

# Weekly Report

Habib Latifzadeh

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

September 18, 2024

- 1 Predator-Prey Model (Lotka-Volterra)
- 2 Vivarium Simulation Workflow & Results
- 3 Model Outputs

# 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:**  $\frac{dx}{dt} = \alpha x - \beta xy$ :
  - $\alpha x$ : Natural growth of prey.
  - $-\beta xy$ : Reduction due to predation.
- **Predator Population  $y(t)$  & Predator Equation:**  $\frac{dy}{dt} = \delta xy - \gamma y$ :
  - $\delta xy$ : Growth by consuming prey.
  - $-\gamma y$ : Natural death in absence of prey.
- **parameters:**
  - $\alpha$ : Growth rate of prey.
  - $\beta$ : Rate of predation.
  - $\gamma$ : Natural death rate of predators.
  - $\delta$ : Predator growth rate from consuming prey.

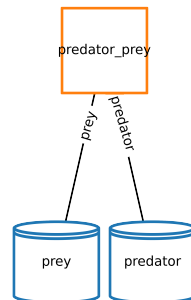
# Vivarium Simulation Workflow and Results

- **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.

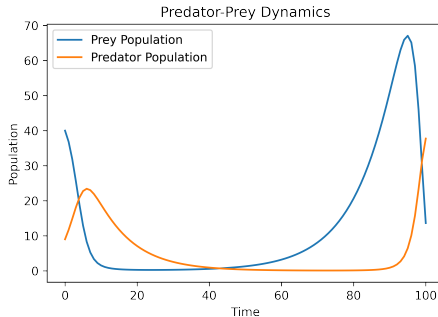
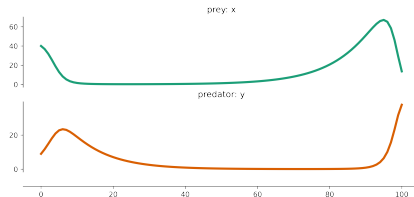
- **Workflow:**

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



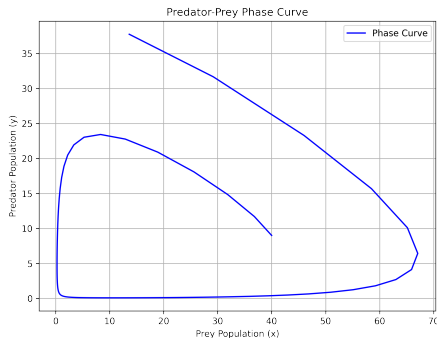
# Model Outputs

- **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.



# Weekly Report

Habib Latifzadeh

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

September 12, 2024

- ④ Research Progress
- ⑤ Vivarium Overview
- ⑥ Vivarium Processes and Structures
- ⑦ Vivarium in Applications
- ⑧ Challenges and Future Directions



- **Vivarium Simulations:** Executed several Vivarium-based simulations modeling biological processes such as Transcription and Translation (Tx and TI) in both deterministic and stochastic frameworks with visualizations of mRNA and protein concentration changes over time. Additionally, a Composite Simulation (TxTI) with hierarchical embedding to simulate agent growth and division.
- **Visual Outputs:** Generated topology diagrams and timeseries data visualizing mRNA and protein concentration ( included in the Jupyter notebook outputs)
- **Next Steps:** Further discussion on integrating these Vivarium simulations into CMV ODE model, and exploring any specific configurations for future runs.

# 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:**

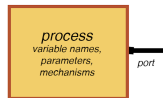
- Parallel execution on multiple CPUs.

- **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.

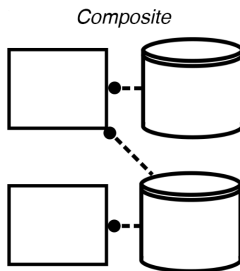
# Vivarium's Interface Architecture

- **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.



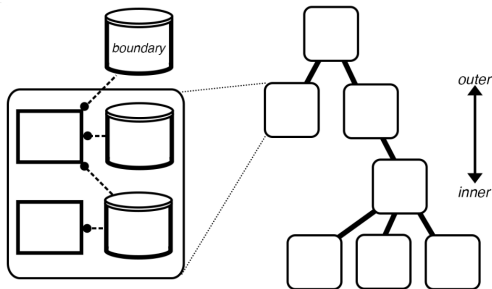
# Vivarium's Interface Architecture

- **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



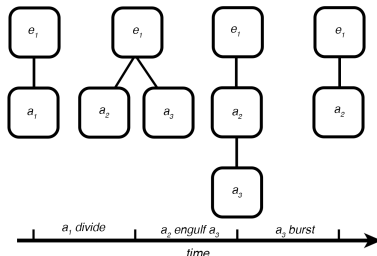
# Vivarium's Interface Architecture

- **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.



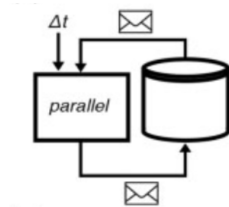
# Vivarium Engine Operation

## 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.



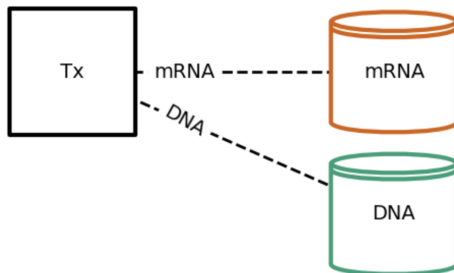
# Vivarium Simulation

## 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.

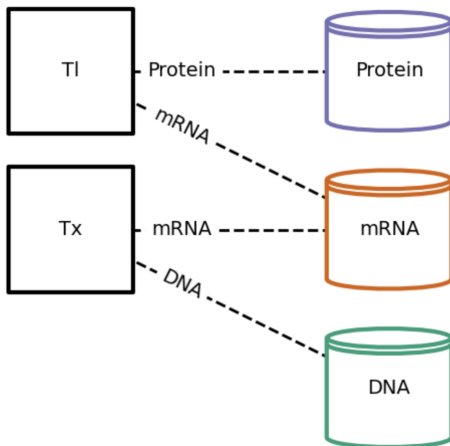
## Vivarium Topology Outputs:

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



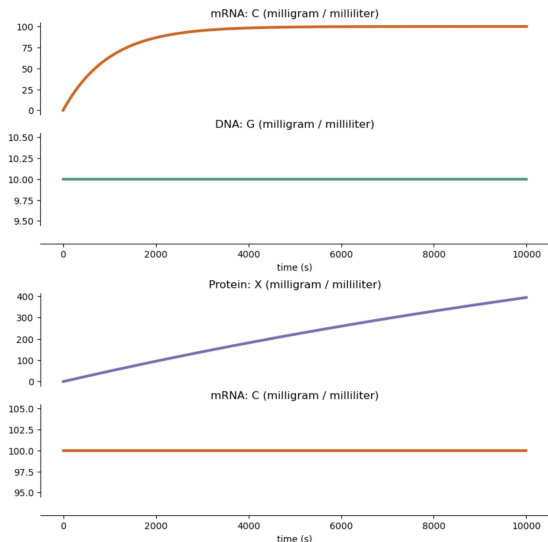
# Vivarium Topology Outputs

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



# Vivarium Topology Outputs:

- Timeseries plots of mRNA and protein concentrations over time.



# Challenges and Future Directions

- **Implementing CMV ODE System in Vivarium:** Integrating the CMV ODE system into the Vivarium framework to model viral dynamics and immune interactions.
- **Agent-Based and ODE Hybrid Modeling:** Combining agent-based models and ODE systems to simulate both individual-level cell interactions and population-level dynamics in the CMV infection model.
- **Stochastic Processes in Vivarium:** Incorporating stochastic elements into the Vivarium ODE system to reflect random events in viral transmission and immune responses.
- **Real-World Data Validation in Vivarium:** Using clinical and experimental data to validate and fine-tune the Vivarium-based CMV ODE model for accurate real-world predictions.

# Weekly Report

Habib Latifzadeh

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

September 5, 2024

- 9 Research Progress
- 10 Agent-Based Modelling
- 11 Vivarium Overview
- 12 Vivarium Processes and Example Applications
- 13 Relevance to CMV Research
- 14 Challenges and Future Directions

- **Vaccine-Induced Antibody Responses:**

- Reviewed gB/MF59 CMV vaccine study, showing antibodies that prevent cell-to-cell spread.
- Plan to integrate antibody data into CMV transmission models to improve predictions.
- Also reading Dr. Permar's study on mRNA-1647 vaccine, which shows stronger neutralization and antibody responses.
- Both studies could help refine our CMV models and improve prevention strategies.

- **Agent-Based Modeling (ABM):**

- Exploring ABM to simulate immune cell responses.
- Discuss ABM strategies with a researcher to enhance model accuracy.

- **Immune Checkpoints and Vaccine Development:**

- Investigating how targeting immune checkpoints could enhance vaccine responses.

- **A bottom-up modelling approach:**

- Systems' behavior emerges from actions and interactions of autonomous agents (e.g., patients and providers) with each other and the environment over time.

- **An experimental approach to:**

- Testing explanatory hypotheses of real-world patterns.
- Translating individual-level assumptions to population models.
- Examining population-level effects as a decision support tool.
- Guiding data collection.



# Introduction to ABMs

- **Definition:** ABMs analyze complex systems by simulating individual interactions (micro-level) to reveal system-level phenomena (macro-level).
- **Key Components:**
  - **Agents:** Autonomous entities interacting with the environment and other agents based on rules, adapt based on interactions.
  - **Environment:** The space where agents interact. It can be real-world locations (GIS integration) that determine how agents move and connect.
  - **Rules:** Govern agent behavior and interactions which Define agent interactions and environment dynamics.

# Why ABMs? & Challenges in ABM Development

## Why ABMs?

- **Advantages:**

- Effective for studying systems involving individual interactions.
- Emergent behavior through agent interactions provides insights into complex phenomena.

- **Complexity:**

- Difficulty in developing for non-computer scientists.
- Need for easy-to-use, efficient platforms for model development.

- **ABM Tools:**

- Provide frameworks to abstract development complexity, allowing users to focus on simulation outcomes.

# Key Desiderata for ABM Tools

- **Efficiency and Ease of Use:** Balancing performance and accessibility.
- **Visualization and Analysis:** Real-time monitoring and data visualization are critical.
- **Stochastic Processes:** Handling randomness in simulations through robust random number generation.

# Popular ABM Tools Overview

- **ActressMAS**: .NET-based, easy-to-use, not for high-performance needs.
- **AgentPy**: Python library, Jupyter integration, supports parallel simulations.
- **Agents.jl**: Julia-based, efficient and scalable, supports GIS data.
- **Care HPS**: C++ tool for high-performance parallel computing.

- **FLAME & FLAME GPU:** For parallel computing, supports XML-based modeling for high-performance simulations on supercomputers and GPUs.
- **GAMA:** Integrated with Eclipse IDE, supports massive simulations with GIS capabilities.
- **NetLogo:** Standard ABM platform, includes VPL, integrates with Python (PyNetLogo) and R (RNetLogo).

- ABMs provide a powerful method for analyzing complex systems using a bottom-up approach.
- Choosing the right tool depends on the user's technical skills, model complexity, and performance needs.
- **Future Directions:** Exploring distributed computing applications like federated learning and blockchain systems in ABM research.

# 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 integrates several libraries to extend its functionality, such as:

- 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 Processes and Example Applications

- **Processes, Stores and Composites:**

- Processes are the building blocks of simulations and represent biological mechanisms (transcription and translation in the gene expression example).
- Each process interacts with "ports" (data inputs/outputs) and "stores" (state variables).
- Composites are higher-level constructs that combine multiple processes. For instance, transcription and translation are combined into a composite to simulate gene expression, integrating the two processes into a unified model.

- **E. coli Colony Simulation:**

- Integrated flux balance analysis (FBA), gene expression (transcription and translation), and multicell physics.
- Demonstrates multiscale modeling potential for CMV simulations.



- **Viral Dynamics and Immune Response:**

- Use Vivarium to simulate viral replication, immune responses, and even vaccine effects at multiple biological scales.

- **Scalability:**

- Applicable for both within-host dynamics and population-level CMV studies using Agent-Based Modeling.
  - ① Within-host ABM simulates interactions between viral particles, immune cells, and other biological entities inside the host (the spread of CMV within tissues or between cells).
  - ② Population-level ABM models interactions between individuals in a population, including infection transmission, vaccination effects, and epidemiological outcomes.

# Challenges and Future Directions

- **CMV Model Refinement and Vivarium integration:**
  - Refining agent-based models (ABM) for simulating immune cell behaviors.
  - Implementation of CMV modeling progress and Vivarium integration.
- **Experimental Data Integration:**
  - Exploring dynamic integration of patient data to improve model accuracy.
- **Grant Writing:**
  - Planning to develop an idea for writing a T32 postdoc grant proposal to secure funding for further research and model development.