



UNIVERSITÀ
DI TORINO

di.unito.it
DIPARTIMENTO
DI INFORMATICA



DAY 1

From Data to Models:

Analyzing and Integrating Biological Data into Mechanistic
Models

Course of Doctoral school
PhD in Complex Systems for Quantitative Biomedicine

Dora Tortarolo, Simone Pernice, Francesca Cordero

DAY 1: Outline

Day 1 – Introduction to Mechanistic Modeling and the Importance of Data

- Overview of computational models in biology.
- Mechanistic approaches: conceptual foundations of Petri Nets and the *epimod* framework.
- Hands-on Activity: the Schlögl model to understand system dynamics and sensitivity.

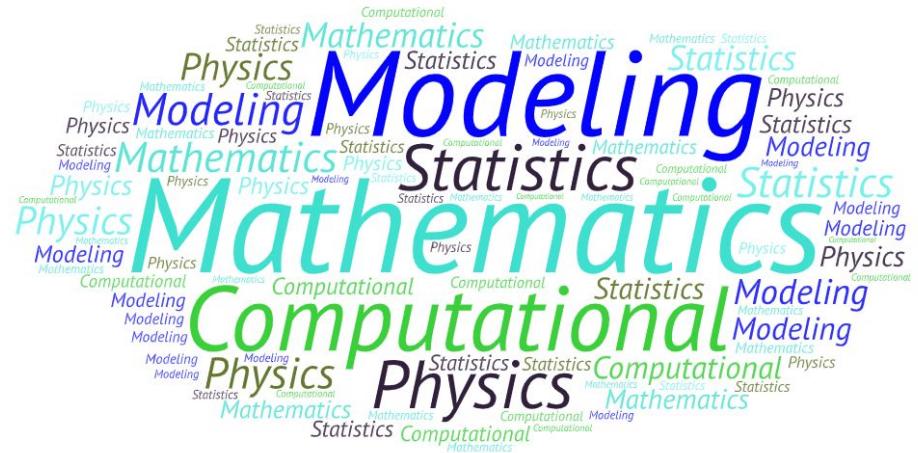


<https://github.com/qBioTurin/From-Data-to-Models.git>

Computational modeling

Definition

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

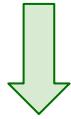


Applications in different fields:

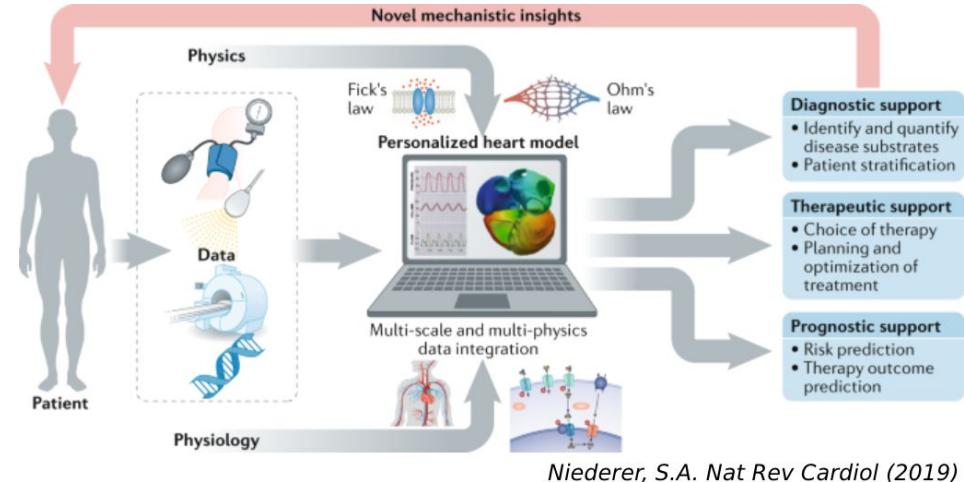
Network analysis, manufacturing systems, resource management, weather forecasting, clinical decision support, tracking infectious diseases, ...

Computational Modeling in Biological Systems

Computational modeling in biology is the process of using mathematical and computational approaches to represent, simulate, and analyze the behavior of the systems.



In the context of **complex biological systems**, it allows researchers to study how components such as genes, proteins, cells, or tissues interact over time and under various conditions.



Niederer, S.A. *Nat Rev Cardiol* (2019)

Computational Modeling in Biological Systems

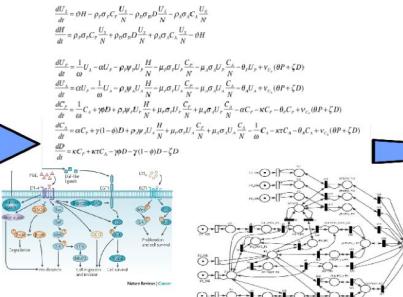
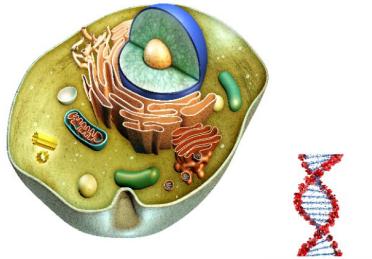


Abstraction of Biological Knowledge

- Translating biological processes into formal models

Multi-Scale Complexity

- From molecular (e.g., gene regulation) to cellular, tissue, and organism levels.
- Models must capture interactions across scales and levels of organization.

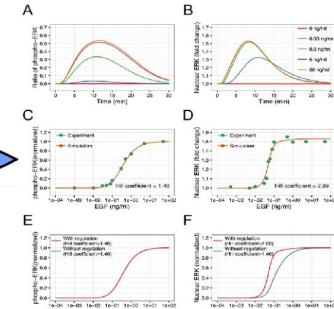


Integration of Experimental Data

- Models are dependent by quantitative data (e.g., from omics, imaging, or lab experiments).
- Enables **parameter estimation, validation, and prediction**.

Simulation and Prediction

- Allows exploration of system dynamics in silico.
- Enables **what-if scenarios**, hypothesis testing, and discovery of emergent behaviors.



Different modeling approaches

Statistical / Data-Driven Models

These models aim to uncover patterns, correlations, and predictions from data. They do **not explicitly model biological mechanisms**, and instead rely on data structure.

Exploratory: Identify structures or patterns in data without prediction (PCA, clustering)

Inferential: Use statistical inference to test hypotheses or estimate parameters (GLMs, ANOVA)

Machine Learning: Learn patterns from unlabeled/labeled data for classification/regression(SVM, RF, DL)

Mechanistic / Knowledge-Driven Models

These models are grounded in **explicit biological rules or hypotheses**, often formulated with equations or logic. They aim to explain causal behavior, not just correlations.

Interaction-based: network reconstruction processes that yield representations capturing structural information only (protein-protein interaction networks).

Mechanism-based:

- **Constraint-based (no dynamics):** commonly exploited to study systems under specific constraints (considered "mechanistic-lite" without dynamics)
- **Dynamical (with dynamics):** Models that represent explicit mechanistic hypotheses, including temporal dynamics.

Different modeling approaches



In this course, we will **focus exclusively on Mechanistic / Knowledge-Driven Models**, which aim to represent biological systems based on explicit, interpretable principles grounded in known biology.

This classification depends principally on:

- 1) the size of the model
- 2) the level of abstraction (amount of details)
- 3) the complexity

Mechanistic / Knowledge-Driven Models

These models are grounded in **explicit biological rules or hypotheses**, often formulated with equations or logic. They aim to explain causal behavior, not just correlations.

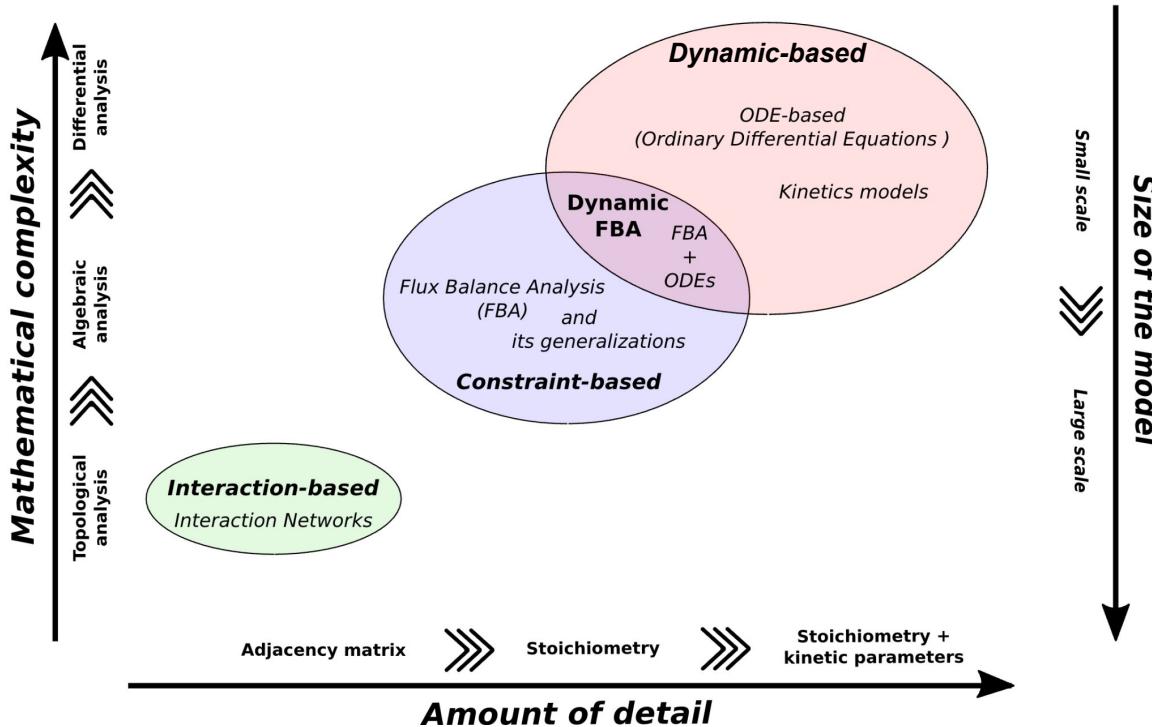
1. Interaction-based: network reconstruction processes that yield representations capturing structural information only (protein-protein interaction networks).

2. Mechanism-based:

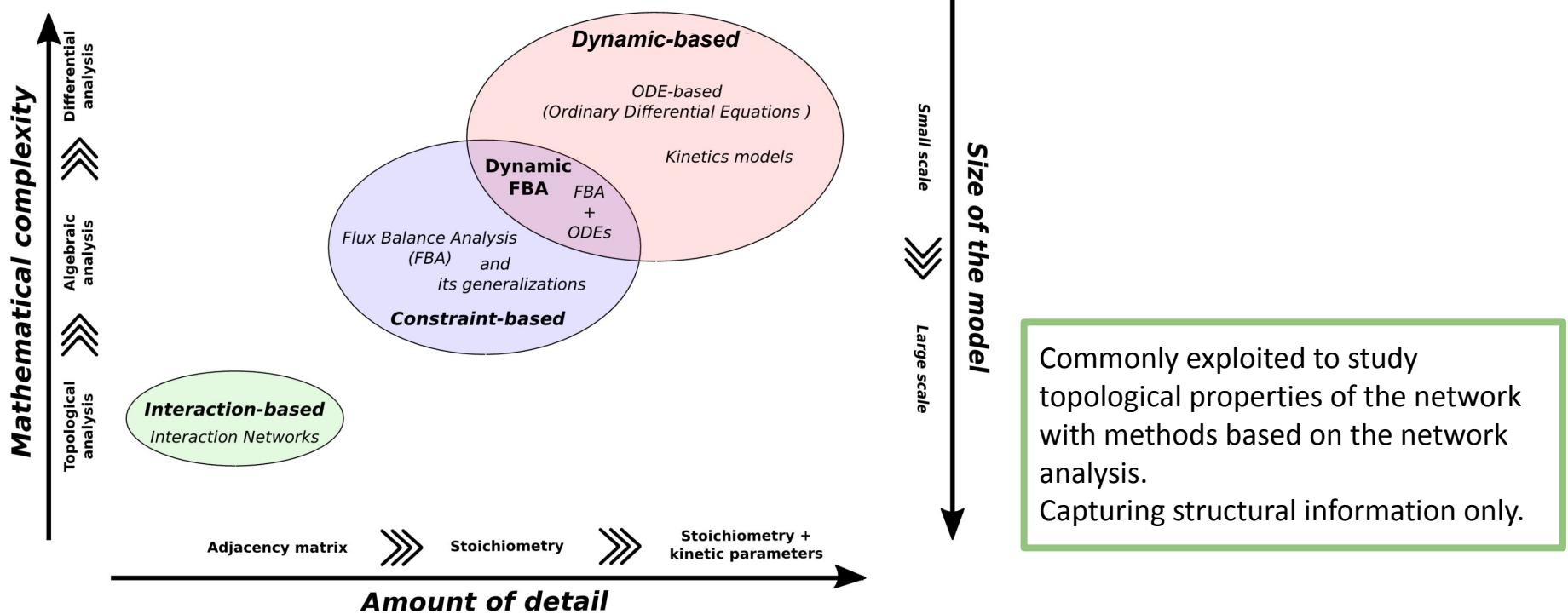
- **Constraint-based (no dynamics):** commonly exploited to study systems under specific constraints (considered "mechanistic-lite" without dynamics)
- **Dynamical (with dynamics):** Models that represent explicit mechanistic hypotheses, including temporal dynamics.



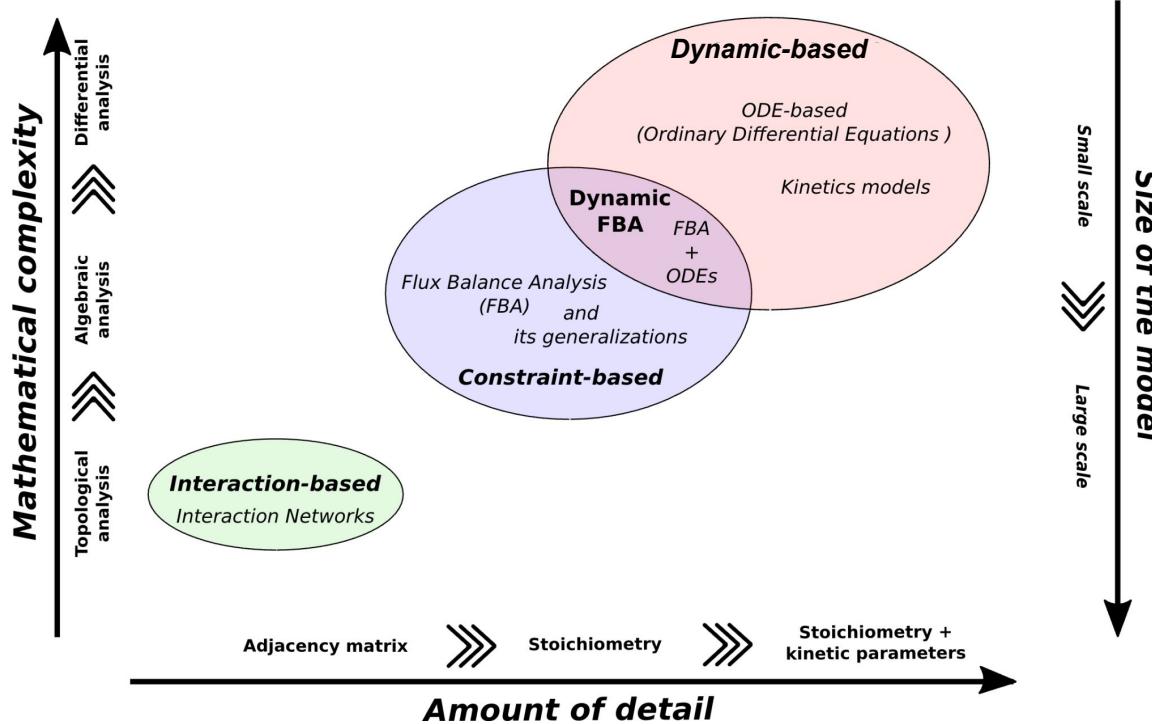
Mechanistic modeling approaches



Different modeling approaches

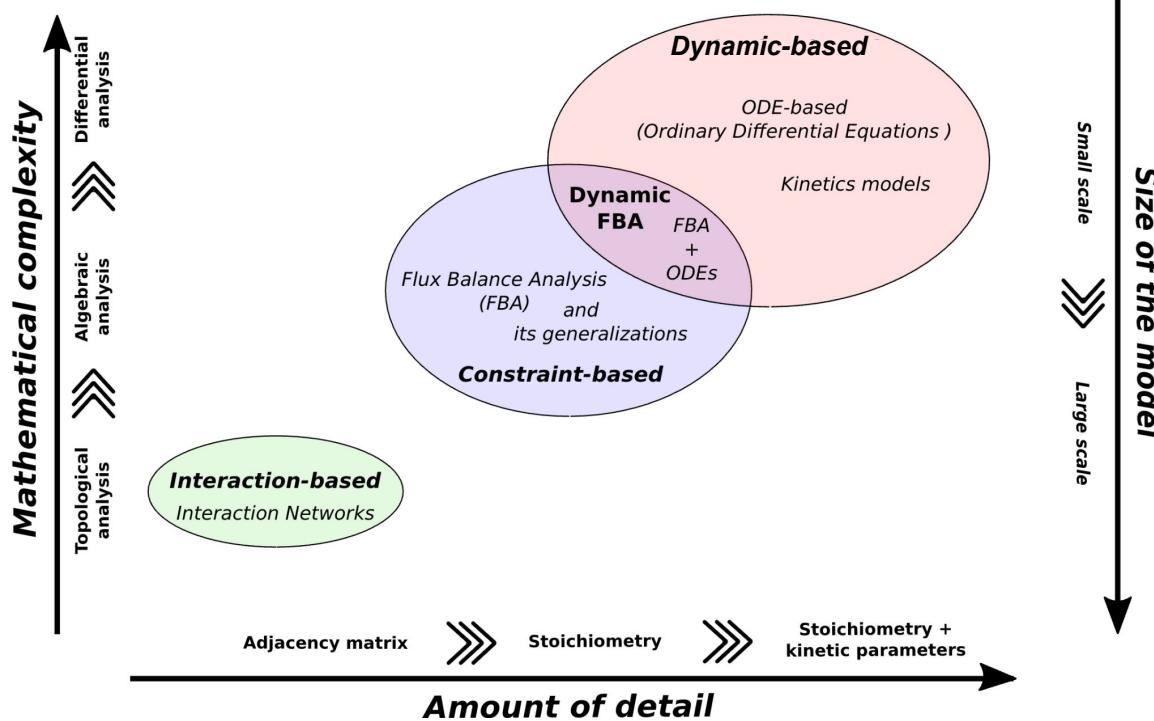


Different modeling approaches



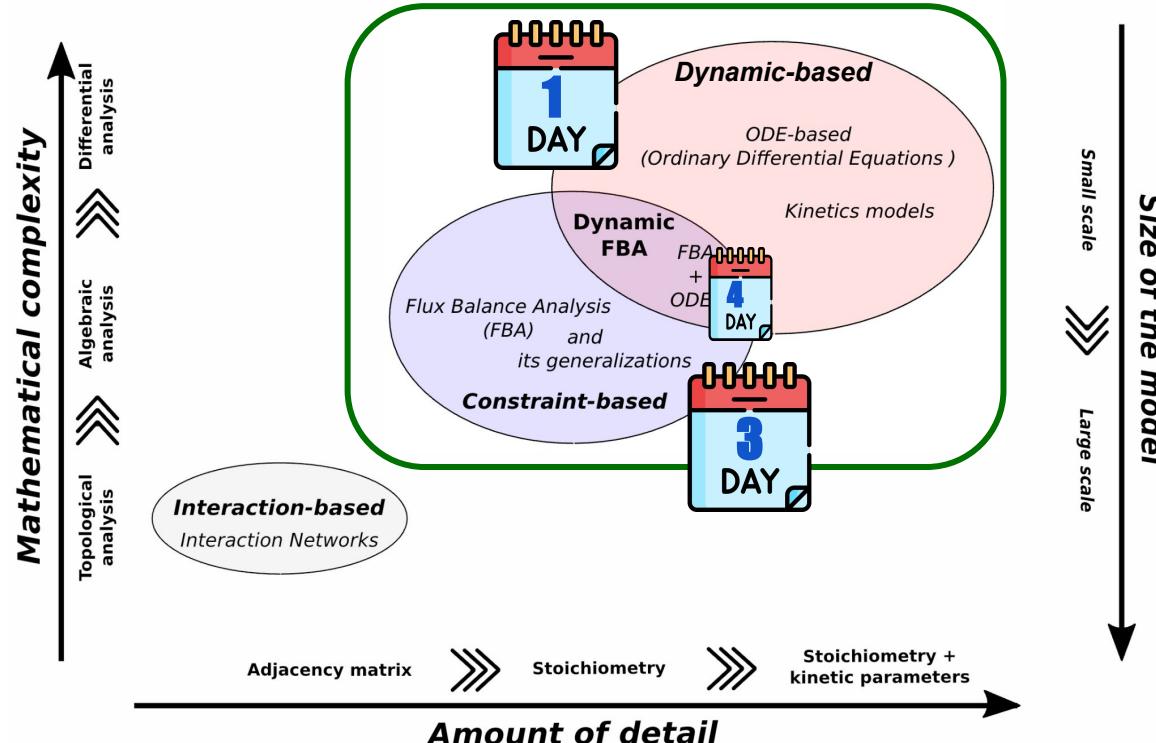
Commonly exploited to study systems under specific constraints. Flux Balance Analysis computes reaction fluxes constrained in a range of values, which minimize/maximize an objective function in a system at the equilibrium. Capturing fluxes information in the steady state.

Different modeling approaches



Commonly exploited to study the dynamics of the system. Requiring a large amount of details to capturing a complete understanding of the system behaviour.

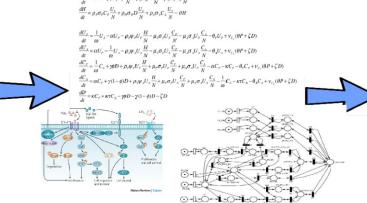
Different modeling approaches



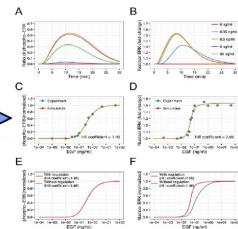
Dynamic based modeling approaches



→



→



Dynamic based modeling approaches

- A dynamic model describes **how the state of a system evolves over time**
- Based on:
 - **Biological assumptions** (e.g., transcription leads to translation)
 - **Mathematical representation** (e.g., equations, rules, simulations)

State variables: concentrations, cell counts, gene activity

Dynamics = How things change.

Mechanisms = Why they change

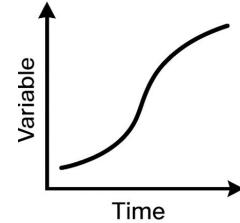
Dynamic based modeling approaches

Once we have the model defined, we can go in **two directions**:



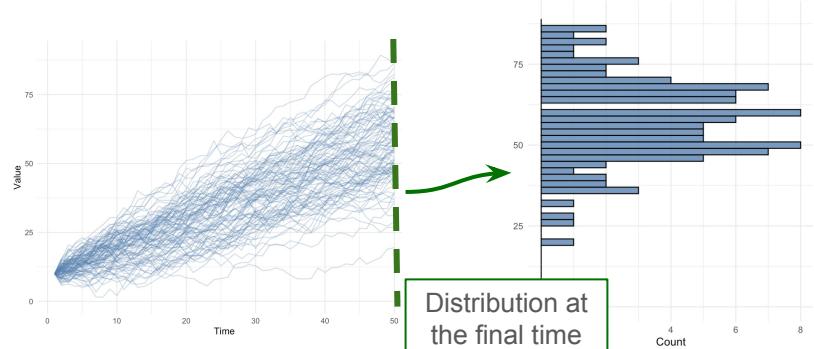
Deterministic approach: we assume the population is large enough that randomness averages out. This leads to **ordinary differential equations (ODEs)**

One input \longrightarrow One dynamic



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

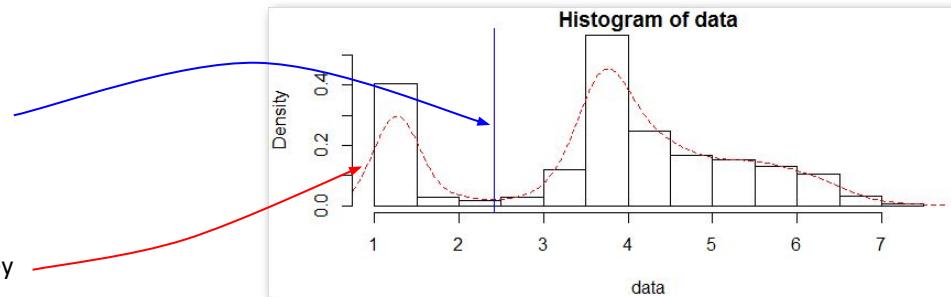
One input \longrightarrow Multiple dynamics



Dynamic based modeling approaches

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 models

Imagine a **small biochemical system** where molecules react randomly. Because molecules are discrete and reactions are probabilistic, you can't predict exactly *what will happen*, only *what's likely to happen*.

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

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

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

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

Stochastic models

Imagine a **small biochemical system** where molecules react randomly. Because molecules are discrete and reactions are probabilistic, you can't predict exactly *what will happen*, only *what's likely to happen*.

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

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

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

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

We're looking at all the ways the system can enter or leave state s_i

The sum adds up the probability flow from every other state s_k to state s_i

Stochastic models

Imagine a **small biochemical system** where molecules react randomly. Because molecules are discrete and reactions are probabilistic, you can't predict exactly *what will happen*, only *what's likely to happen*.

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

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

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

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

We're looking at all the ways the system can enter or leave state s_i

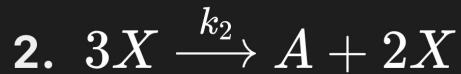
The sum adds up the probability flow from every other state s_k to state s_i



You're **not tracking molecules**, you're tracking the **probability of being in each state**.

Example: Shlogel model

Let's walk through an **intuitive explanation of the Kolmogorov equation** using the **Schlögl model**



There is a single free component, X, and two reservoirs, A and B, assumed constant.

Here, A and B are constant so the system's state is fully described by the **number of X molecules** at time t, say x.

Goal of the Kolmogorov Equation:

What is the probability $P(x,t)$ that there are exactly x molecules of X at time t?

Say $x=20$ molecules of X.

We want to compute how the probability $P(20,t)$ changes

Example: Shlogel model

Let's walk through an **intuitive explanation of the Kolmogorov equation** using the **Schlögl model**



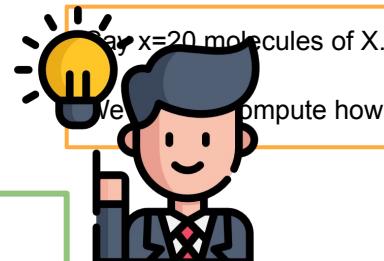
There is a single free component, X, and two reservoirs, A and B, assumed constant.

Here, A and B are constant so the system's state is fully described by the **number of X molecules** at time t, say x.

Goal of the Kolmogorov Equation:

What is the probability $P(x,t)$ that there are exactly x molecules of X at time t?

SINCE We're looking at all the ways the system can enter or leave state $x=20$
THEN We have to identify these "ways"



say $x=20$ molecules of X.

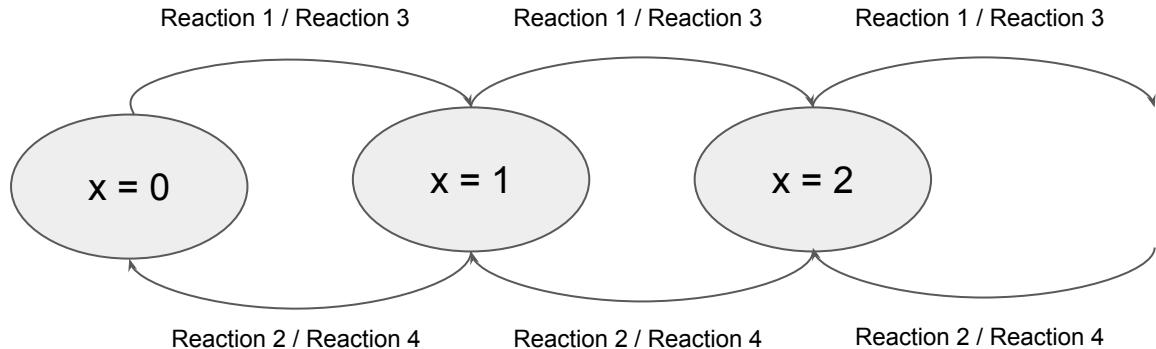
We want to compute how the probability $P(20,t)$ changes

Example: Shlogel model

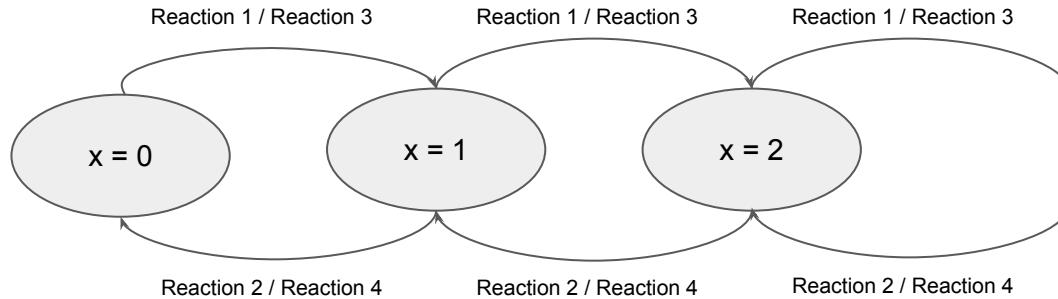
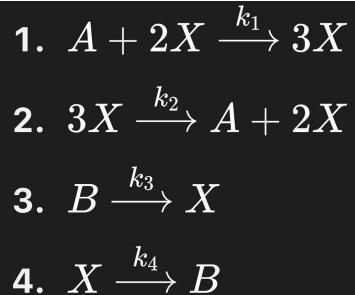
Here, A and B are constant so the system's state is fully described by the **number of X molecules** at time t, say x.

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$

Each reaction changes x by ± 1



Example: Shlogel model



Say $x=20$ molecules of X.

We want to compute how the probability $P(20,t)$ changes:

1. **Enter state 20** (e.g., from state 19 or 21)
2. **Leave state 20** (e.g., to state 21 or 19)

INTO state 20 (increasing $P(20,t)$)

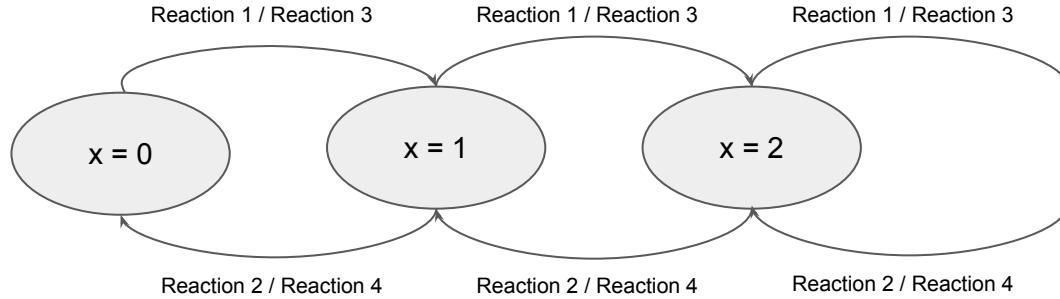
- From $x=19$ via a reaction that **adds 1 molecule** (e.g., $B \rightarrow X$)
- From $x=21$ via a reaction that **removes 1 molecule** (e.g., $X \rightarrow B$)

OUT OF state 20 (decreasing $P(20,t)$)

- From $x=20$ to 21 via a reaction that adds 1 molecule
- From $x=20$ to 19 via a reaction that removes 1 molecule

Example: Shlogel model

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$



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



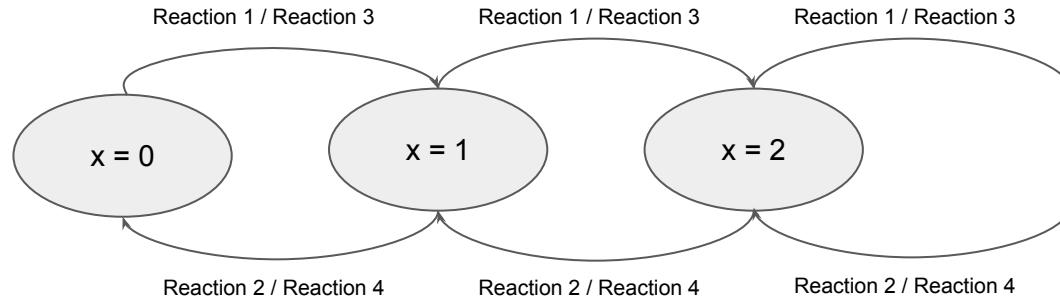
$$\frac{d\pi(20, t)}{dt} = \pi(19, t)q_{19,20} + \pi(21, t)q_{21,20} + \pi(20, t)q_{20,20}$$

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

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

Example: Shlogel model

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$



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



$$\frac{d\pi(20, t)}{dt} = \pi(19, t)q_{19,20} + \pi(21, t)q_{21,20} + \pi(20, t)q_{20,20}$$

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

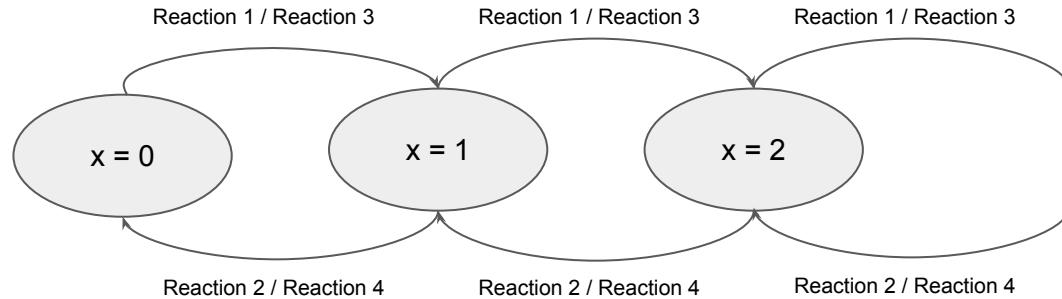
$q_{k,i}$ = rate from s_k to s_i



Why do not have $\pi(18, t)q_{18,20}$?

Example: Shlogel model

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$



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



$$\frac{d\pi(20, t)}{dt} = \pi(19, t)q_{19,20} + \pi(21, t)q_{21,20} + \pi(20, t)q_{20,20}$$

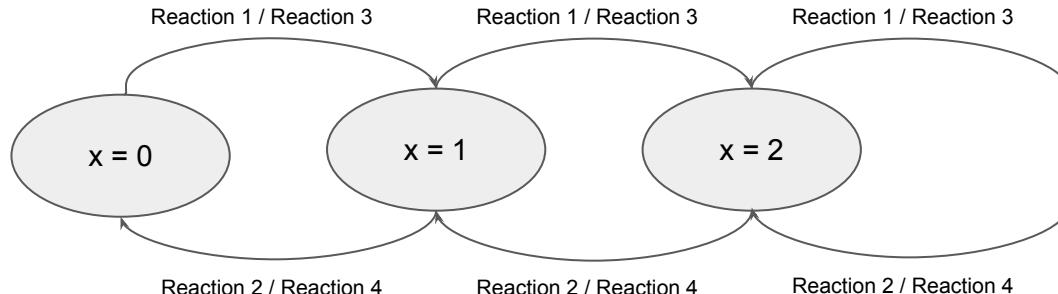
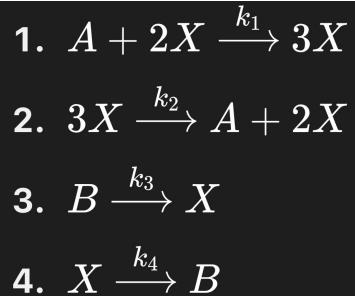
$\pi(s_i, \nu)$ = probability to be in the state s_i at time ν

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



What are the transition rates???

Example: Shlogel model



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

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

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

Propensity functions focus on the rate of a specific event, while transition rates focus on the speed of moving between states.

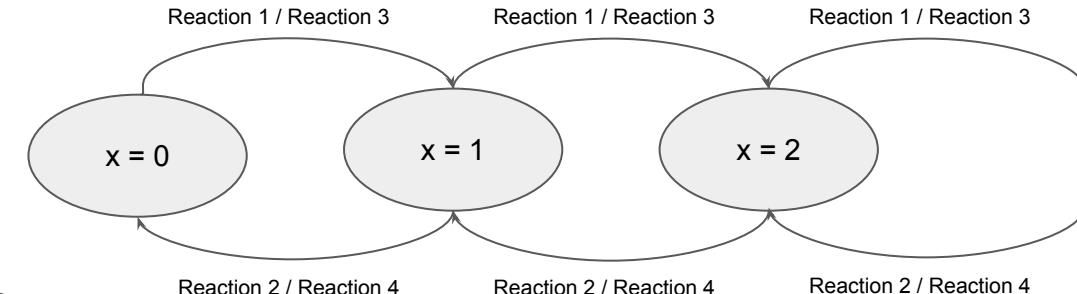
The transition rate, often denoted as $q_{k,i}$, represents the speed at which a system transitions from one state (s_k) to another (s_i).

In many stochastic models, the transition rate is derived from the **propensity function**.

A propensity function, often denoted as $a(x)$, represents the rate at which an event occurs, given the current state x . It essentially describes the "tendency" or "propensity" for a particular event to happen.

Example: Shlogel model

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$



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

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

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

Let's define the propensity functions as $a_j(x)$ where $j=1,2,3,4$ and x is the current number of X molecules.

So, for example:

$$q_{x \rightarrow x+1} = a_1(x) + a_3(x)$$

$$q_{x \rightarrow x-1} = a_2(x) + a_4(x)$$

Example: Shlogel model

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

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

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

$$q_{x \rightarrow x+1} = a_1(x) + a_3(x)$$

$$q_{x \rightarrow x-1} = a_2(x) + a_4(x)$$

Each reaction has a **rate constant**, and the **propensity function** combines that with the **number of possible reactant combinations** at state x.

$$a_j(x) = k_j(\text{number of reactant combinations at state } x)$$

Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

Example: Shlogel model

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

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

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

$$q_{x \rightarrow x+1} = a_1(x) + a_3(x)$$

$$q_{x \rightarrow x-1} = a_2(x) + a_4(x)$$

Each reaction has a **rate constant**, and the **propensity function** combines that with the **number of possible reactant combinations** at state x.

$$a_j(x) = k_j(\text{number of reactant combinations at state } x)$$

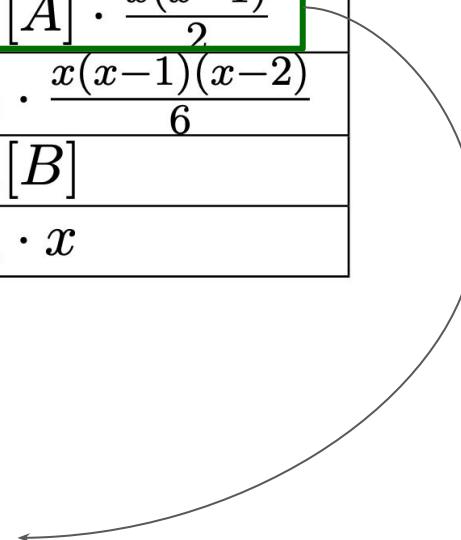
Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

k_1, k_2, k_3, k_4 are rate constants

Example: Shlogel model

Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

- This reaction needs **2 molecules of X** and **1 molecule of A** at the same time
- $x(x-1)/2$ counts **how many pairs** of X can react.
- The more X you have, the **faster** the reaction happens.
- $[A]$ is constant, so it just scales the rate.

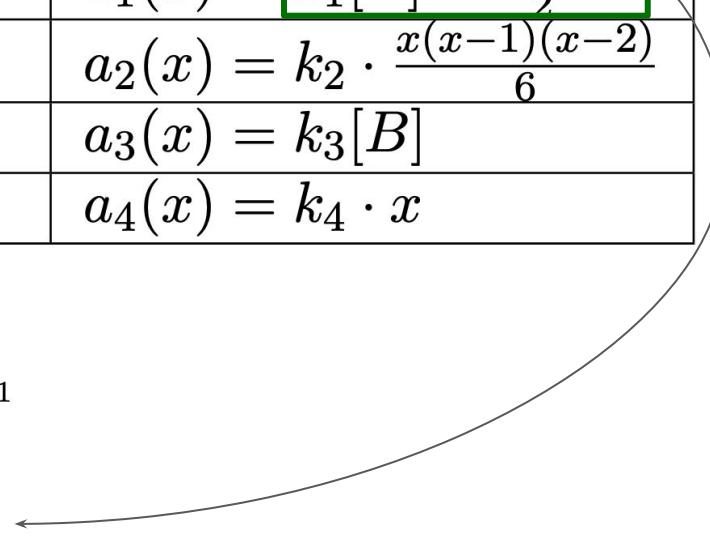


Example: Shlogel model

Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

k_1 is in units like $[molecules]^{-1} \times time^{-1}$

The count $\frac{x(x-1)}{2}$ is dimensionless



So $a_1(x)$ has units of $\frac{1}{time}$

Example: Shlogel model

Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

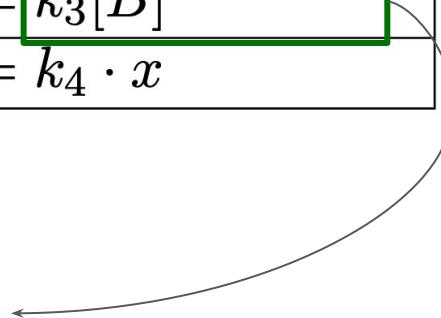
Needs 3 molecules of X at once.

The count $\frac{x(x-1)(x-2)}{6}$ counts the number of triplets of X that can react.

Example: Shlogel model

Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

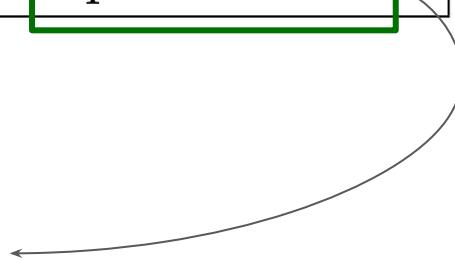
- Happens at a **constant rate**, no matter how many X you have.
- It just adds new X molecules — like an **injection** of X.



Example: Shlogel model

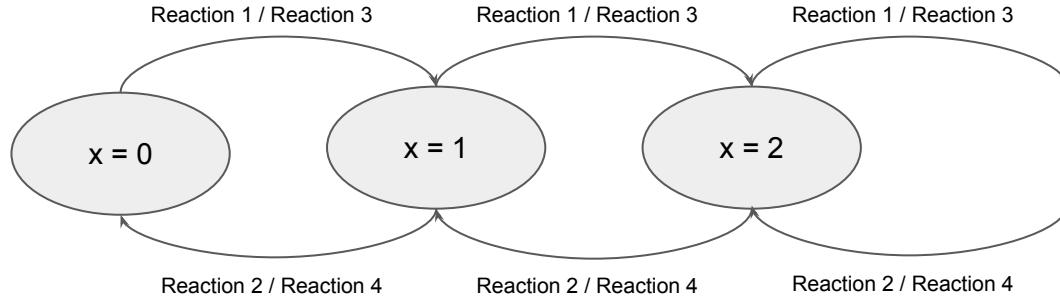
Reaction	Change in x	Propensity $a_j(x)$
$A + 2X \rightarrow 3X$	+1	$a_1(x) = k_1[A] \cdot \frac{x(x-1)}{2}$
$3X \rightarrow A + 2X$	-1	$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6}$
$B \rightarrow X$	+1	$a_3(x) = k_3[B]$
$X \rightarrow B$	-1	$a_4(x) = k_4 \cdot x$

- Each X molecule has the **same chance** of disappearing.
- So if you have more X, this happens **more often**.



Example: Shlogel model

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$



$$\frac{d\pi(0,t)}{dt} = \pi(1,t) \cdot q_{1 \rightarrow 0} + \pi(0,t) \cdot q_{0 \rightarrow 0}$$

...

$$\frac{d\pi(19,t)}{dt} = \pi(18,t) \cdot q_{18 \rightarrow 19} + \pi(20,t) \cdot q_{20 \rightarrow 19} + \pi(19,t) \cdot q_{19 \rightarrow 19}$$

$$\frac{d\pi(20,t)}{dt} = \pi(19,t) \cdot q_{19 \rightarrow 20} + \pi(21,t) \cdot q_{21 \rightarrow 20} + \pi(20,t) \cdot q_{20 \rightarrow 20}$$

$$\frac{d\pi(21,t)}{dt} = \pi(20,t) \cdot q_{20 \rightarrow 21} + \pi(22,t) \cdot q_{22 \rightarrow 21} + \pi(21,t) \cdot q_{21 \rightarrow 21}$$

...

$$q_{x \rightarrow x+1} = a_1(x) + a_3(x) \quad (\text{production reactions})$$

$$q_{x \rightarrow x-1} = a_2(x) + a_4(x) \quad (\text{degradation reactions})$$

$$q_{x \rightarrow x} = -(q_{x \rightarrow x+1} + q_{x \rightarrow x-1})$$

$$a_1(x) = k_1 \cdot [A] \cdot \frac{x(x-1)}{2} \quad (2X + A \rightarrow 3X)$$

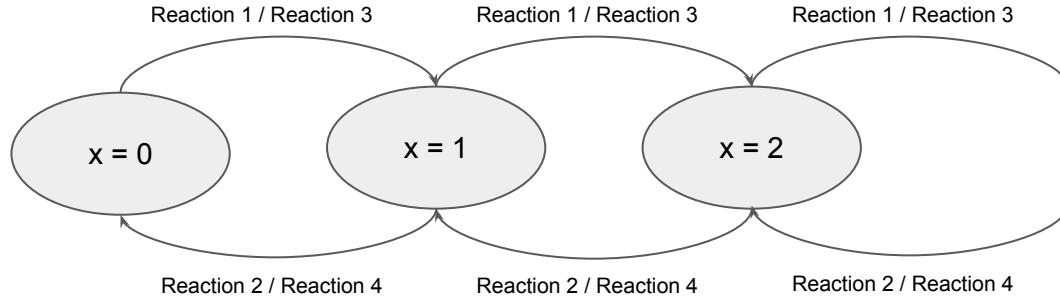
$$a_2(x) = k_2 \cdot \frac{x(x-1)(x-2)}{6} \quad (3X \rightarrow 2X + A)$$

$$a_3(x) = k_3 \cdot [B] \quad (B \rightarrow X)$$

$$a_4(x) = k_4 \cdot x \quad (X \rightarrow B)$$

Example: Shlogel model

1. $A + 2X \xrightarrow{k_1} 3X$
2. $3X \xrightarrow{k_2} A + 2X$
3. $B \xrightarrow{k_3} X$
4. $X \xrightarrow{k_4} B$



$$\frac{d\pi(0,t)}{dt} = \pi(1,t) \cdot q_{1 \rightarrow 0} + \pi(0,t) \cdot q_{0 \rightarrow 0}$$

...

$$\frac{d\pi(19,t)}{dt} = \pi(18,t) \cdot q_{18 \rightarrow 19} + \pi(20,t) \cdot q_{20 \rightarrow 19} + \pi(19,t) \cdot q_{19 \rightarrow 20}$$

$$\frac{d\pi(20,t)}{dt} = \pi(19,t) \cdot q_{19 \rightarrow 20} + \pi(21,t) \cdot q_{21 \rightarrow 20} + \pi(20,t) \cdot q_{20 \rightarrow 21}$$

$$\frac{d\pi(21,t)}{dt} = \pi(20,t) \cdot q_{20 \rightarrow 21} + \pi(22,t) \cdot q_{22 \rightarrow 21} + \pi(21,t) \cdot q_{21 \rightarrow 22}$$

...



$$q_{x \rightarrow x+1} = a_1(x) + a_3(x) \quad (\text{production reactions})$$

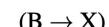
$$q_{x \rightarrow x-1} = a_2(x) + a_4(x) \quad (\text{degradation reactions})$$

$$q_{x \rightarrow x} = -(q_{x \rightarrow x+1} + q_{x \rightarrow x-1})$$

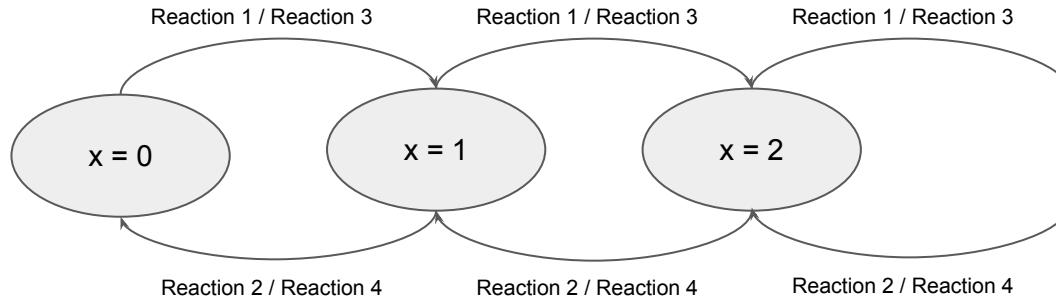
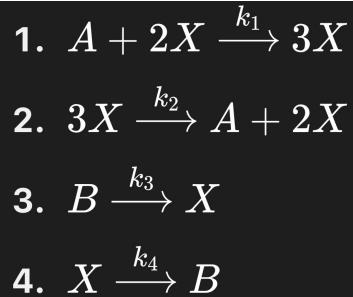
How many equations?

$$a_3(x) = k_3 \cdot [B]$$

$$a_4(x) = k_4 \cdot x$$



Example: Shlogel model



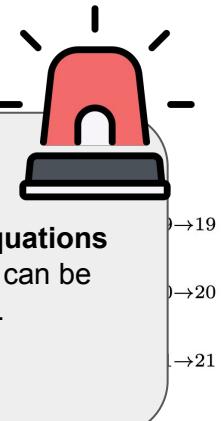
$$\frac{d\pi(0,t)}{dt} = \pi(1,t) \cdot q_{1 \rightarrow 0} + \pi(0,t) \cdot q_{0 \rightarrow 0}$$

$$\frac{d\pi(19)}{dt}$$

$$\frac{d\pi(20)}{dt}$$

$$\frac{d\pi(21)}{dt}$$

The number of **Kolmogorov equations** depends on how many **states** can be reached in your system.



→19
→20
→21

$$q_{x \rightarrow x+1} = a_1(x) + a_3(x) \quad (\text{production reactions})$$

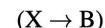
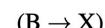
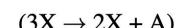
$$q_{x \rightarrow x-1} = a_2(x) + a_4(x) \quad (\text{degradation reactions})$$

$$q_{x \rightarrow x} = -(q_{x \rightarrow x+1} + q_{x \rightarrow x-1})$$

How many equations?

$$a_3(x) = k_3 \cdot [B]$$

$$a_4(x) = k_4 \cdot x$$



Kolmogorov equations

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



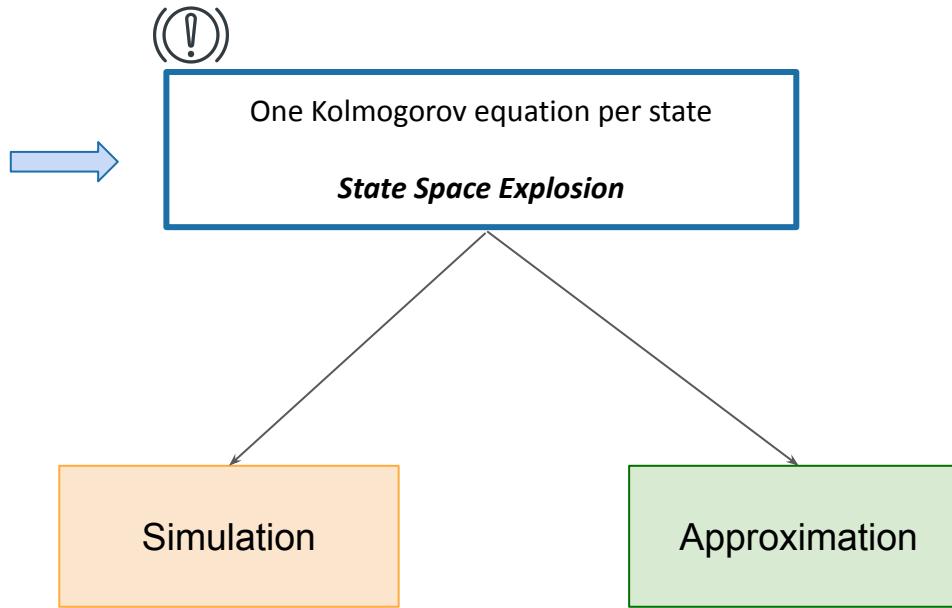
Kolmogorov equations

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



- [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*

Kolmogorov equations

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

Simulation

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

- [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*

Kolmogorov equations

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

Simulation

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

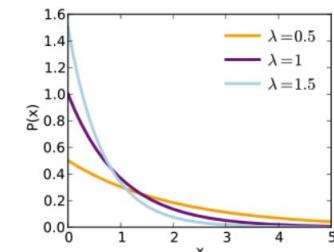


Key concept

The total **waiting time** until *any* reaction occurs is modeled as an **exponentially distributed random variable**.

$$\tau \sim \text{Exponential } (a_0(x)) \quad \text{where} \quad a_0(x) = \sum_r a_r(x)$$

Probability density function



The **more active** the system is (the higher the total propensity), the **sooner** the next event is likely to happen.

Events are **memoryless** — the probability of something happening in the next second doesn't depend on how long you've already waited.

- [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*

Kolmogorov equations

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

Simulation

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

Given:

- initial state (initial numbers of molecules of each species in the system)
- a set of chemical reactions

Gillespie's algorithm computes a possible evolution of the system

SSA define a system as a **random process** that evolves in time, one event at a time.

At each step, two things happen:

1. You **decide when** the next event occurs.
2. You **decide which** event occurs.

Kolmogorov equations

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

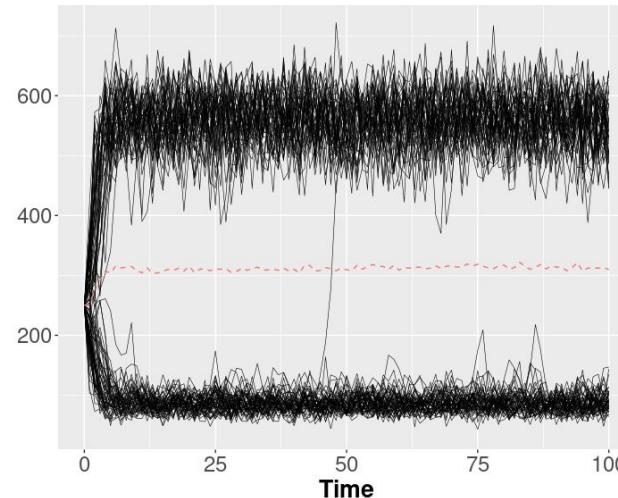
Simulation

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

Given:

- initial state (initial numbers of molecules of each species in the system)
- a set of chemical reactions

Gillespie's algorithm computes a possible evolution of 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*

Kolmogorov equations

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



Longer simulation with an increasing number of events!!

Kolmogorov equations

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**.

The idea is to allow **several events to take place in a single** (longer) time step, under the condition that event rates do not change too much during that time.

This way, instead of simulating each event one by one, we **approximate how many of each event happen** within the time step using a random count based on their rates.

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



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

Longer simulation with an increasing number of events!!

Kolmogorov equations

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**.

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!!



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*

Deterministic Approximation



Deterministic
Approximation



One Ordinary Differential Equation (ODE) per variable that define the state of the system!

$$\frac{d\pi(0,t)}{dt} = \pi(1,t) \cdot q_{1 \rightarrow 0} + \pi(0,t) \cdot q_{0 \rightarrow 0}$$

...

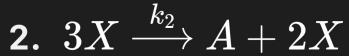
$$\frac{d\pi(19,t)}{dt} = \pi(18,t) \cdot q_{18 \rightarrow 19} + \pi(20,t) \cdot q_{20 \rightarrow 19} + \pi(19,t) \cdot q_{19 \rightarrow 19}$$

$$\frac{d\pi(20,t)}{dt} = \pi(19,t) \cdot q_{19 \rightarrow 20} + \pi(21,t) \cdot q_{21 \rightarrow 20} + \pi(20,t) \cdot q_{20 \rightarrow 20}$$

$$\frac{d\pi(21,t)}{dt} = \pi(20,t) \cdot q_{20 \rightarrow 21} + \pi(22,t) \cdot q_{22 \rightarrow 21} + \pi(21,t) \cdot q_{21 \rightarrow 21}$$

...

Deterministic Approximation



One Ordinary Differential Equation (ODE) per variable that define the system's state!

A and B are constant so the system's state is fully described by the **number of X molecules** at time t

$$\frac{d\pi(0,t)}{dt} = \pi(1,t) \cdot q_{1 \rightarrow 0} + \pi(0,t) \cdot q_{0 \rightarrow 1}$$

...

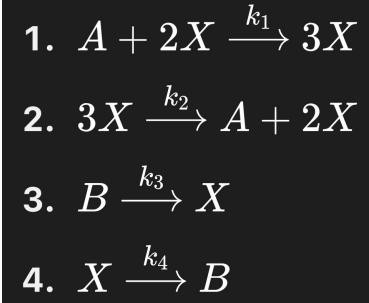
$$\frac{d\pi(19,t)}{dt} = \pi(18,t) \cdot q_{18 \rightarrow 19} + \pi(20,t) \cdot q_{20 \rightarrow 19} + \pi(19,t) \cdot q_{19 \rightarrow 19}$$

$$\frac{d\pi(20,t)}{dt} = \pi(19,t) \cdot q_{19 \rightarrow 20} + \pi(21,t) \cdot q_{21 \rightarrow 20} + \pi(20,t) \cdot q_{20 \rightarrow 20}$$

$$\frac{d\pi(21,t)}{dt} = \pi(20,t) \cdot q_{20 \rightarrow 21} + \pi(22,t) \cdot q_{22 \rightarrow 21} + \pi(21,t) \cdot q_{21 \rightarrow 21}$$

...

Deterministic Approximation



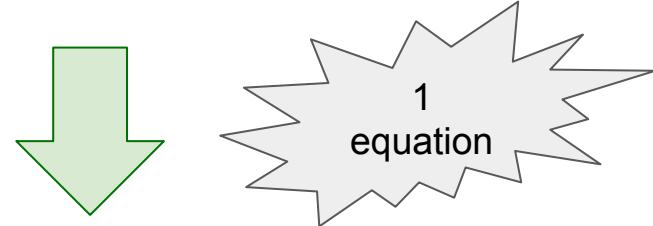
One Ordinary Differential Equation (ODE) per variable that define the system's state!

A and B are constant so the system's state is fully described by the **number of X molecules** at time t

$$\frac{d\pi(0, t)}{dt} = \pi(1, t) \cdot q_{1 \rightarrow 0} + \pi(0, t) \cdot q_{0 \rightarrow 1}$$

...

$$\begin{aligned}\frac{d\pi(19, t)}{dt} &= \pi(18, t) \cdot q_{18 \rightarrow 19} + \pi(20, t) \cdot q_{20 \rightarrow 19} + \pi(19, t) \cdot q_{19 \rightarrow 19} \\ \frac{d\pi(20, t)}{dt} &= \pi(19, t) \cdot q_{19 \rightarrow 20} + \pi(21, t) \cdot q_{21 \rightarrow 20} + \pi(20, t) \cdot q_{20 \rightarrow 20} \\ \frac{d\pi(21, t)}{dt} &= \pi(20, t) \cdot q_{20 \rightarrow 21} + \pi(22, t) \cdot q_{22 \rightarrow 21} + \pi(21, t) \cdot q_{21 \rightarrow 21} \\ &\dots\end{aligned}$$

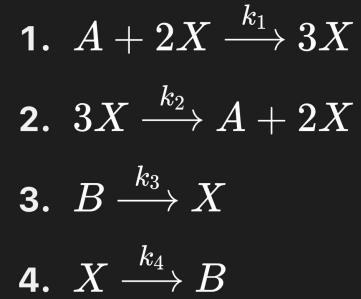


$$\frac{dx}{dt} = (+1) \cdot a_1(x) + (-1) \cdot a_2(x) + (+1) \cdot a_3(x) + (-1) \cdot a_4(x)$$

Substituting:

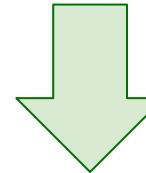
$$\frac{dx}{dt} = k_1 a \frac{x(x-1)}{2} - k_2 \frac{x(x-1)(x-2)}{6} + k_3 b - k_4 x$$

Deterministic Approximation



One Ordinary Differential Equation (ODE) per variable that define the system's state!

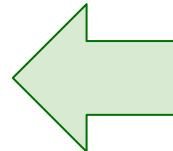
A and B are constant so the system's state is fully described by the **number of X molecules** at time t



When the number of molecules is large, stochastic fluctuations average out, and the **expected concentration X** evolves smoothly.

$$\frac{x(x-1)}{2} \approx \frac{x^2}{2}, \quad \frac{x(x-1)(x-2)}{6} \approx \frac{x^3}{6}$$

$$\frac{dx}{dt} = \frac{k_1}{2}ax^2 - \frac{k_2}{6}x^3 + k_3b - k_4x$$



$$\frac{dx}{dt} = (+1) \cdot a_1(x) + (-1) \cdot a_2(x) + (+1) \cdot a_3(x) + (-1) \cdot a_4(x)$$

Substituting:

$$\frac{dx}{dt} = k_1a\frac{x(x-1)}{2} - k_2\frac{x(x-1)(x-2)}{6} + k_3b - k_4x$$

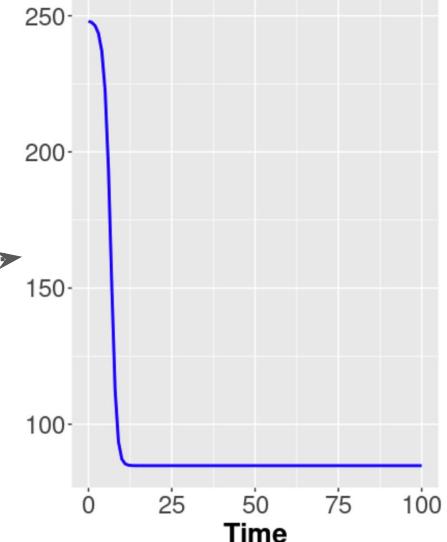
Deterministic Approximation



$$\frac{dx}{dt} = \frac{k_1}{2}ax^2 - \frac{k_2}{6}x^3 + k_3b - k_4x$$

Solving (integrating)
the equations

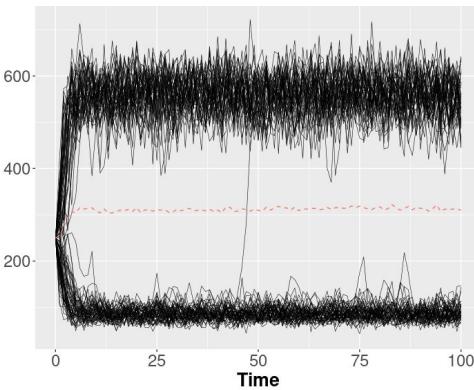
This is the classical form of the Schlögl model ODE



Dynamic-based models

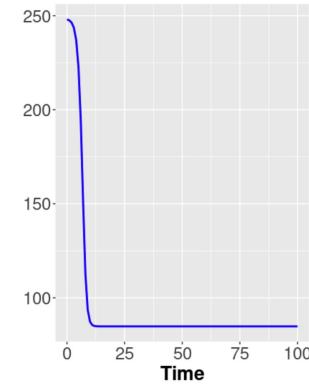


Dynamic-based models



Simulation

Approximation



*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

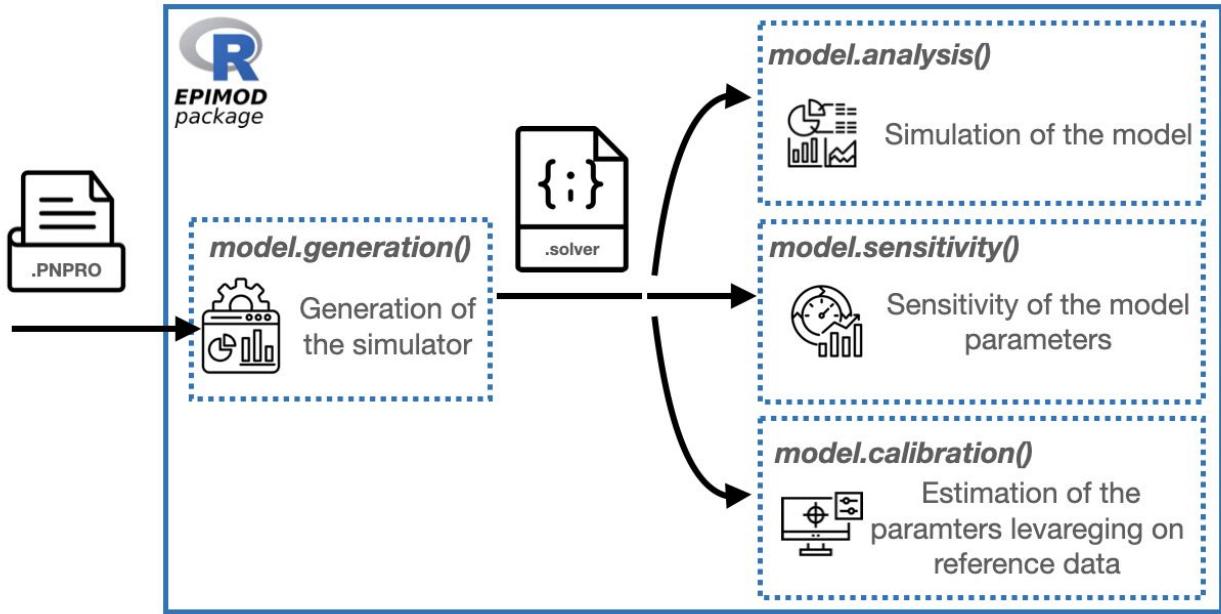
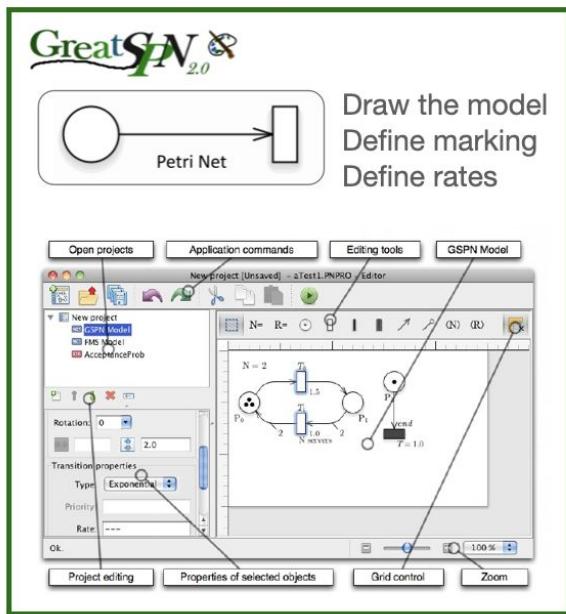
Epimod Library:

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

Docker Containers:

- Simulations reproducibility
- Simplifies deployment, updates, and maintenance

GreatMod



Petri Net: graphical formalism to draw model

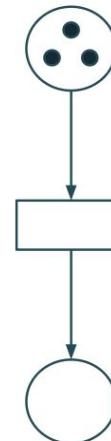
Petri Net (PN) is graphical formalism (bipartite graph), conveniently used for the analysis of complex models. It allows us to derive qualitative and quantitative properties of the system.

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

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

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

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



Petri Net: graphical formalism to draw model

Arcs Multiplicity:

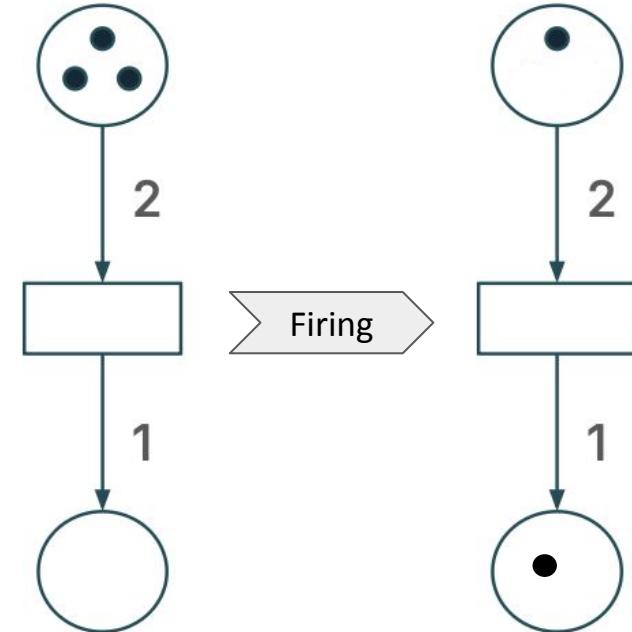
- Arcs can have multiplicities, indicating the number of tokens transferred between places and transitions.

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

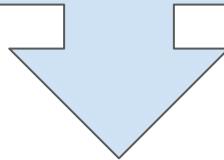
Firing Rate:

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



Petri Net: graphical formalism to draw model

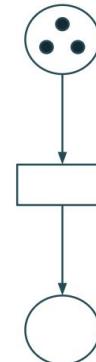
Exponentially distributed random delays are associated with transition firings (time to wait before the tokens are removed from/ added to places).



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



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

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

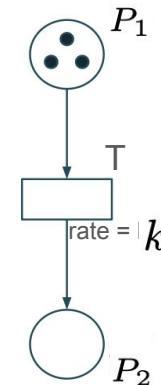
$q_{k,i}$ = rate from s_k to s_i

Transition velocities

The transition firings velocity is derived from the **propensity function!**

Mass Action (The Default)

- **Mass Action kinetics** assumes that transition velocities are **proportional** to the number of **available tokens** from the **input places**.
- This is a **simplifying assumption**, often valid when:
 - All molecules are well-mixed.
 - Transitions occur freely without spatial/structural constraints.



$$a_T(x) = x_{P_1} * k$$

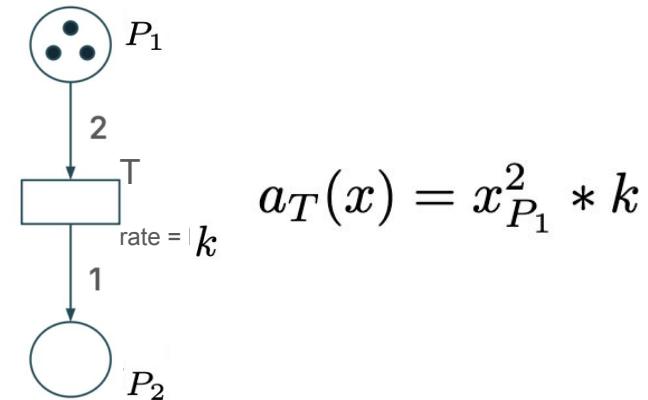
$$x = [x_{P_1}, x_{P_2}] \longrightarrow \text{The number of tokens in each place}$$

Transition velocities

The transition firings velocity is derived from the **propensity function!**

Mass Action (The Default)

- **Mass Action kinetics** assumes that transition velocities are **proportional** to the number of **available tokens** from the **input places**.
- This is a **simplifying assumption**, often valid when:
 - All molecules are well-mixed.
 - Transitions occur freely without spatial/structural constraints.



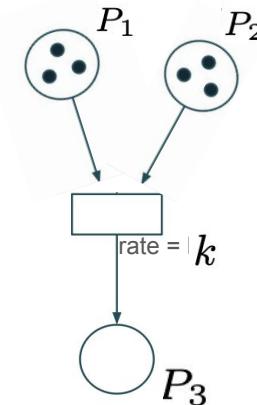
$$x = [x_{P_1}, x_{P_2}] \longrightarrow \text{The number of tokens in each place}$$

Transition velocities

The transition firings velocity is derived from the **propensity function!**

Mass Action (The Default)

- **Mass Action kinetics** assumes that transition velocities are **proportional** to the number of **available tokens** from the **input places**.
- This is a **simplifying assumption**, often valid when:
 - All molecules are well-mixed.
 - Transitions occur freely without spatial/structural constraints.



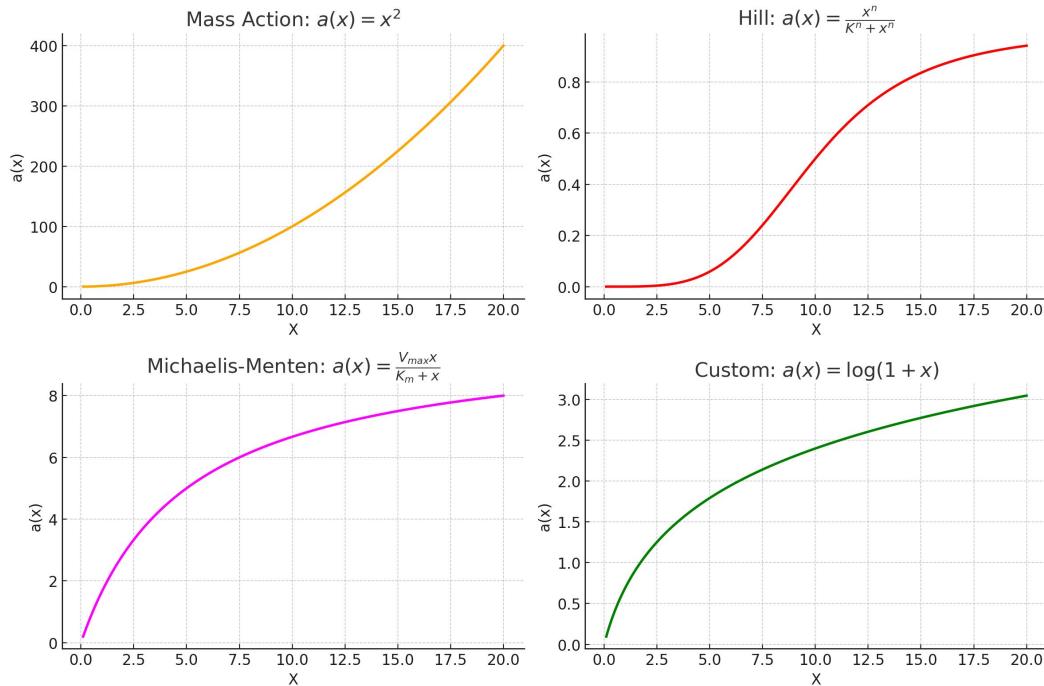
$$x = [x_{P_1}, x_{P_2}, x_{P_3}] \longrightarrow \text{The number of tokens in each place}$$

Transition velocities

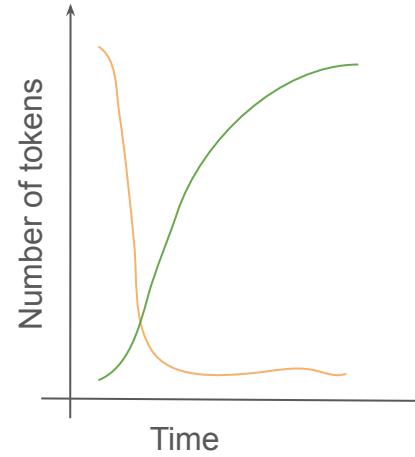
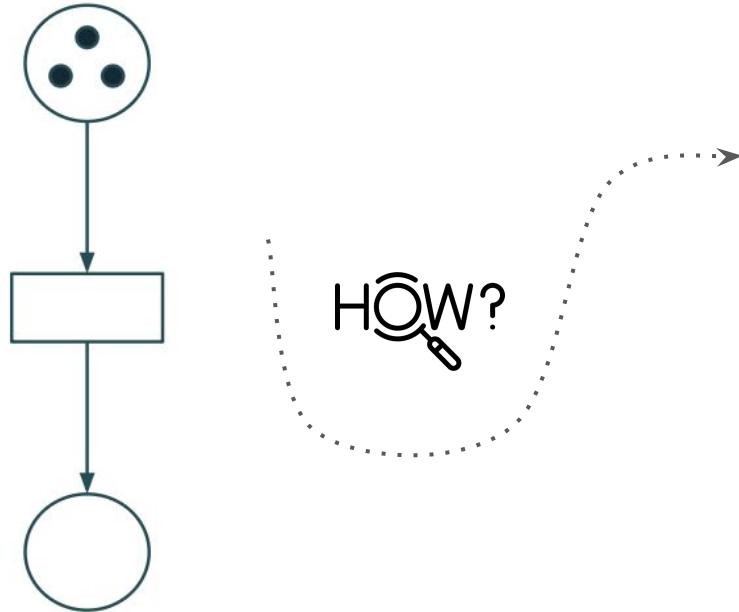
The transition firings velocity is derived from the **propensity function!**

Many biological systems **do not follow mass action**:

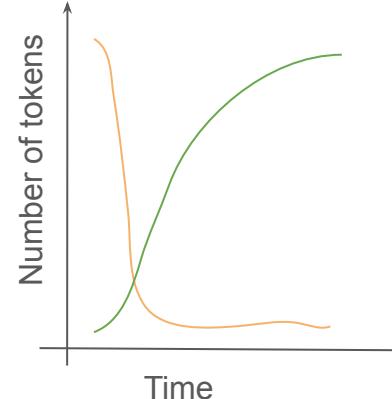
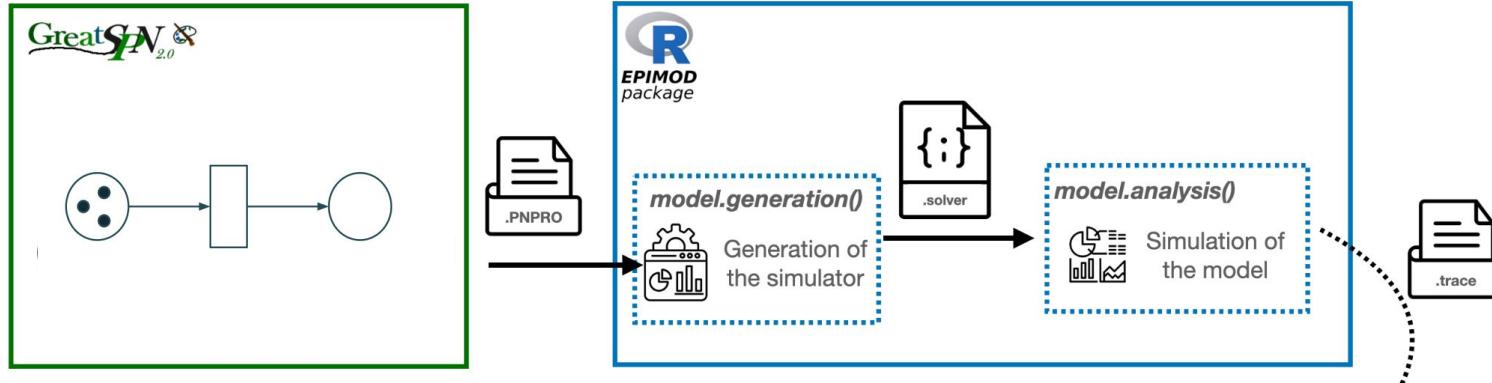
- **Enzyme reactions** reach saturation → **Michaelis-Menten**
- **Gene regulation** often shows switch-like behavior → **Hill kinetics**
- **Feedback or inhibition** may lead to nonlinear or **custom rates**



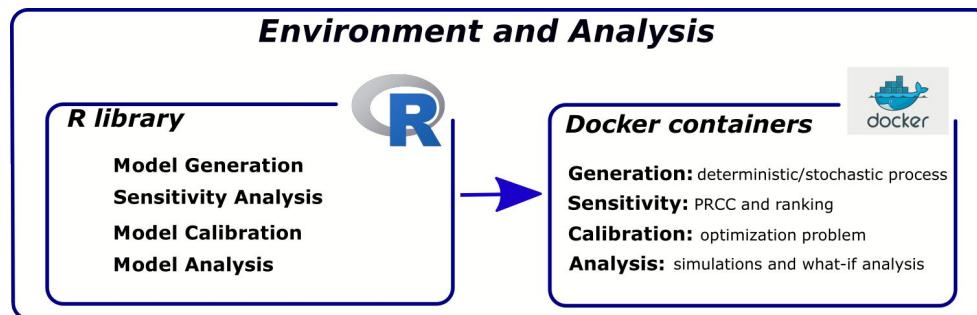
From graphics to dynamics



From graphics to dynamics

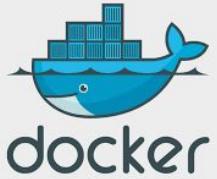


- General modeling framework for the analysis of epidemiological/biological systems;
- Tool easily accessible by any researcher even without advanced mathematical and computational skills.



Main features:

- **Graphical formalism:** you don't have to write equations, you draw them!
- **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



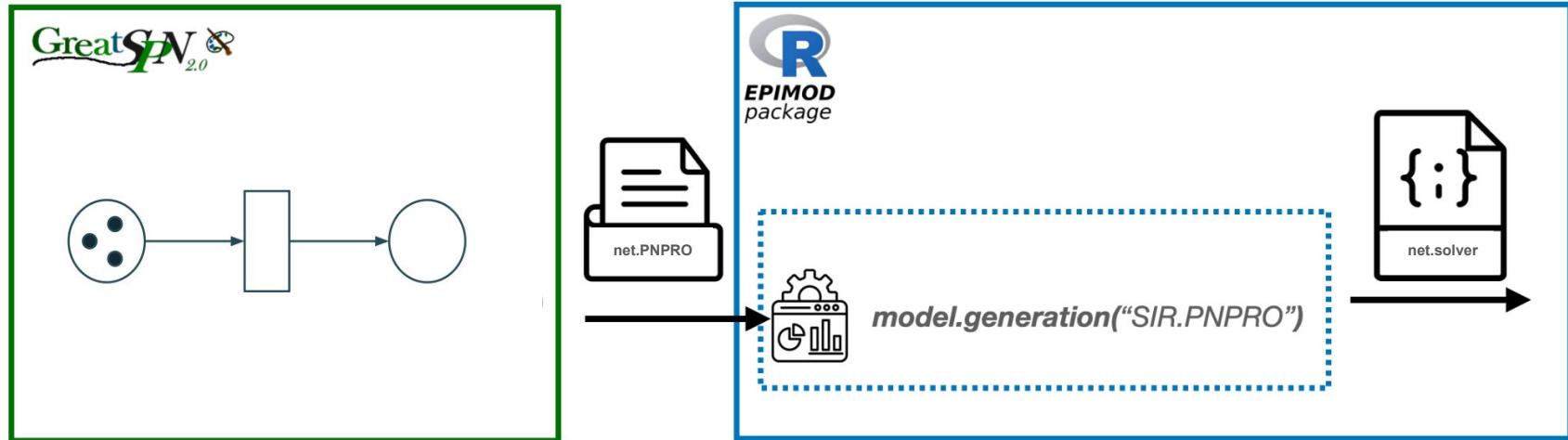
- Reproducibility is guaranteed by providing **Docker containers** packing all the required software and libraries frozen at a given version.
- Docker images are equivalent to Virtual Machines but **Docker container provide an operating-system-level virtualization** by abstracting the user space rather 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

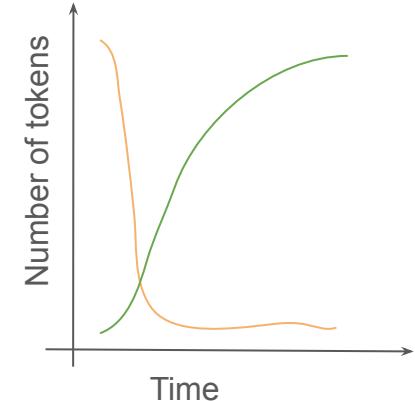
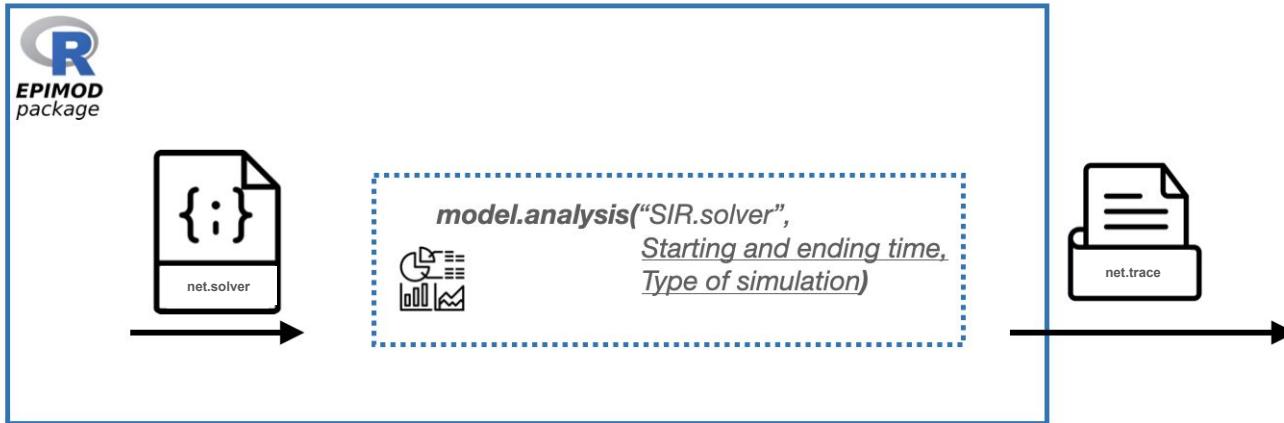
Model Generation

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



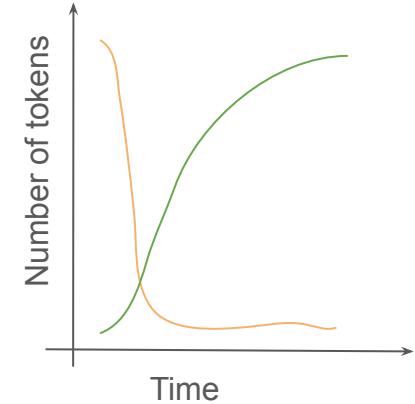
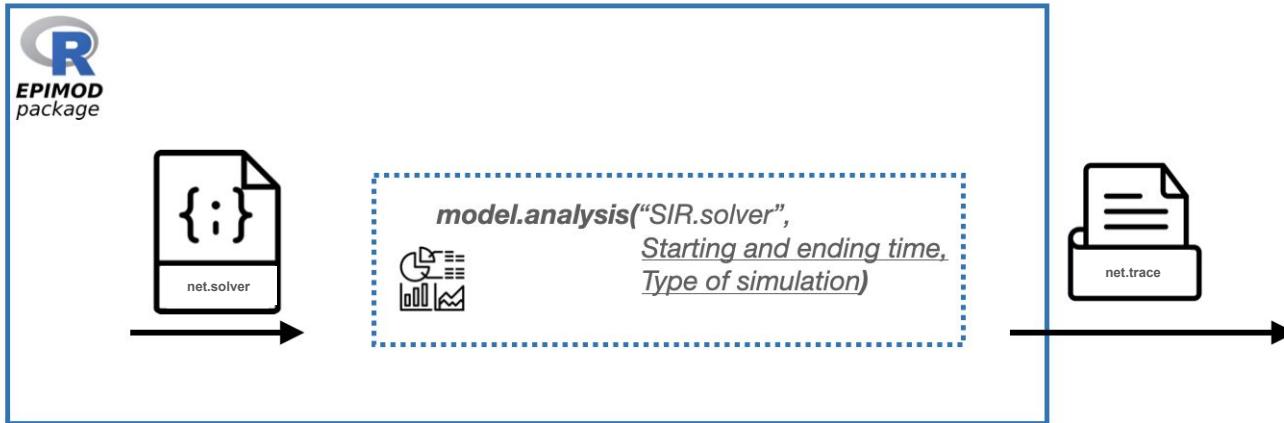
Model Analysis

`model_analysis()`: play with your brand new model



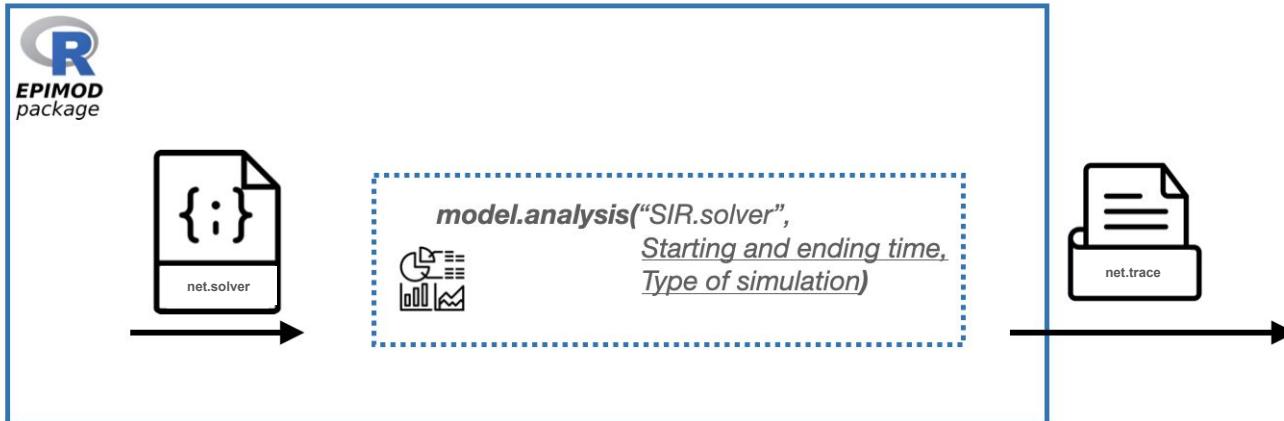
Model Analysis

`model_analysis()`: play with your brand new model

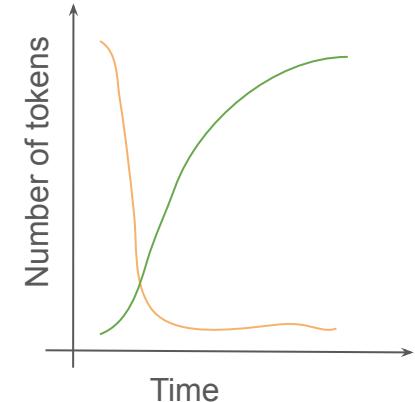


Model Analysis

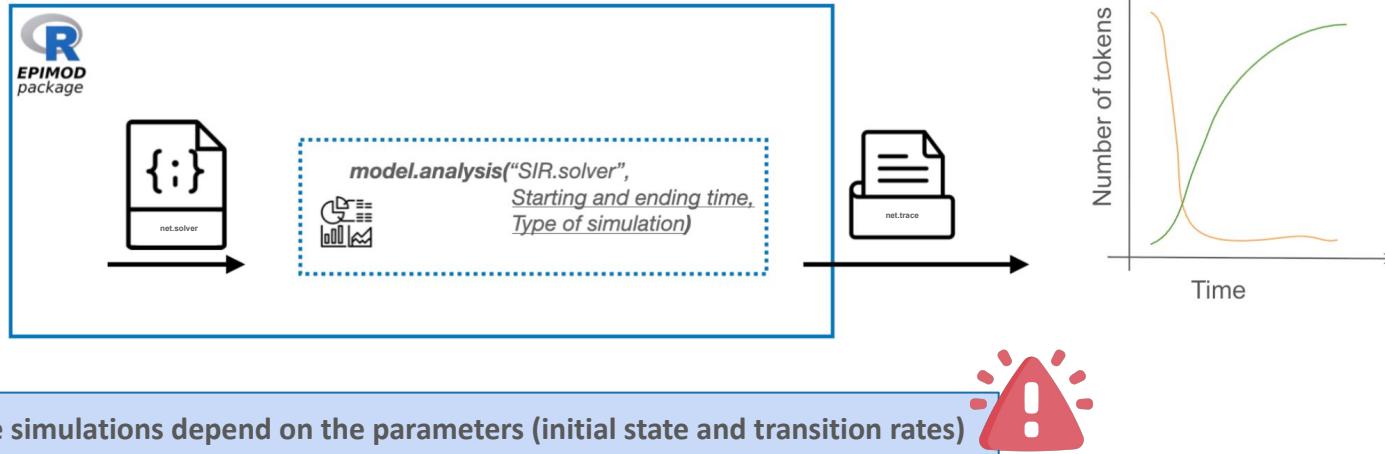
`model_analysis()`: play with your brand new model



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



Changing Model Parameters



— 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()*



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	input parameters needed by the R function, which can be an R built-in function or a <u>user-defined function</u>
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	



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

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

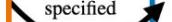


functions_fname.R

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

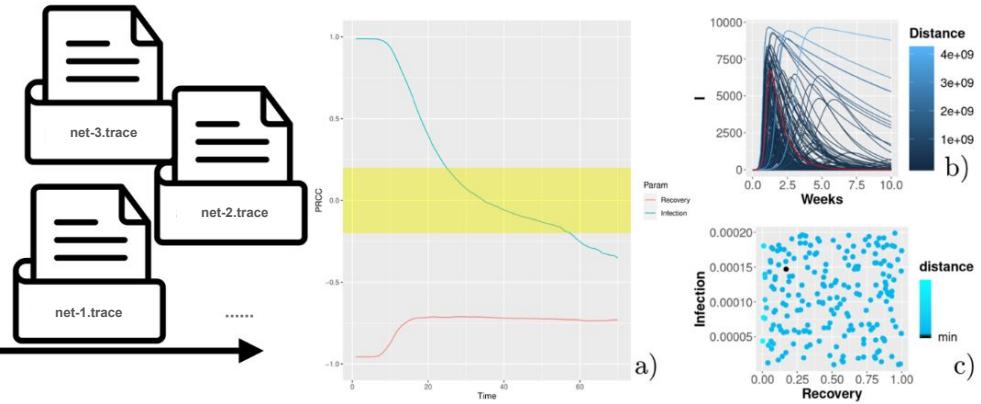
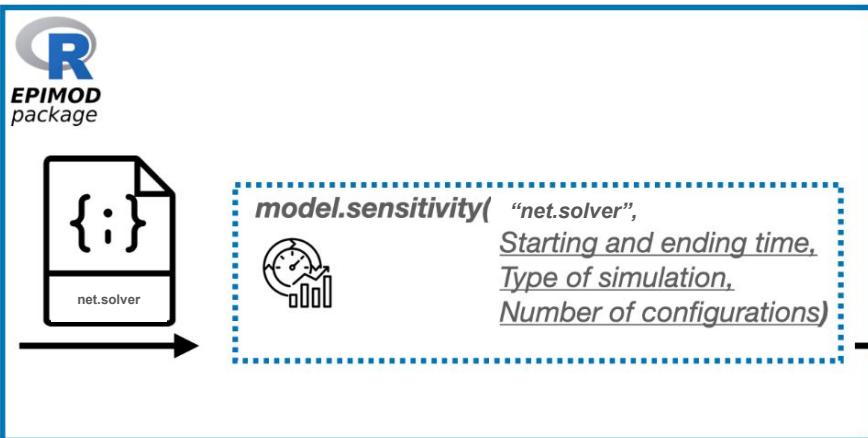
Example	Purpose	Purpose	Notes
init_generation	Generates the initial marking values	parameters_fname (with tag i)	Initial conditions
target	Extracts the target values from model output (for PRCC calculation)	target_value	Selects one or more places to track over time.
mse	Computes the error/ distance between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

user-defined function included in the R script specified



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 Sensitivity

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

What is PRCC?

Partial Rank Correlation Coefficient (PRCC) measures how strongly a model **output** depends on a parameter, while **controlling** for the influence of other parameters.

How It Works

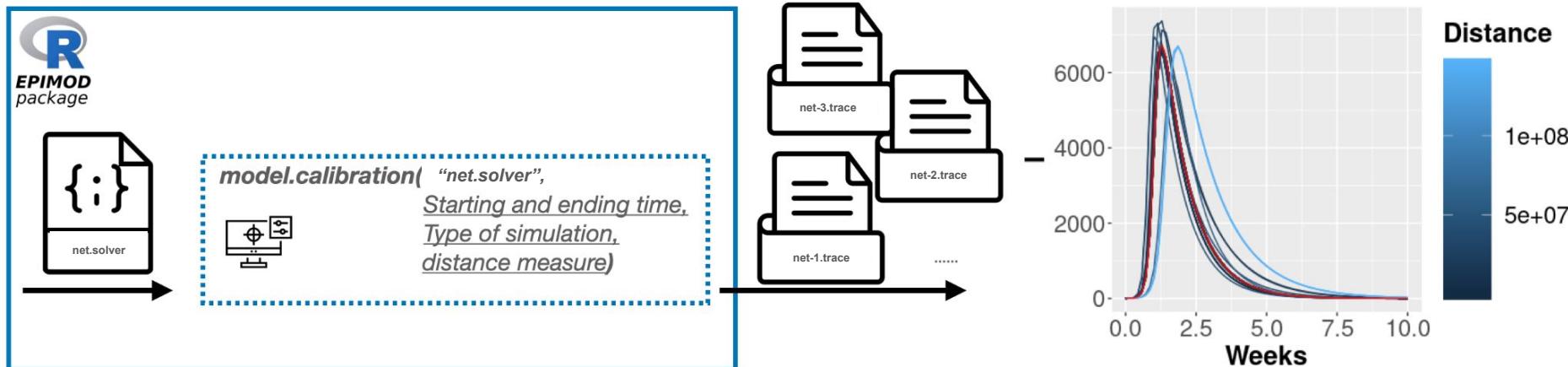
1. Sample parameters using **Latin Hypercube Sampling** (LHS)
2. Simulate the model for each parameter set
3. Record output of interest
4. Compute PRCC between each parameter and output

PRCC Value	Interpretation
~1	Strong positive effect
~0	No effect
~-1	Strong negative effect

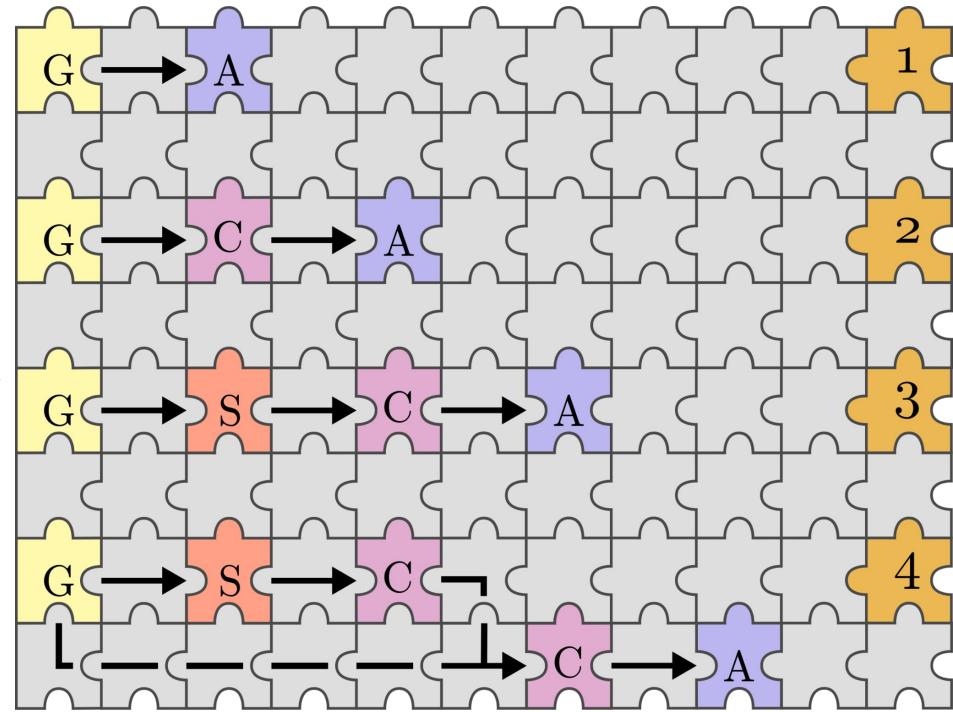
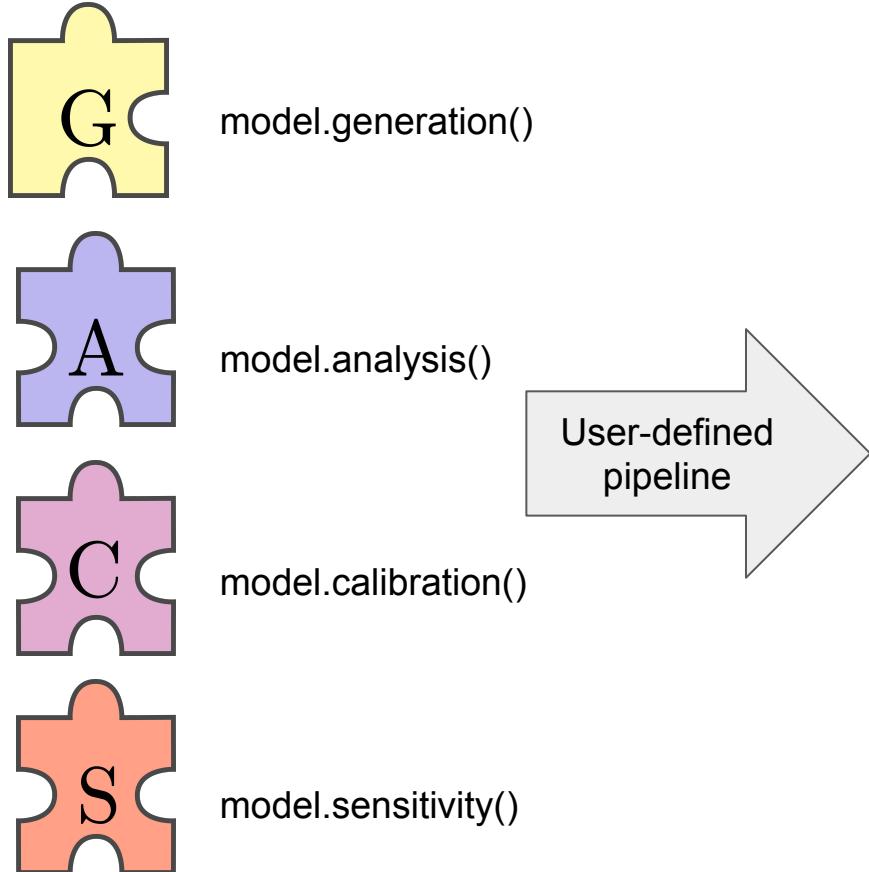
Model Calibration

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

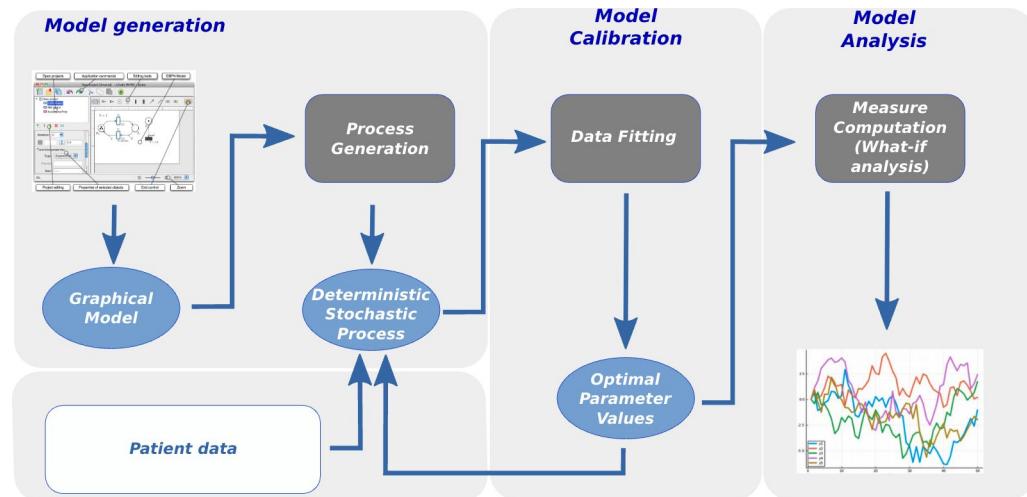
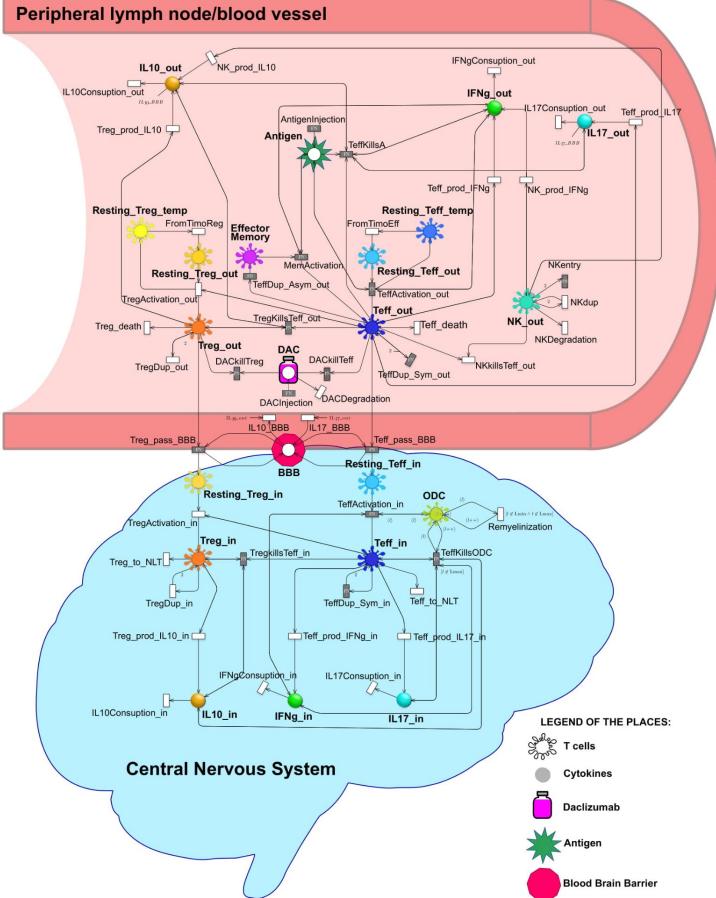
Minimise the distance between the reference data and the model's output according to a given (user provided) distance metric.



Epimod is modular



Epimod is modular - Multiple Sclerosis Model

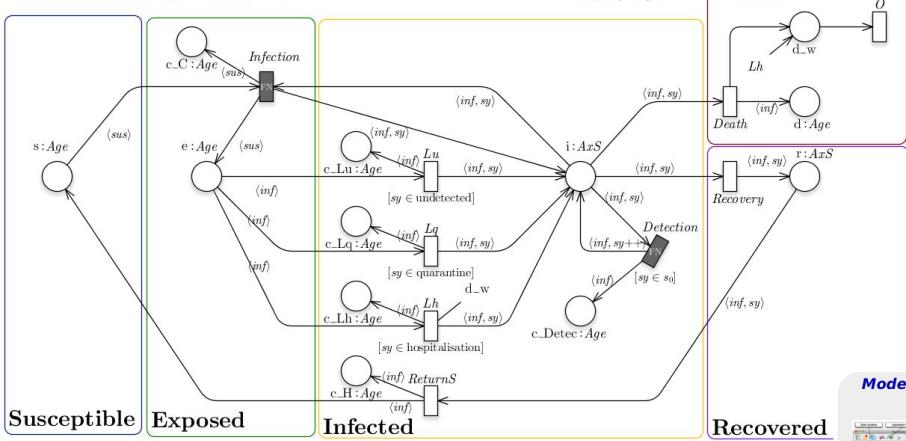


Epimod is modular - COVID-19 Model

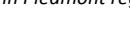
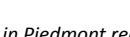
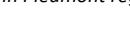
domain $AxS = Age \times Symptom$

class $Symptom = \text{circular } \{s_0\} \text{ is undetected} + \{s_1\} \text{ is quarantine} + \{s_2\} \text{ is hospitalisation}$

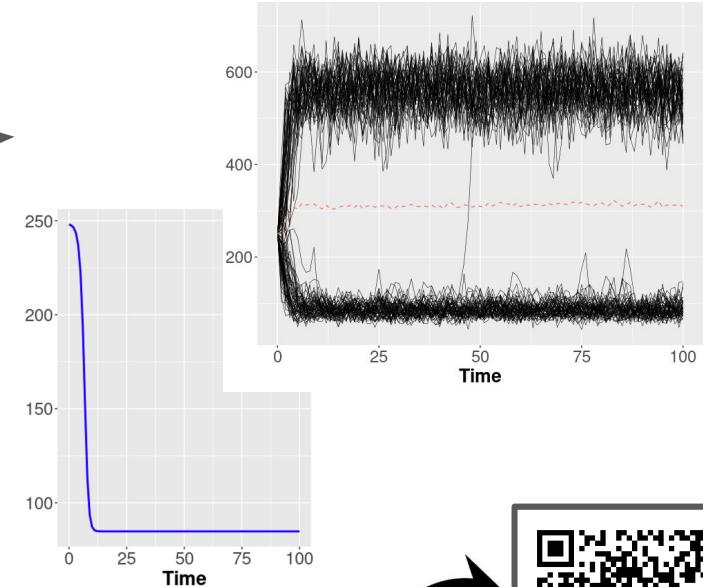
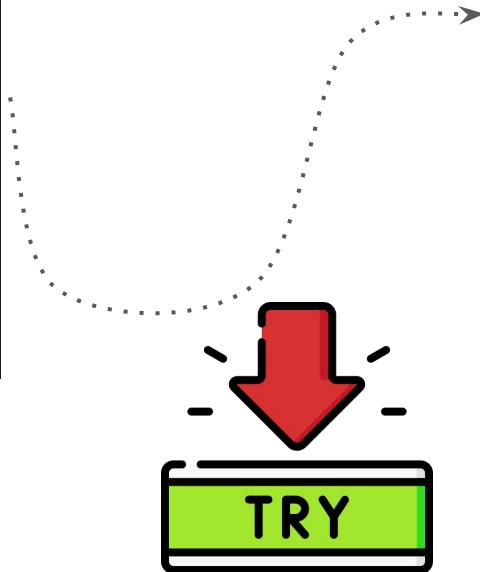
class $Age = \{a_0\} \text{ is } A_0 + \{a_1\} \text{ is } A_1 + \{a_2\} \text{ is } A_2$



var $sus : Age$
var $sy : Symptom$
var $inf : Age$



Case study: Shlogel model



<https://github.com/qBioTurin/From-Data-to-Models.git>