

Weekly Report

Habib Latifzadeh

Dr. Chan Group/ Duke University, Department of Biostatistics and Bioinformatics

September 18, 2024

- 1 Vivarium Overview
- 2 Vivarium Processes and Structures
- 3 Vivarium in Applications
- 4 Predator-Prey Model (Lotka-Volterra)
- 5 Vivarium Simulation Workflow & Results
- 6 Model Outputs

Vivarium Overview

- **Modular Design:**

- Flexible modeling by composing individual biological processes into composite models.
- Biological processes modeled as independent modules connected through ports and stores.

- **Multiscale Integration:**

- An integrative platform designed to support the simulation of biological processes across multiple scales (deterministic, stochastic, and agent-based models).

- **Scalable Computation:**

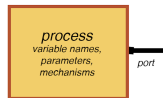
- Parallel execution on multiple CPUs.

- **Libraries and Paradigms:**

- vivarium-cobra: For flux balance analysis of metabolic networks using COBRA.
- vivarium-bioscrape: For modeling chemical reaction networks (CRNs) like transcription, translation, and regulation using the Bioscrape.
- vivarium-multibody: For spatial multicellular physics and diffusion, which models interactions between cells in a shared environment.

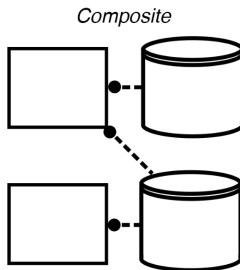
Vivarium's Interface Architecture

- **Process (rectangular flowchart):** A modular model contains parameters, an update function, and ports.
 - The building blocks of simulations and represent biological mechanisms (e.g., transcription and translation in gene expression).
 - Each process interacts with "ports" (data inputs/outputs) and "stores" (state variables).
- **Store (flowchart symbol):** Holds state variables and schemas, determining how updates to the system occur.
- Stores act as repositories where data (variables) used and generated by the processes are kept. Each variable in a store is defined with specific attributes that dictate how it behaves within the simulation.
- Schema: Each variable has a schema that defines its data type (integer, float, string) and the methods that can be applied to it. This schema ensures that different processes handle the data consistently and correctly.



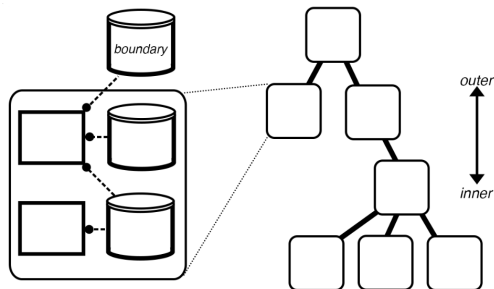
Vivarium's Interface Architecture

- **Composites:** Bundles of Processes and Stores wired together by a bipartite network/ Topology, with Processes connecting to Stores through their ports.
 - For instance, two processes, transcription and translation, are combined into a composite to simulate gene expression



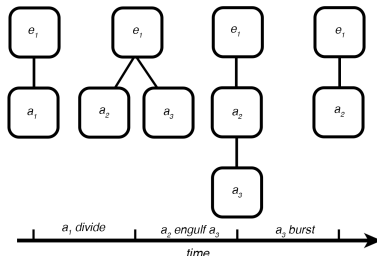
Vivarium's Interface Architecture

- **Compartmentments:** Stores with inner sub-Stores and Processes. Processes connect across compartments via boundary stores.
- **Hierarchy:** Compartments are embedded in a hierarchy as a place network with discrete layers, outer compartments above and inner compartments below.
- **Ports:** These serve as interfaces for the processes. Ports allow processes to connect to each other and to shared resources, facilitating the flow of data between them.



Hierarchy Updates in Modeling Systems

- **Divide Update:** Adds new compartment by splitting existing one.
 - *Purpose:* Simulates cell division—splits one cell into two distinct compartments.
- **Engulf Update:** One compartment absorbs another, which is then subsumed.
 - *Purpose:* Models processes like phagocytosis or integration of one system component into another.
- **Burst Update:** Deletes an engulfed compartment post-engulfment.
 - *Purpose:* Represents the breakdown or digestion of the engulfed entity within the system.



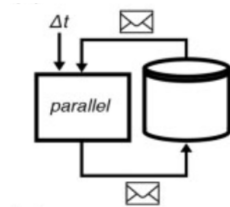
Vivarium Engine Operation

System Construction:

- The Vivarium engine constructs stores based on the declared schemas for each port of the processes.
- It assembles processes and stores into a hierarchy, executing their interactions over time.
- Aimed to support large models with thousands of mathematical equations.

Parallel Processing:

- Vivarium can distribute processes onto different OS processes (distinct from Vivarium processes) to handle complex and large-scale simulations efficiently.



Vivarium Applications

- The difference equation $\Delta x = f(r, x) \cdot \Delta t$ describes the dynamics of the system.
- A **Vivarium store** is a computational object that holds the system's state variables x .
- A **Vivarium process** is a computational object containing the update function f , which governs how variables evolve over time, describes the inter-dependencies between variables, and maps them from one time t to the next time $t + \Delta t$.
- **Processes** are configured by parameters r , which define how the update functions map input values to output values, giving them a distinct shape.

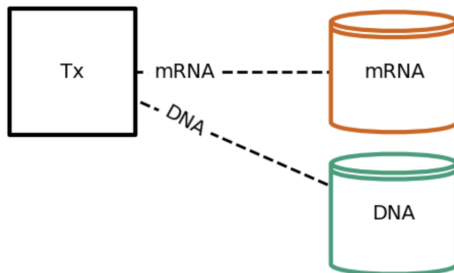
Vivarium Simulation

Simulated Biological Processes:

- **Transcription (Tx):** DNA to mRNA.
- **Translation (Tl):** mRNA to protein.
- **Stochastic Tx:** Randomized transcription process.
- **TxTl Composite:** Combined Tx and Tl, includes agent-based growth and division.

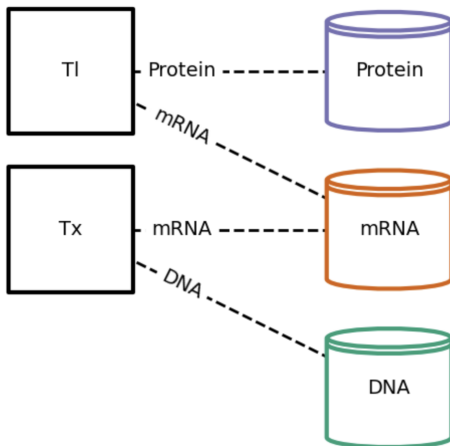
Vivarium Topology Outputs:

- Topology of Transcription (interactions between DNA, mRNA).



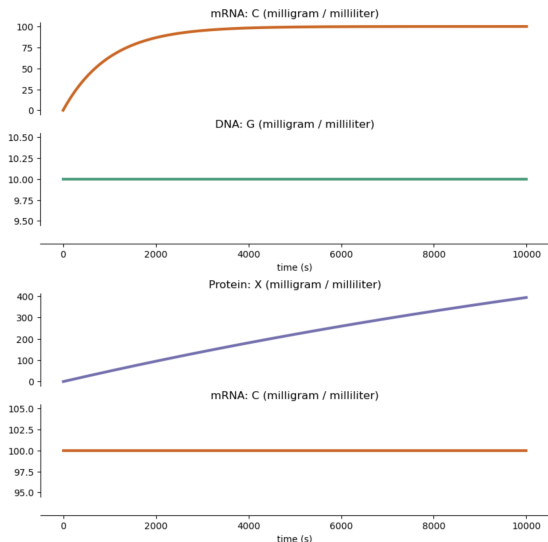
Vivarium Topology Outputs

- Topology diagrams of Translation (interactions between DNA, mRNA, protein).



Vivarium Topology Outputs:

- Timeseries plots of mRNA and protein concentrations over time.



Predator-Prey Model (Lotka-Volterra)

- **Overview:** Model simulates dynamics between two species, one as predator and the other as prey.
- **Prey Population $x(t)$ & Prey Equation:** $\frac{dx}{dt} = \alpha x - \beta xy$:
 - αx : Natural growth of prey.
 - $-\beta xy$: Reduction due to predation.
- **Predator Population $y(t)$ & Predator Equation:** $\frac{dy}{dt} = \delta xy - \gamma y$:
 - δxy : Growth by consuming prey.
 - $-\gamma y$: Natural death in absence of prey.
- **parameters:**
 - α : Growth rate of prey.
 - β : Rate of predation.
 - γ : Natural death rate of predators.
 - δ : Predator growth rate from consuming prey.

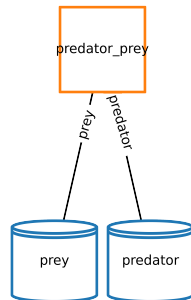
Vivarium Simulation Workflow and Results

- **Vivarium Implementation & Topology Diagram:**

- Processes: Prey growth, predation, predator growth, predator death.
- Stores: Population sizes of predators and prey.
- Composites: Combined processes simulate predator-prey interactions.

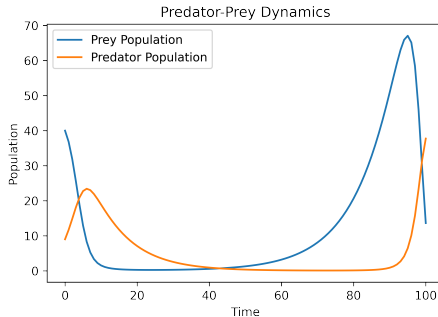
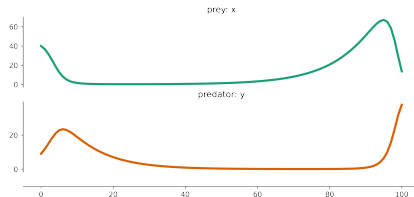
- **Workflow:**

- Initialization of prey and predator populations.
- Model execution using Lotka-Volterra equations.



Model Outputs

- **Vivarium result:** Time series of prey and predator populations.



Model Outputs

- **Phase Curve (Prey vs Predator):** The plot, the prey population on the x-axis and the predator population on the y-axis, shows how the populations change with respect to each other, rather than over time.

