Doctor

Number of agents: The number must be a positive integer

Determined flow

Type: Activity:

Deterministic Stochastic

Fixed deterministic value: Value

Add room Remove last room

Add flow Remove flow

Random flow

Type: Activity: Weight:

Deterministic Stochastic

Fixed deterministic value: Value

Entry flow

Select type of entrance:
 Daily Rate
 Time window

Save time

1 slot

Entry time: hh:mm

Select Days of the Week

Monday
 Tuesday
 Wednesday
 Thursday
 Friday
 Saturday
 Sunday

Add slot Remove slot

Agent definition

Determined flow

Random flow

Entry/Exit flow



Agent page: agent definition



Agent definition i

Agent name:

doctor_firstshift

Remove agent**Copy information from:**▼Copy**Agent class:**

Doctor

Number of agents:

4

An agent is defined by:

- a name,
- a class (category that groups different types of agents), and
- the number of agents in the simulation.



Agent page: agent determined flow



Determined flow i

Type:

Activity:

Deterministic
Stochastic

Fixed deterministic value:

Add room
Remove last room

1 flow

Drag the rooms in the desired order

Spawnroom-None - Exponential 5 min - Very Light

Clinic-None - Deterministic 480 min - Light

(2) Spawnroom-None - Exponential 5 min - Very Light

The flow must always begin
and end with the
spawn-room.

The user can select the intensity of the activity that the agent performs in the selected room. This is important to determine the quantity of quanta exhaled by an infected agent.

Options: very light, light, quite hard, hard.



Agent page: agent random flow



The user can add random events that interrupt the agent's determined flow.

Random flow i

Type:	Activity:	Weight:	<input type="radio"/> Deterministic <input type="radio"/> Stochastic <small>Fixed deterministic value:</small> <input style="width: 100px; height: 20px; margin-bottom: 5px;" type="text"/> Value	Add room
<small>Click on an event to remove it (except the 'Do nothing' event)</small> Show 5 entries				Search:
Room	Distribution	Activity	Time	Weight
Do nothing	Deterministic	Light	0	0.99986
Bathroom-Personnel	Exponential	Light	3	0.00014

Showing 1 to 2 of 2 entries

Previous 1 Next

By default the event *Do nothing* is present with probability 1.

The probability that an event might occur will be checked every step.



Agent page: agent time window



Entry flow

Select type of entrance: Daily Rate Time windowSave time

1 slot

Entry time:

06:00

Select Days of the Week

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

Associate with a determined flow:

1 flow

Add slotRemove slot

The user can define the entry time and the specific days of the week when the agent will enter the environment.

Copy information from:**Number of agents:**

The number of agents is specified in the agent definition panel.



Agent page: agent daily rate



Entry flow

Select type of entrance:

 Daily Rate Time window

Save time

The user can define an entrance rate for the agent using either a deterministic value or a stochastic distribution, along with specifying an initial and a final generation time and the specific days of the week when the agent will enter the environment.

1 slot

Entrance rate:

Deterministic

Stochastic

Distribution:

Truncated Positive Normal

Mean:

20

Sd:

10

Initial generation time:

00:00

Final generation time:

23:59

Select Days of the Week

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

Add slot

Remove slot



Resources page



Build your ABM

Set resources

Select global resources for each room and agent type:

0

Select second choice for each agent:

Same room

Set

Select type and area:

Resources: the number of agents that can occupy a room simultaneously.

For each room, the list of all agents that might reach it is showed. The user can set the rooms' resources available for each agent.

The sidebar menu includes: Home, Canvas, Rooms, Agents, Resources (selected), Infection, What-If, Configuration, Run, Settings, and Post Processing.



Resources page



Set resources

Select global resources for each room and agent type:

0

Select second choice for each agent:

Same room

Set

Select type and area:

Bathroom-Personnel

Room	Maximum	receptionist_firstshift	reception_secondshift	reception_thirdshift	doctor_firstshift	doctor_secondshift	doctor_thirdshift
bathroom	4	4	4	4	4	4	4

Select second choice room in Random Flow for receptionist_firstshift:

Skip room

Select second choice room in Random Flow for reception_secondshift:

Skip room

Select second choice room in Random Flow for reception_thirdshift:

Skip room

Select second choice room in Random Flow for doctor_firstshift:

Skip room

Select second choice room in Random Flow for doctor_secondshift:

Skip room

Select second choice room in Random Flow for doctor_thirdshift:

Skip room



Infection page



Build your ABM

≡

Home

Canvas

Rooms

Agents

Resources

Infection

What-If

Configuration

Run

Settings

Post Processing

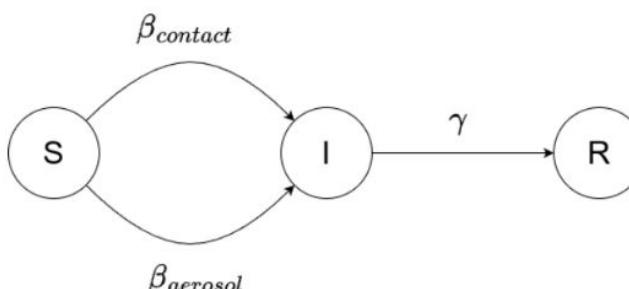
Disease model

Model:

SIR

Disease model: SIR. Beta (contact): 0.024, Beta (aerosol): 410, Gamma: 0.1 (Deterministic)

SIR: Susceptible - Infected - Recovered.



The evolution of the infection is modeled through a compartmental model.

Description:

The SIR model is a straightforward compartmental model used to study the dynamics of infectious diseases. It divides the population into three key groups. The first group, susceptible (S), includes individuals who are vulnerable to the disease. When these individuals come into contact with infected people or share a confined space with them, they can contract the disease and transition to the infected group. The second group, infected (I), consists of individuals who are carrying the disease and can transmit it to those who are susceptible. The third group, recovered (R), includes individuals who have overcome the infection and gained immunity. Overall, the SIR model offers a simplified framework to analyze the spread of infectious diseases within a population, focusing on the interactions between the susceptible, infected, and recovered groups.

① $\beta_{\text{contact}} =$
0.024

② $\beta_{\text{aerosol}} =$
410

③ $\gamma =$

Deterministic Stochastic

Fixed deterministic value:

0.1

Save

Two contagion processes: contact [1] and aerosol [2].

[1] Hoertel, N., et al. A stochastic agent-based model of the SARS-CoV-2 epidemic in France.

[2] Gkantzas, S., et al. *airborne.cam*: A risk calculator of SARS-CoV-2 aerosol transmission under well-mixed ventilation



Infection page: SEIRDS

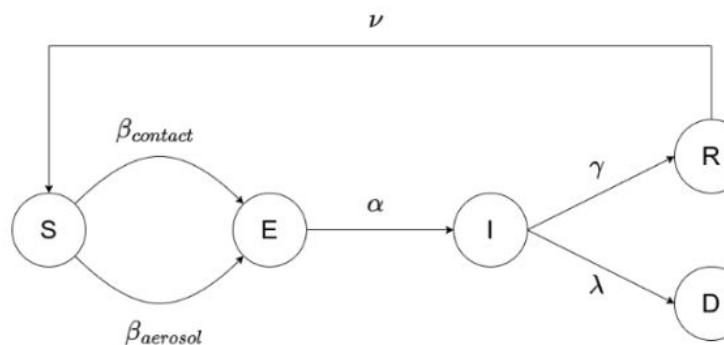


Disease model

Model:

SEIRDS

β_{contact} and β_{aerosol} are not the infection rates!



SEIRDS: Susceptible - Exposed - Infected - Recovered - Dead - Susceptible.

Description:

The SEIRDS model is an extension of the SEIRD model that introduces the possibility of waning immunity, adding a dynamic aspect to the compartmental framework. It includes five compartments. The first compartment, susceptible (S), represents individuals who are vulnerable to the disease. These individuals can contract the disease and transition to the exposed compartment upon contact with infected individuals or sharing a closed space with them. The second compartment, exposed (E), consists of individuals who have been infected but are in an incubation period, during which they cannot yet transmit the disease. The third compartment, infected (I), includes individuals who are actively infected and capable of spreading the disease to susceptible individuals. The fourth compartment, recovered (R), represents individuals who have overcome the infection and gained immunity. The fifth compartment, dead (D), accounts for individuals who have died as a result of the disease. Unlike the SEIRD model, the SEIRDS model assumes that immunity is temporary. After a certain period, individuals in the recovered compartment lose their immunity and return to the susceptible compartment, re-entering the cycle. This feature makes the SEIRDS model particularly relevant for analyzing diseases where immunity wanes over time.



What-If page



Build your ABM

Countermeasures

Saved Countermeasures

Measure	Type	Parameters	From	To
All	All	All	All	All
No data available in table				
Measure	Type	Parameters	From	To
All	All	All	All	All
No data available in table				

Ventilation

Masks

Vaccination

Swabs

Quarantine

External screening

Virus parameters

Saved Virus Parameters

Virus

Virus variant factor: 1

Virus severity: 0.22

Save

Initial infected

Initial infected: 1

Initial infected:

What-If scenarios are useful to understand the potential impact of hypothetical situations.

Options:

- Environmental ventilation
- Masks usage
- Virus parameters
- Vaccination
- Swabs (internal screening)
- Quarantine
- External screening
- Initial infected
- Outside contagion



What-If page: ventilation



Global

Ventilation

Ventilation:

- Global
- Different for each room

Ventilation (in ACH):

0.3 (poorly ventilated)

From (day):
1

To (day):
60

Ventilation is a process that replace contaminated with clean air at n Air Changes per Hour (ACH).

Different for each room

Ventilation

Ventilation:

- Global
- Different for each room

Room:

Waitingroom-None

Ventilation (in ACH):

5 (well ventilated)

From (day):
1

To (day):
60

Options: no ventilation, poorly ventilated, domestic, offices/school, well ventilated, typical maximum, hospital setting.



What-If page: masks



Global

Masks reduce the virus inhalation and exhalation rate.

Masks

Mask:

- Global
- Different for each agent

Mask type:

No mask

% mask: i

From (day):

To (day):

Different for each agent

Options: no mask, surgical, FFP2.

Masks

Mask:

- Global
- Different for each agent

Agent:

doctor_firstshift

Mask type:

FFP2 mask

% mask: i

From (day):

To (day):



What-If page: vaccination



Global

Vaccination

Vaccination:

- Global
- Different for each agent

Fraction of vaccinated agents:

1

Vaccine efficacy:

1

Vaccination immunizes people by putting them in the Recovered state.

Vaccine coverage (day):

Deterministic Stochastic

Fixed deterministic value:

180

At (day):

1

Different for each agent

Vaccination

Vaccination:

- Global
- Different for each agent

Agent:

doctor_firstshift

Fraction of vaccinated agents:

0.5

Vaccine efficacy:

0.8

Vaccine efficacy models the fraction of the vaccinated population that is effectively immune to infection.

Vaccine coverage (day):

Deterministic Stochastic

Fixed deterministic value:

180

At (day):

1



What-If page: swab



Global

Swabs

Swab:
 Global
 Different for each agent

Sensitivity:
Specificity:

Agent:

▾

Swab:

No swab
 Swab

A swab every how many days?

Deterministic Stochastic

Fixed deterministic value:

From (day):

To (day):

Internal screening campaigns are useful to detect infected people and to monitor the spread of the infection in the environment.

Different for each agent

Swabs

Swab:
 Global
 Different for each agent

Sensitivity:
Specificity:

A swab every how many days?

Deterministic Stochastic

Fixed deterministic value:

From (day): To (day):

These screening campaigns are conducted **within** the environment!



What-If page: quarantine



Global

Quarantine

Quarantine:

- Global
- Different for each agent

Quarantine days:

Deterministic Stochastic

Quarantine room for severe cases:

Spawnroom-None

Fixed deterministic value:

10

Swab:

- No swab
- Swab

Sensitivity:

1

Specificity:

1

A swab every how many days?

Deterministic Stochastic

Fixed deterministic value:

7

From (day):

1

To (day):

60

Isolating infected people is useful to slow down the spread of the virus.

Different for each agent

Quarantine

Quarantine:

- Global
- Different for each agent

Agent:

doctor_firstshift

Quarantine:

- No quarantine
- Quarantine

Quarantine days:

Deterministic Stochastic

Fixed deterministic value:

7

Swab:

- No swab
- Swab

Sensitivity:

1

Specificity:

1

A swab every how many days?

Deterministic Stochastic

Fixed deterministic value:

3

From (day):

1

To (day):

60



What-If page: external screening



Global

External screening

External screening:

- Global
- Different for each agent

Agent:

doctor_firstshift

Screening campaigns: i

0.01

Symptoms: i

0.05

External screening campaigns are useful to account for interactions **outside** the environment.

Different for each agent

External screening

External screening:

- Global
- Different for each agent

Screening campaigns: i

0.0003

Symptoms: i

0.0075

Two probabilities:

- test an individual because he/she follows activities that involve screening campaigns (like practise sports)
- test an infected individual due to symptoms.



What-If page: virus



Virus variant factor models the virus or a particular variant of the virus under study [1].

Virus

Virus variant factor:  1

Virus severity:  0.22

Virus severity models the fraction of infected with severe symptoms [2].

[1] J.L. Jimene, et al. *COVID-19 Aerosol Transmission Estimator*.

[2] Tolksdorf K, et al. *Influenza-associated pneumonia as reference to assess seriousness of coronavirus disease (COVID-19)*.



What-If page: initial infected



Random

Initial infected

Initial infected: ⓘ

- Random
- Global
- Different for each agent

Initial infected:

1

Global

Initial infected

Initial infected: ⓘ

- Random
- Global
- Different for each agent

Initial infected:

1

Different for each agent

Initial infected

Initial infected: ⓘ

- Random
- Global
- Different for each agent

Agent:

doctor_firstshift

Initial infected:

1

The number of initial infected within the environment can pilot the evolution of the infection.



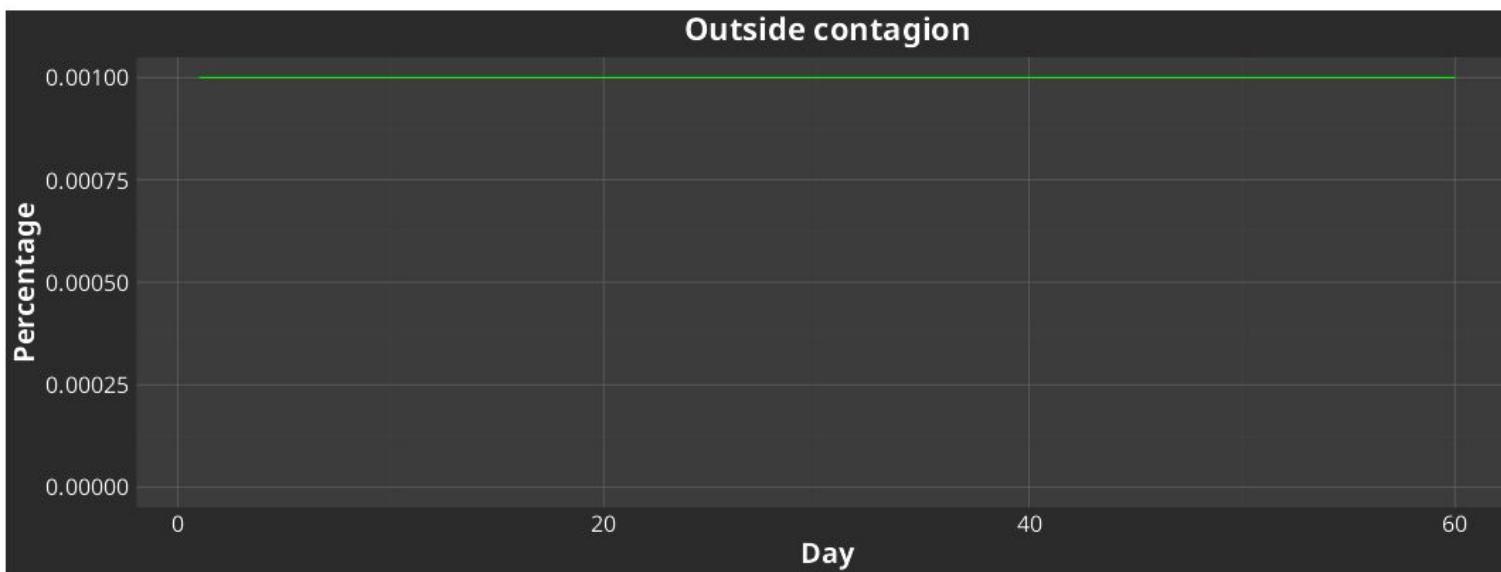
What-If page: outside contagion

Outside contagion i

Browse... outside_contagion.csv
Upload complete

i Load

Population: i
1



Outside contagion accounts for the fact that any individual will likely catch the infection outside the environment.



Configuration page



Build your ABM

[Home](#)[Canvas](#)[Rooms](#)[Agents](#)[Resources](#)[Infection](#)[What-If](#)[Configuration](#)[Run](#)[Settings](#)[Post Processing](#)

Configuration parameters

Simulation days:

100

Initial day:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

Initial time:

00:00

Step:

60

Definition of some basic configurations





Run page



Build your ABM

Run models

Seed: 1749561324

Number of simulations: 10

Number of parallel simulations: 10

Select run type:

- Docker
- Local with 3D visualisation
- Local
- Docker

Simulation Log

Run a defined model.

Stop



Setting page



Build your ABM

- [Home](#)
- [Canvas](#)
- [Rooms](#)
- [Agents](#)
- [Resources](#)
- [Infection](#)
- [What-if](#)
- [Configuration](#)
- [Run](#)
- [Settings](#)
- [Post Processing](#)

Set floor dimension

Height (meter)
Floor height dimension (default 80m)

Width (meter)
Floor width dimension (default 100m)

[Set dimension](#)

Load a saved model

[Browse...](#) Select an RDs file.

[Load](#)

Save the model

[Check model](#)

[Save the model](#)

[Link the model to FLAME GPU 2](#)

References

[1] HOERTEL, N., Blachier, M., Blanco, C., Olfson, M., Massetti, M., Rico, M. S., Limosin, F., & Leleu, H. (2020). A stochastic agent-based model of the SARS-CoV-2 epidemic in France. *Nature Medicine*, 26(9), 1417–1421. doi:<https://doi.org/10.1038/s41591-020-1001-6>

[2] GKANTONAS, S., Zabotti, D., Mesquita, L. C., Mastorakos, E., & de Oliveira, P. M. (2021). airbone.cam: A risk calculator of SARS-CoV-2 aerosol transmission under well-mixed ventilation conditions. Available at: <https://airbone.cam>

[3] J.L. Jimenez and Z. Peng. COVID-19 Aerosol Transmission Estimator. <https://tinyurl.com/covid-estimator>

[4] Tolsdorf K, Buda S, Schuler E, Wieler LH, Haas W. Influenza-associated pneumonia as reference to assess seriousness of coronavirus disease (COVID-19). *Euro Surveill*. 2020 Mar;25(11):2000258. doi: <https://doi.org/10.2807/1560-7917.EU.2020.25.11.2000258>. Epub 2020 Mar 16. PMID: 32186278; PMCID: PMC7096775

FLAME GPU 2 references:

[5] Richmond, P., Chisholm, R., Heywood, P., Leach, M., Chimeh, M. K. FLAME GPU. Version 2.0.0-rc. Dec. 2022. <https://doi.org/10.5281/zenodo.7434228>

[6] Richmond, P., Chisholm, R., Heywood, P., Chimeh, M. K., Leach, M. FLAME GPU 2: A framework for flexible and performant agent based simulation on GPUs. In: Software: Practice and Experience (2023). <https://doi.org/10.1002/spe.3207>

ABM school references:

[7] Baccega, Daniele, Pernice, Simone, Terna, Pietro, Castagno, Paolo, Moirano, Giovenale, Richiardi, Lorenzo, Sereno, Matteo, Rabellino, Sergio, Maule, Milena Maria and Beccuti, Marco (2022) 'An Agent-Based Model to Support Infection Control Strategies at School' *Journal of Artificial Societies and Social Simulation* 25 (3) 2 <http://jasss.soc.surrey.ac.uk/25/3/2.html>. doi: <https://doi.org/10.18564/jasss.4830>

[8] Daniele Baccega, Simone Pernice, Paolo Castagno, Matteo Sereno, and Marco Beccuti. Evaluating the Impact of Mask and Quarantine Policies on the Spread of COVID-19 in Schools using computational modeling. In the 18th Conference on Computational Intelligence Methods for Bioinformatics & Biostatistics (CIBB 2023)

Definition of some settings, load/save models, references

Epidemic Modelling Frameworks for Studying Infectious Disease Dynamics 108



Post Processing page



Build your ABM

Uploading simulation i

Select Folder Load

Query on Disease Status i

Resulting Simulations i

Disease Visualisation i +

2D Simulation Visualisation i +

Contact Matrix +

Home

Canvas

Rooms

Agents

Resources

Infection

What-if

Configuration

Run

Settings

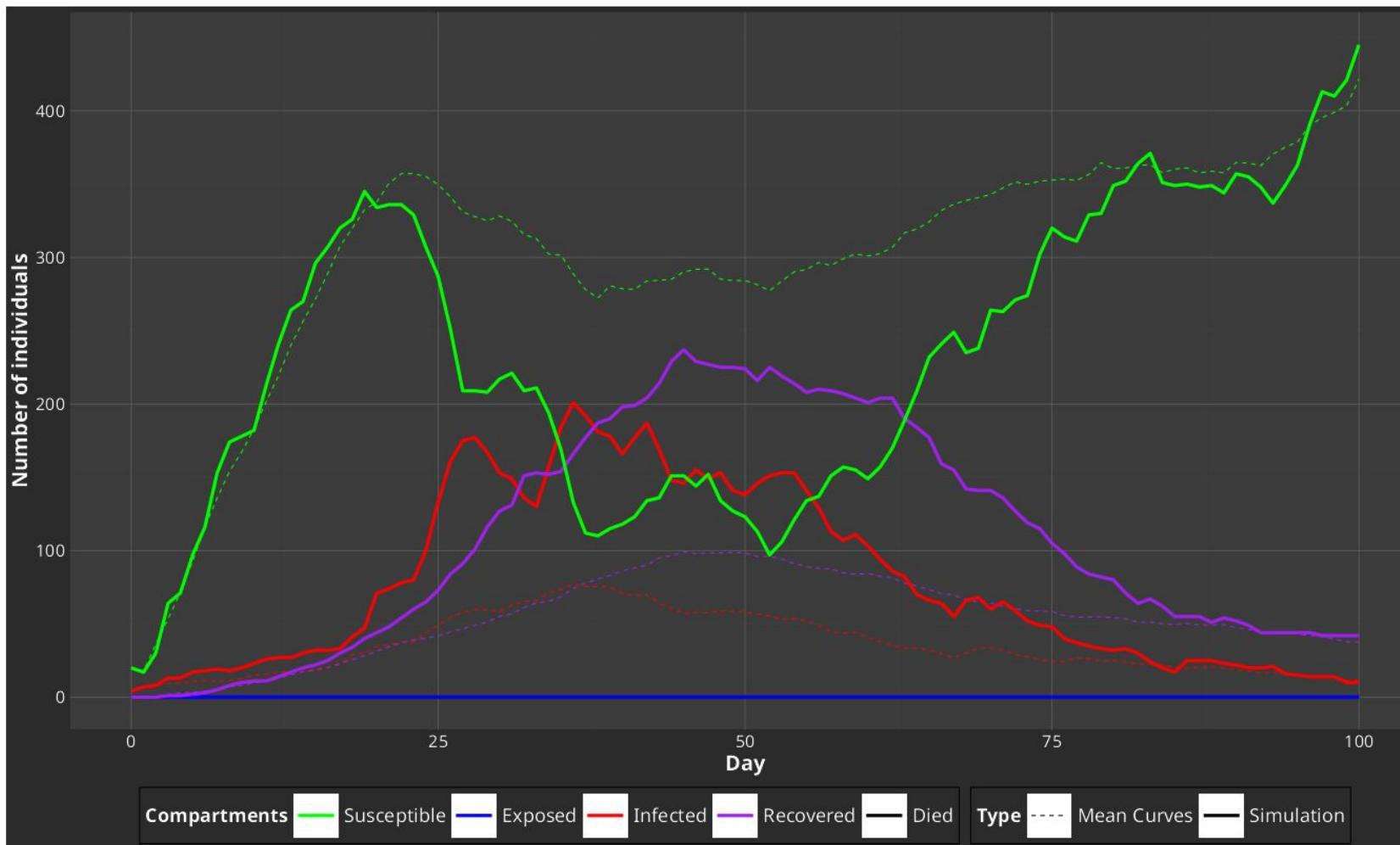
Post Processing



Post Processing page: disease evolution



Disease Visualisation



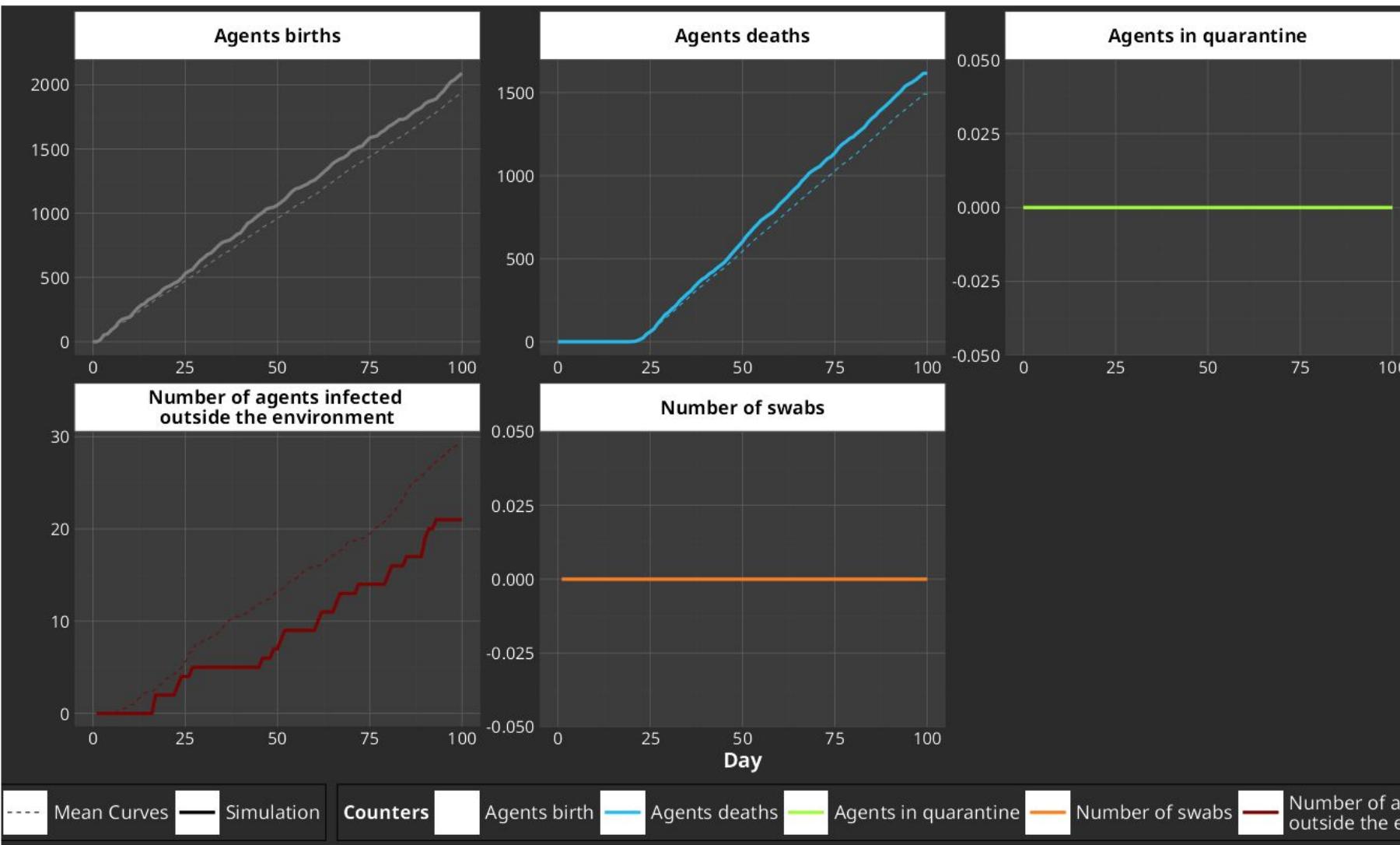
Disease progression in the selected simulation (with average).



Post Processing page: other plots



Show:
 Mean curves
 Area from all simulations



Some other useful information in the selected simulation (with average).



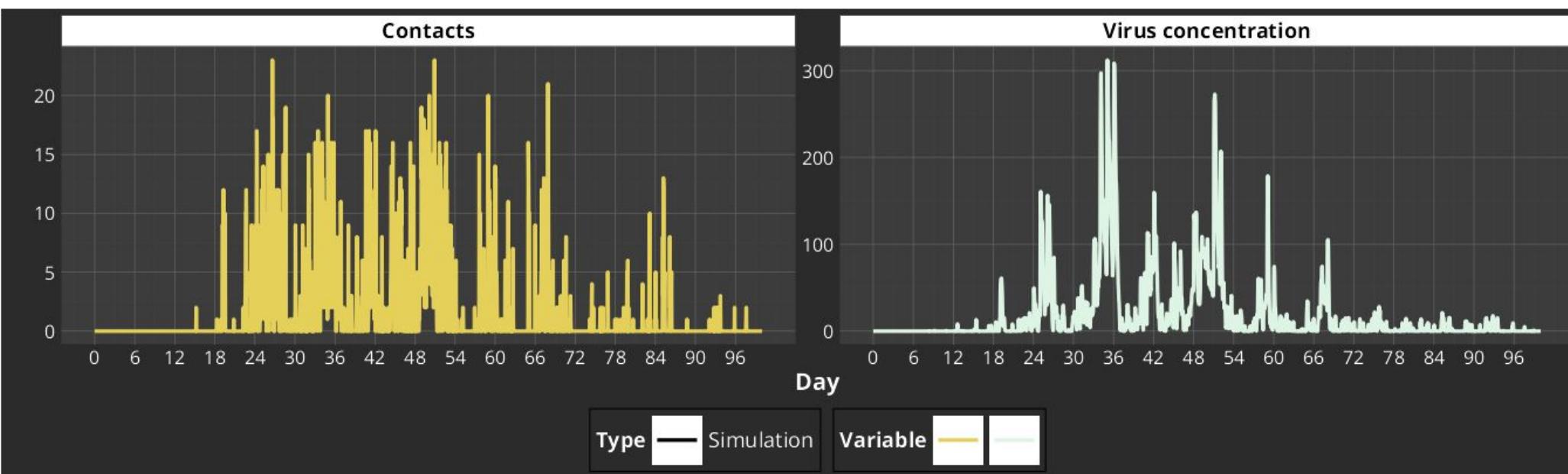
Post Processing page: contacts and virus concentration



Choice of the room: i

floor0 ; None ; waitingroom ; ID 3

Number of contacts and virus concentration inside the waiting room.





Post Processing page: 2D visualization



2D Simulation Visualisation ⓘ

Features ⓘ

Select floor to visualise: All Select agent type to visualise: All Select colour room: Name Show in the plot:

- None
- ID
- Name
- Type
- Area
- Agent ID

2D visualisation

Time in the animation (sec): 21,600 4,569,720 5,552,940 Set the animation step (sec): 60

Next

53d:21h:22m:0s (# steps: 76162)
floor0

Agent type: △ doctor_secondshift + reception_secondshift ▽ patient

Disease state: ● S ● E ● I ● R ● D

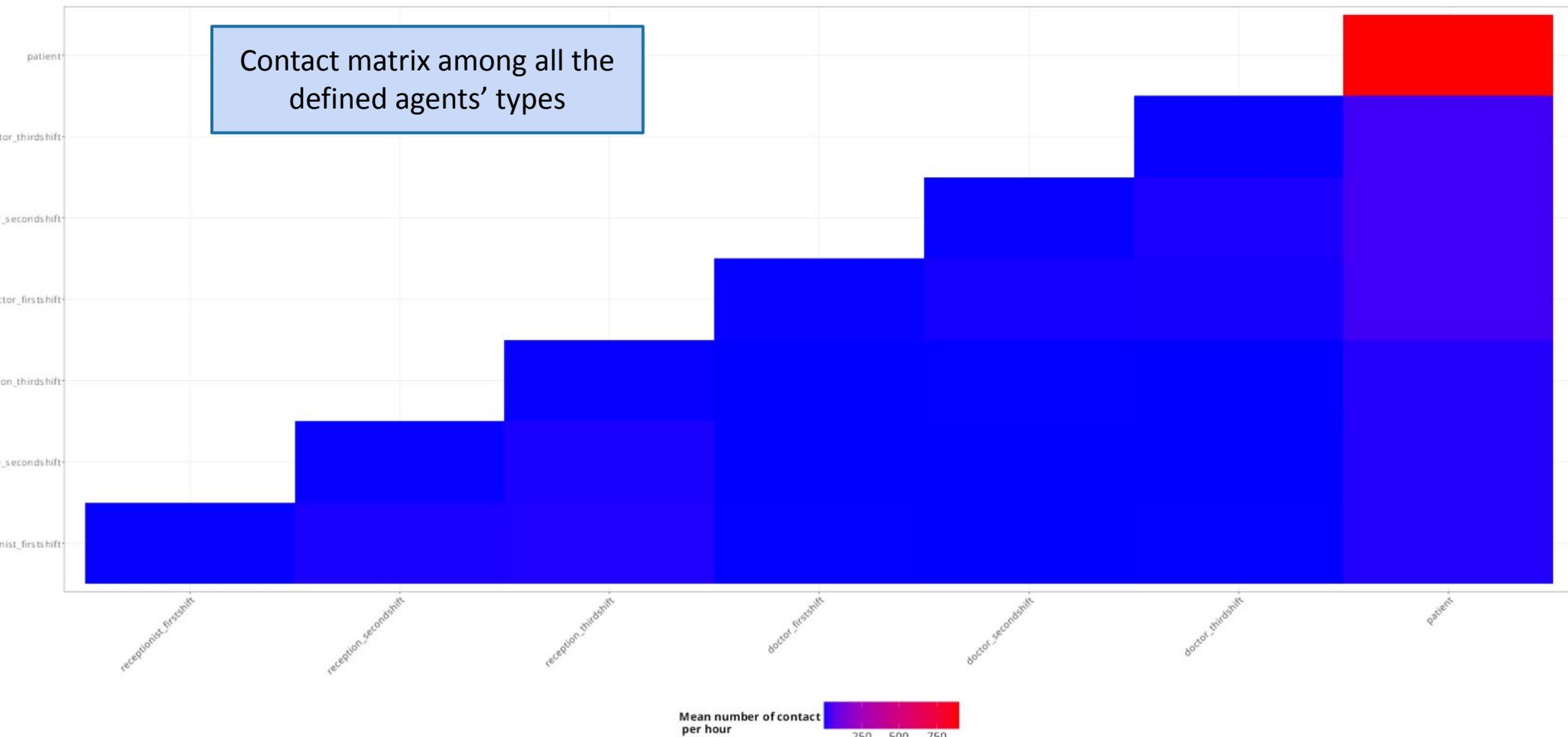
2D visualization in
Forge4Flame



Post Processing page: contact matrix

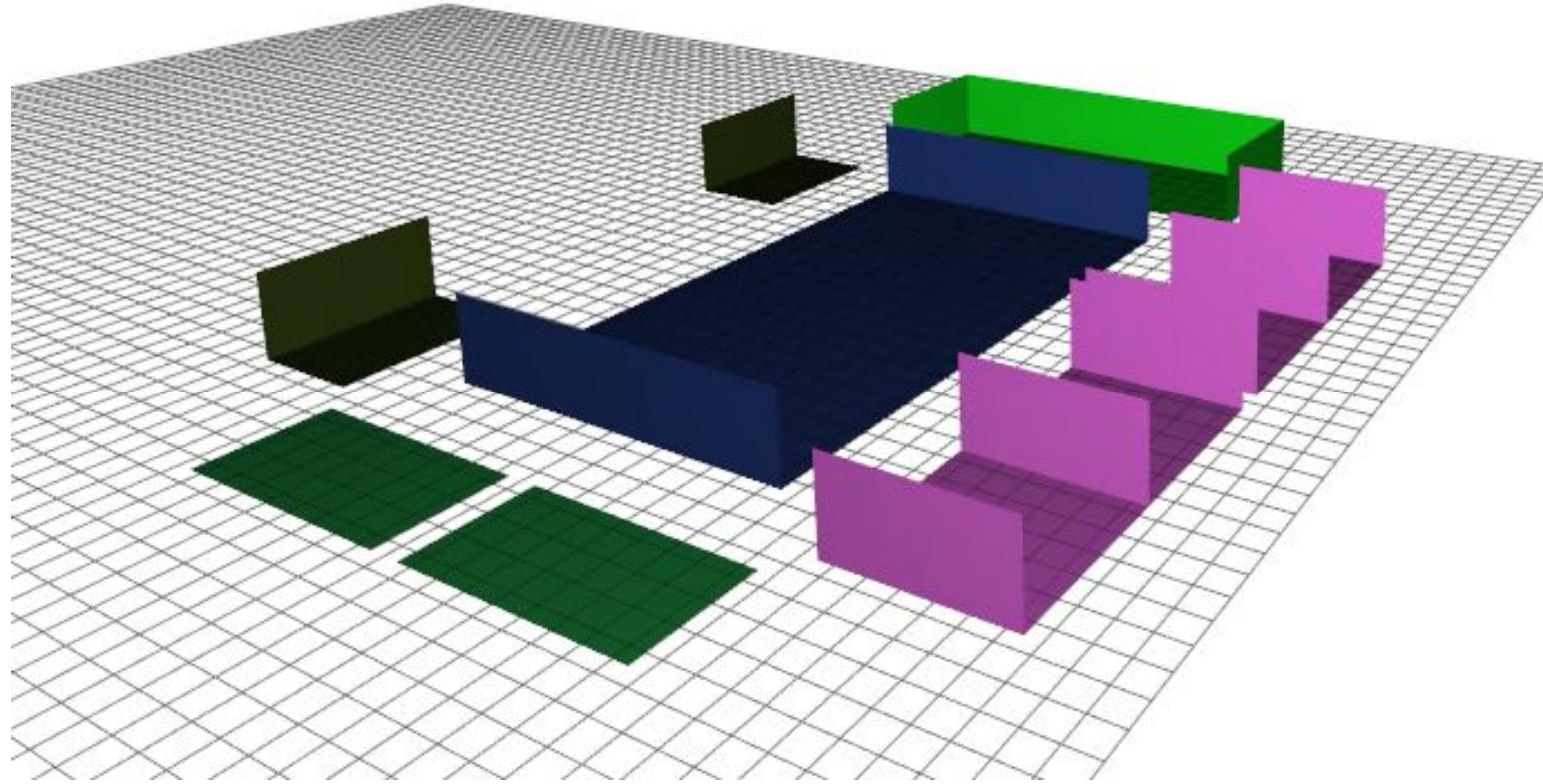


Contact Matrix





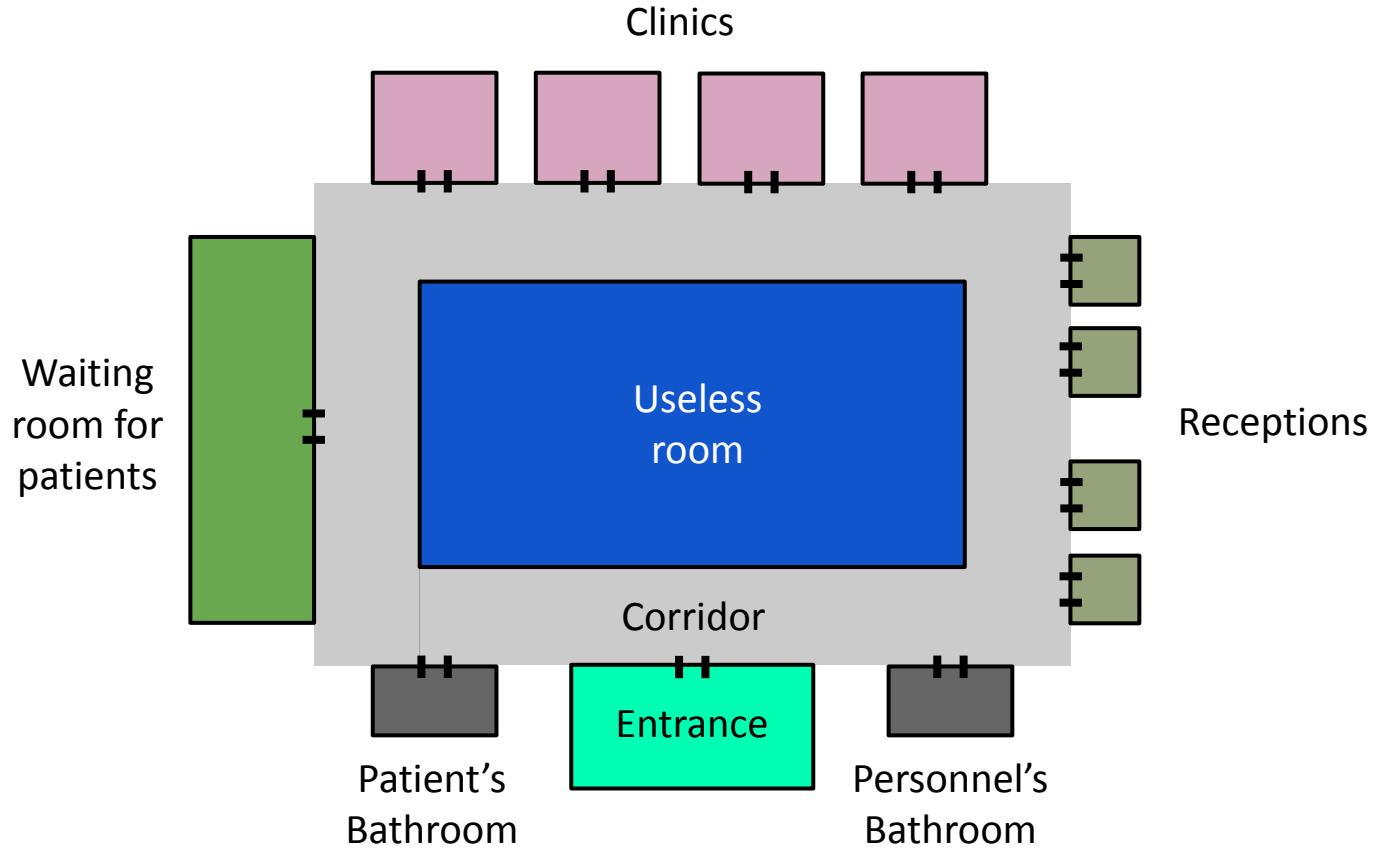
3D visualization in FLAME GPU 2



3D visualisation
provided by FLAME
GPU 2

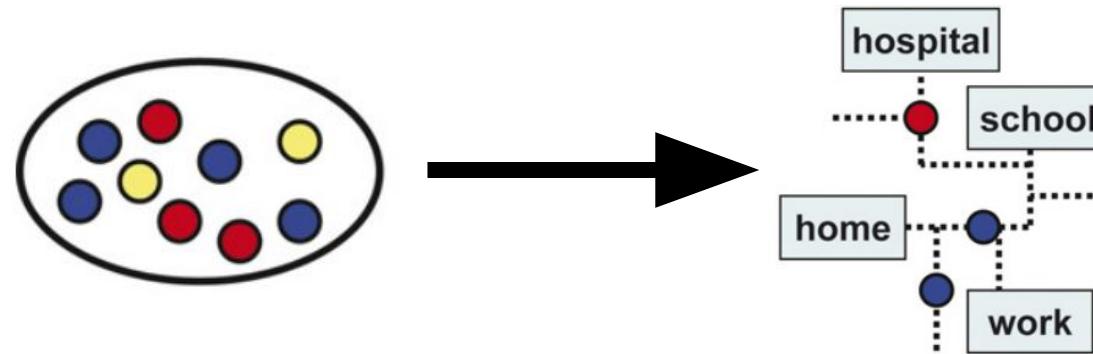


Case study: small ambulatory



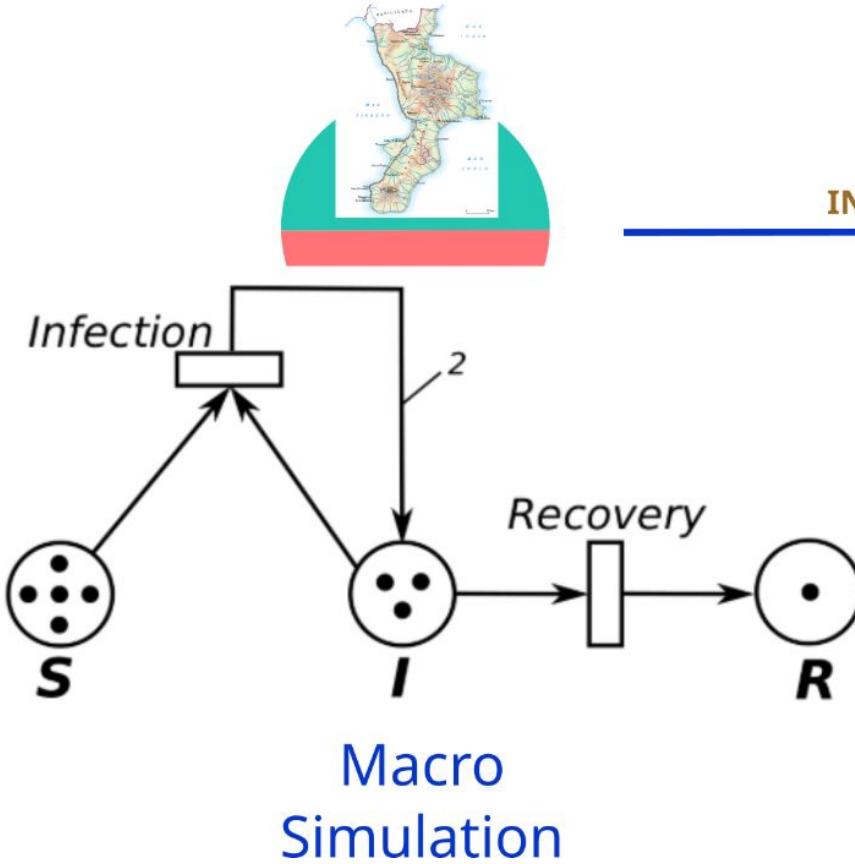
<http://github.com/qBioTurin/IEEEICHI2025>

From macro- to micro-level models

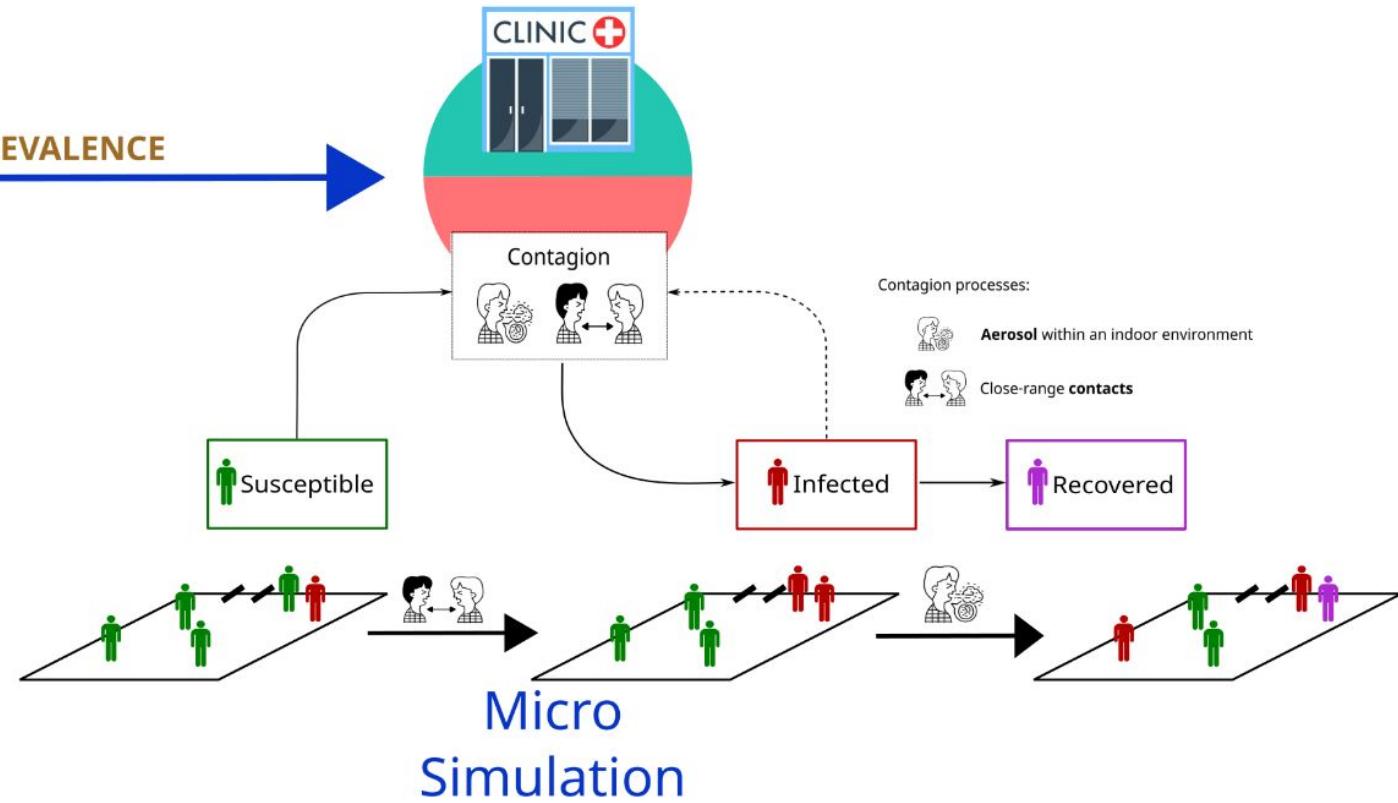


From macro to micro-level models

Large Scale



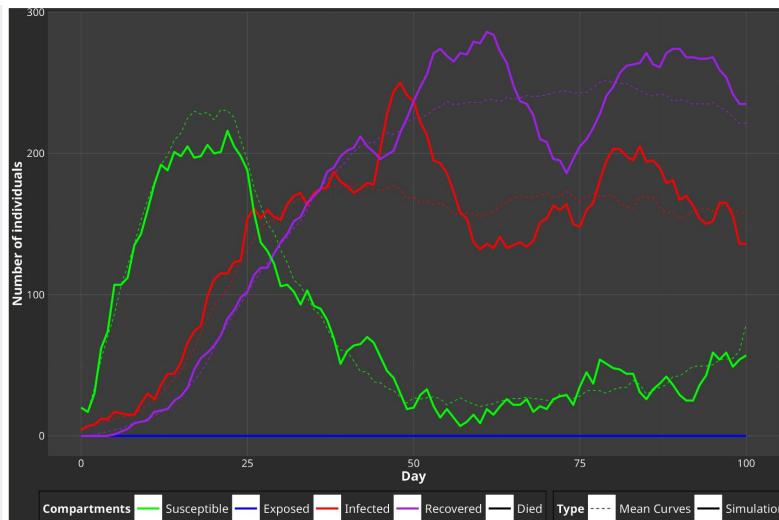
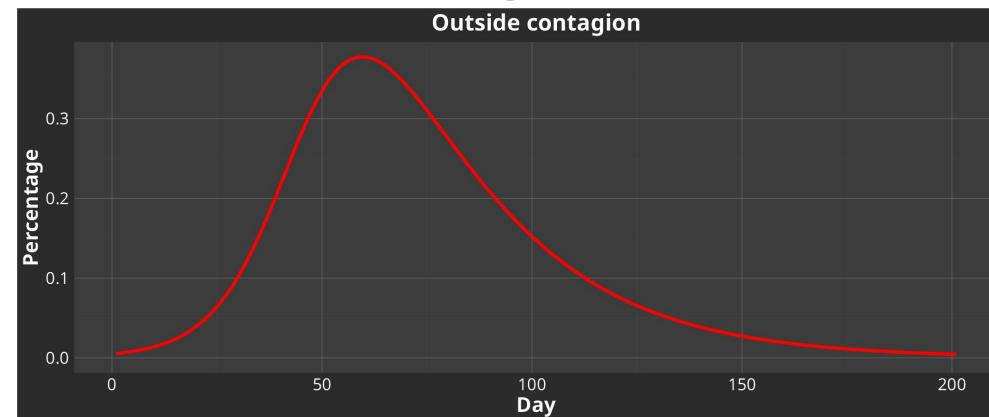
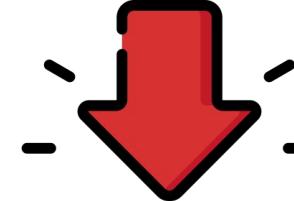
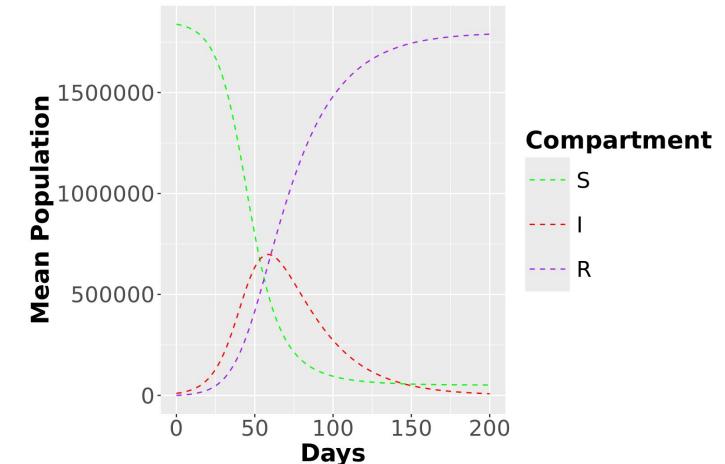
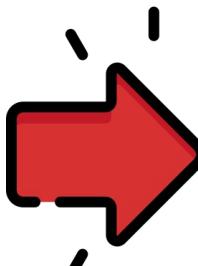
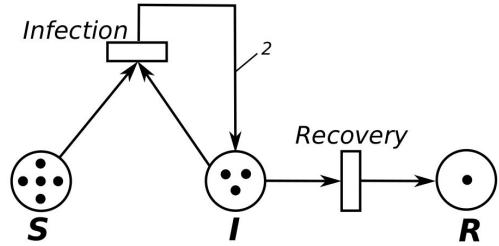
Small Scale



From macro to micro-level models



```
1 m;S;1838568;  
2 m;I;10000;  
3 m;R;0;  
4 c;Recovery; 0.04;  
5 c;Infection; 0.00000008;
```



Conclusion

Estimating Intervention Impact

Both macro and micro models help assess policies like lockdowns and vaccinations, each with their own strengths and limitations.

Individual Behavior Matters

Micro-level models offer detailed insights into how personal choices shape epidemic dynamics.

Simplicity vs. Detail

A key modeling challenge—striking the right balance depends on your goal: explanation, forecasting, or policy evaluation.

Choosing the Right Model

- **Macro models:** Best for broad trends, quick insights, low data requirements.
- **Micro models:** Ideal for detailed scenarios, heterogeneous behavior, and exploring emergent effects.



We hope you've gained a clearer understanding of when and how to use different computational models to study and respond to epidemics.



We'll be around during the entire conference!



Found bugs or have suggestions? Leave feedback on our GitHub.

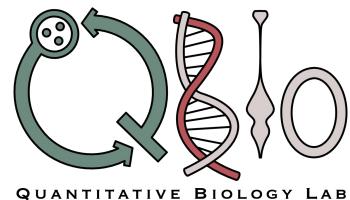
simone.pernice@unito.it irene.terrone@unito.it daniele.baccega@unito.it



UNIVERSITÀ
DI TORINO

DIPARTIMENTO
DI INFORMATICA

di.unito.it



QUANTITATIVE BIOLOGY LAB



IEEE - International Conference on
Healthcare Informatics
Università della Calabria, Italy
June 18th-21st, 2025

Thank you for the attention!



<https://qbio.di.unito.it/>

