Stochastic Process

Gillespie

Sergio Peignier sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

Table of contents

Gillespie

Stochastic Simulation

Simulation of a **system** of **variables** changing **randomly** with given **probabilities**.

- **Stochastic model**
- **Random values** $\sim$ Pseudo-random generator

} **Outcome estimation**

Repeat procedure $\rightarrow$ collect outcomes $\rightarrow$ Estimate Distrib.

Gillespie Algorithm Principles

Method generating a **statistically correct possible solution** of a **stochastic equation**

Simulate complex systems

e.g., reactions within cells (computational systems biology)

Difference w.r.t. deterministic models

- **Continuous** and **deterministic** models (ODE): bulk reactions (millions of molecules)
- **Gillespie algorithm** simulates reaction **explicitly**.

Physical Hypothesis

- Based on **molecules collision** in a reaction vessel
- **Well mixed** reaction environment.
- Reactions involve **at most two molecules**.
- Reaction involving **more than 2 molecules**:
  modeled as a **sequence of binary reactions**.

Gillespie Variables

- State vector $\mathbf{x}(t) = \{x_1(t), \ldots x_I(t)\}$:
- Reactions: $\mathbf{r} = \{r_1, \ldots, r_J\}$ with:
- Rate constants: $\mathbf{k} = \{k_1, \ldots, k_J\}$
- Reaction rates: $\lambda(\mathbf{x}(t), \mathbf{k}) = \{\lambda_{r_1}, \ldots, \lambda_{r_J}\}$
- Total rate R_{tot}
- State change matrix:
  V s.t. v_{i,r_j}: change on state i due to reaction r_j

Pseudocode

function GILLESPIE($\mathbf{x}_0$, $\mathbf{k}$, T, V)

  $\mathbf{x} \leftarrow \mathbf{x_0}$

  repeat

    //Compute the reaction rates

    $\{\lambda_{r_1}, \ldots, \lambda_{r_J}\} \leftarrow \lambda(\mathbf{x}(t), \mathbf{k})$

    $R_{tot} \leftarrow \sum_{r_i} \lambda_{r_i}$

    //Draw next reaction time

    $\delta_t \sim Exponential(\lambda = R_{tot})$

    Draw a reaction r_j with Prob. prop. to λ_j.

    $\mathbf{x} \leftarrow \mathbf{x} + V_{.,r_j}$

    $t \leftarrow t + \delta t$

  until $t \geq T$

end function

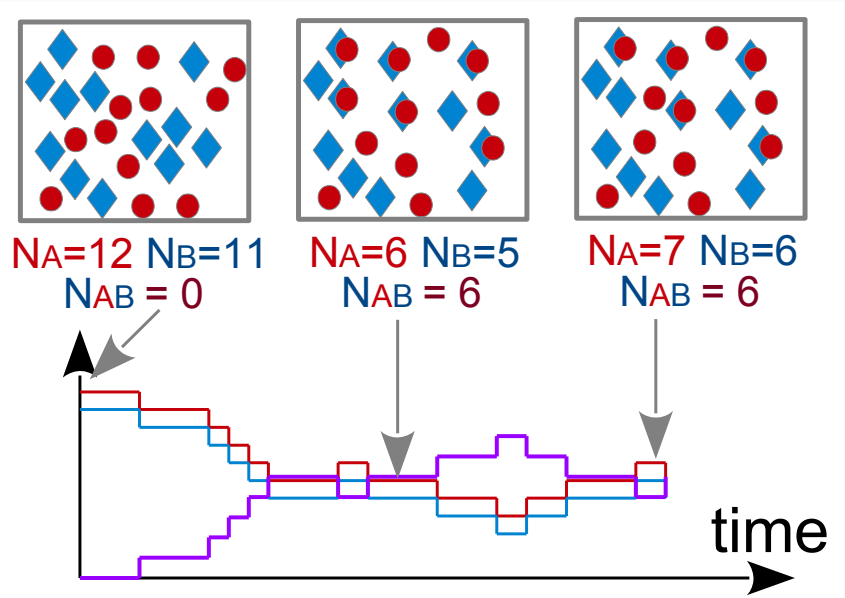
Example | Problem Description

- **Reactions:**
  - Reaction 1: $A + B \rightarrow AB$
  - Reaction 2: $AB \rightarrow A + B$
- **Reaction rate constant/coefficient:**
  - Reaction 1: k_1
  - Reaction 2: k_2
- **Number of molecules**:
  - Molecules of type A: n_A
  - Molecules of type B: n_B
  - Dimers AB: n_{AB}
- **Reaction rates:**
  - Reaction 1: $k_1 \times n_A \times n_B$
  - Reaction 2: $k_2 \times n_{AB}$
  - Total reaction rate: $R_{tot} = k_1 \times n_A \times n_B + k_2 \times n_{AB}$

Gillespie Execution

Repeat:

- Compute the **next reaction time**
  $\delta_t \sim Exponential(\lambda = R_{tot})$
- **Choose** and **apply** one **reaction** with **Probabilities**:
  - $P(A + B \rightarrow AB) = k_D n_A n_B / R_{TOT}$
  - $P(AB \rightarrow A + B) = 1 - P(A + B \rightarrow AB)$



Tau-Leaping

Tau-Leaping Algorithm Principles

- **Based** on the **Gillespie algorithm**
- **Analogous** to the **Euler method** for ODEs
- **Perform all reactions** for an **interval** of length *tau* before updating **states**.

$$x(t + \tau) = x(t) + Poisson(\tau x'(t))$$

- **Faster than Gillespie** $\rightarrow$ Simulate **larger systems**.

Tau-Leaping Variables

- State vector $\mathbf{x}(t) = \{x_1(t), \ldots x_I(t)\}$:
- Reactions: $\mathbf{r} = \{r_1, \ldots, r_J\}$ with:
- Rate constants: $\mathbf{k} = \{k_1, \ldots, k_J\}$
- Reaction rates: $\lambda(\mathbf{x}(t), \mathbf{k}) = \{\lambda_{r_1}, \ldots, \lambda_{r_J}\}$
- Total rate R_{tot}
- State change matrix:
  V s.t. v_{i,r_j}: change on state i due to reaction r_j
- State change vectors $\mathbf{v}(t) = \{v_1(t), \ldots v_I(t)\}$

Tau-Leaping Pseudocode

function TAU-LEAPING($\mathbf{x}_0$, $\mathbf{r}$, V, T, τ, $\mathbf{v}$)
  $\mathbf{x} \leftarrow \mathbf{x}_0$
  repeat
    Compute reaction rates $\{\lambda_{r_1}, \ldots, \lambda_{r_J}\} \leftarrow \lambda(\mathbf{x}(t))$
    Compute τ // if needed
    for all $v_i \in \mathbf{v}$ **do**
      $v_i \leftarrow 0$
    end for
    for all $r_j \in \mathbf{r}$ **do**
      // Nb. times r_j occurs during $[t, t + \tau]$
      Draw $K_{r_i} \sim Poisson(\lambda_i \cdot \tau)$
      $\mathbf{v} \leftarrow \mathbf{v} + K_{r_i} \cdot V_{.,j}$
    end for
    $t \leftarrow t + \tau$
    $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$
    Sanity check $\mathbf{x} > 0$
  until $t \geq T$
end function

Differences w.r.t. Gillespie

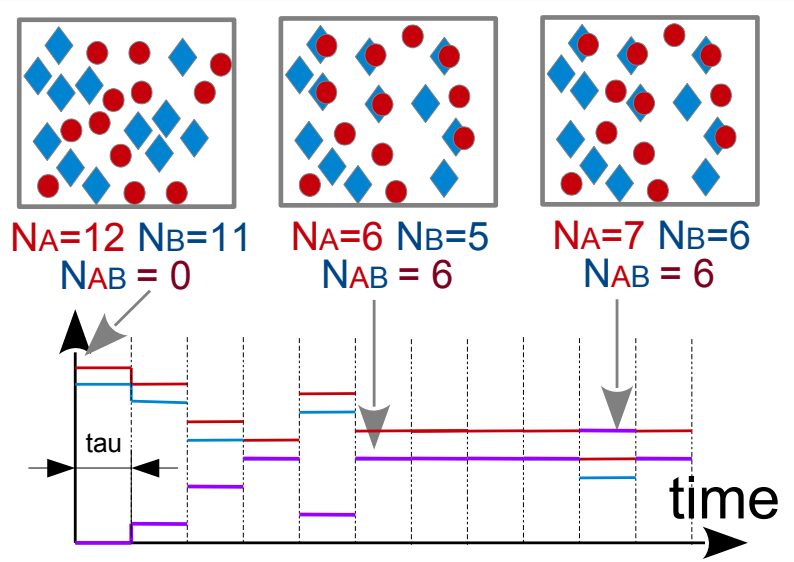


Figure 1: Tau-leaping

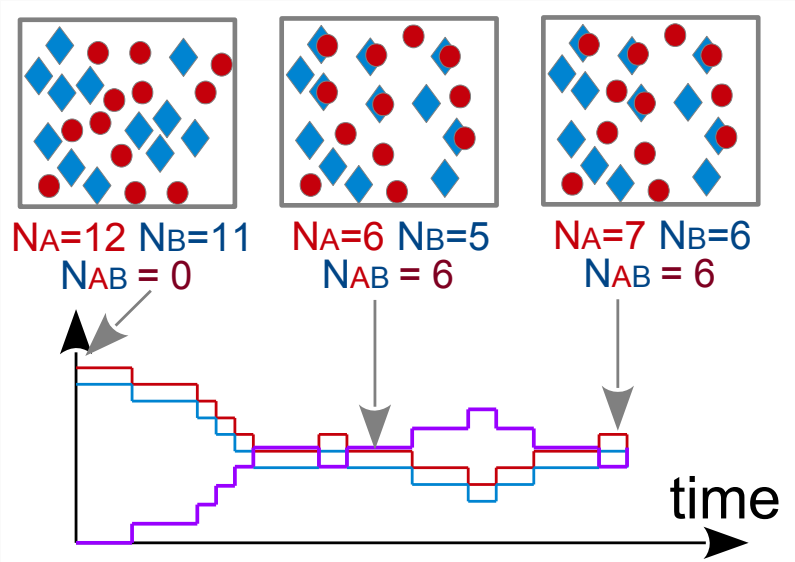


Figure 2: Gillespie