# Comparing Three Agent-Based Models Implementations of Vector-Borne Disease Transmission Dynamics

María Sofía Uribe, Mariajose Franco, Luisa F. Londoño(✉), Paula Escudero, Susana Álvarez, and Rafael Mateus

Universidad EAFIT, Medellín, Colombia
{msuribec,mfrancoo,llondo61,pescuder,salvarezz1,rmateusc}@eafit.edu.co

**Abstract.** Aedes aegypti, the vector responsible for transmitting diseases such as dengue, zika, and chikungunya, poses a significant public health threat in many regions. Understanding the dynamics of Aedes aegypti propagation is crucial for designing effective control and prevention strategies.

Agent-Based Models (ABMs) have emerged as valuable tools for studying complex systems like vector-borne disease dynamics. Hybrid Agent-Based Models (HABMs), a variation of these models that incorporates Ordinary Differential Equations to model mosquitoes and ABMs to model humans, have been proposed by several authors.

This study presents a comparative analysis of three HABMs to model Aedes aegypti propagation dynamics, with a focus on the impact of different modeling frameworks. The first model was built using Repast Simphony, a widely used ABM framework. It incorporates key factors such as mosquito life cycle, environmental conditions, and human-mosquito interactions. To enhance computational performance, the second model is migrated to a high-performance environment using Repast HPC. This migration leverages parallel computing capabilities to simulate larger populations. The third model is migrated to Mesa-Geo, a Python library specifically designed for geospatial agent-based modeling. This migration facilitates the integration of geospatial data into the model.

Preliminary results show that migrating the model to a high performance environment enables more comprehensive analyses and reduces simulation runtime. Moreover, migrating to Mesa-Geo provides enhanced geospatial capabilities, and allows us to analyze the results in a graphical interface, which facilitates communication with decision makers.

The main contributions of this research are: 1) insights into the trade-offs and benefits of using Repast Simphony, Repast HPC, and Mesa-Geo for modeling the transmission of viruses, and 2) a guide to researchers and stakeholders in selecting the most suitable modeling framework based on their specific requirements and available computational resources.

**Keywords:** Repast Simphony · Repast HPC · Mesa-Geo · HABMs · ABMs · Vector-Borne Diseases

M. Tabares et al. (Eds.): CCC 2023, CCIS 1924, pp. 330–349, 2024.
https://doi.org/10.1007/978-3-031-47372-2_27

## 1 Introduction

The Aedes aegypti mosquito is recognized as the main carrier of diseases like Dengue, Chikungunya, and Zika, making it a significant species. The prevalence of these diseases has been on the rise in recent years, causing millions of people around the globe to be affected each year. Transmission of these diseases happens when infected female mosquitoes bite humans, and their presence is closely associated with human habitation. While Dengue, Zika, and Chikungunya impact individuals across various social strata, they exhibit a greater propensity to affect the economically disadvantaged, particularly those residing in suburban regions. Consequently, factors such as climate, social conditions and human behavior play a crucial role in the spread of this mosquito species and the occurrence of the diseases it carries.

This research is specifically driven by the case of Bello, a city in Colombia, which possesses distinctive characteristics that make it an ideal subject for examining the impact of dengue in densely populated urban areas. Bello's proximity to other urban centers and transportation hubs heightens the risk of disease transmission, rendering it a potential epicenter for outbreaks that can propagate to other parts of the country.

Agent-based modeling and simulation (ABMs) is a powerful framework for understanding the dynamics of vector-borne disease transmission and devising effective control and prevention strategies. ABMs captures the complex interactions between individual agents, environmental characteristics, and intervention strategies, considering the heterogeneous characteristics and behaviors of the agents. It also allows the incorporation of spatial and temporal dimensions and their impact on disease transmission.

However, the complexity of ABMs models in this context requires significant computational resources, which can be addressed by employing High-Performance Computing (HPC). HPC provides the computational power to handle large numbers of agents and their interactions, ensuring timely simulations. Furthermore, HPC allows for the scalability of ABMs models, accommodating the increasing complexity and size of an urban area such as Bello. This combination of ABMs and HPC could enable researchers to gain valuable insights into disease dynamics, assess the effectiveness of various strategies, and make informed decisions for control and prevention efforts.

Limited research has been conducted on agent-based models that capture the intricate dynamics of vector-borne disease transmission, particularly in relation to the diverse geospatial characteristics of urban areas. This paper aims to address this gap by presenting an exploratory study that offers a comparative analysis of three distinct agent-based modeling implementations using different programming languages and simulation toolkits. The first implementation employs Python, the second utilizes Repast Simphony (Java), and the third utilizes Repast HPC. The analysis specifically considers factors such as computational capacity, computer memory usage, and implementation characteristics. By evaluating these aspects, this study aims to provide valuable insights into the strengths and limitations of each implementation, creating opportunities for further advancements in agent-based modeling approaches for studying vector-borne diseases dynamics in urban settings.

This paper commences with a comprehensive review of the existing literature regarding the utilization of ABMs in modeling vector-borne diseases within urban areas. Subsequently, it outlines the methodology employed to tackle the research problem. The paper proceeds to present the findings in the results section, followed by the conclusions drawn from the study and an exploration of potential future research directions.

## 2 Literature Review

The existing literature on the use of agent-based models (ABMs) to simulate the spread of vector-borne diseases has predominantly emphasized the relation between human behavior and disease propagation. However, there is a notable gap in considering other crucial factors, such as geospatial, social, and climate characteristics, within these models. For example, [20] employed agent-based models (ABMs) to investigate the influence of human behavior on the transmission of infectious diseases. The study revealed that even minor alterations in human behaviors can yield significant variations in outcomes, ultimately exerting a profound impact on virus propagation. Similarly, a model to address the propagation of Dengue and Chikungunya outbreaks in Colombia was developed in [6]; however, the model overlooked important factors such as spatial complexity and urban area density, limiting its ability to fully capture the dynamics of disease transmission. Furthermore, [9] implemented an ABMs that focused on simulating mosquito population dynamics at a neighborhood level. The study revealed a notable relation between urban topology, human population density, and adult mosquito flight. To account for the diverse geography of the area, the researchers incorporated distinct values for characteristics like temperature and light into different zones of the spatial representation. In [15], an agent-based simulation framework to analyze and predict mosquito population density and its impact on dengue spread, is proposed. The proposed framework provides visualization and forecasting capabilities to study the epidemiology of a certain region and aid public health departments in emergency preparedness. They identified several expected dengue cases and their direction of spread, which can help in detecting epidemic outbreaks. In a similar manner, [1] developed an agent-based model to investigate the African trypanosomiasis disease dynamics and as a tool for scenario testing at an appropriate spatial scale to allow the design of logistically feasible mitigation strategies. They implemented an agent-based model because this vector-borne disease is prevalent in sparsely populated rural environments, and the traditional compartmentalised models such as SIR don't always capture the spatial and demographic heterogeneity within an area, and the varying exposure to the disease that this can cause. They incorporated spatial data for the Luangwa Valley case study, along with demographic data for its inhabitants.

The literature has also shown that the utilization of hybrid agent-based models incorporating ordinary differential equations offers the advantage of capturing both individual-level interactions and population-level dynamics, enabling a more comprehensive understanding of the propagation of vector-borne diseases. For instance, in a study by [16], a novel hybrid model was proposed for simulating the transmission of mosquito-borne diseases. Instead of explicitly modeling the individual movement of each mosquito agent, the model employed Ordinary Differential Equations (ODEs) to capture the dynamics of mosquito populations. The authors argued that this approach offered a higher level of detail compared to using solely agent-based methods, while also improving computational efficiency.

Moreover, high-performance agent-based models (ABMs) provide a promising approach to study the propagation of vector-borne diseases. Recognizing the need for computational power in analyzing the complex dynamics of disease spread, researchers have started investigating the use of HPC agent-based models or migrated existing models to frameworks that support HPC and parallel simulation. An example of an HPC model in this context is the work by [18], which focuses on assisting decision-making for disease prevention and control. However, to the best of the authors' knowledge, there have been no other High-Performance Computing (HPC) implementations dedicated to modeling the propagation of Aedes aegypti mosquitoes.

Similarly, [11] describe how they converted an existing agent-based model for simulating the population dynamics of the Anopheles gambiae mosquito, one of Africa's most significant malaria vectors, to a parallel model. The initial model was created using AGILESim, a Java application designed exclusively for simulating populations of disease vectors and the migrated model was implemented in OpenCL. The authors' findings suggest that using OpenCL for bigger population sizes is particularly successful, with a speed up of 46 times faster than the Java version of AGiLESim. [19] describe the design, implementation, and parallelization of an epidemiological ABM that is used to model the spread of direct contact infectious illnesses using real-world data. High memory consumption and long execution times are both problems addressed by the authors in this research. To address the memory issue, the authors implement an innovative feature which they name the bitstring approach. This approach consists in using an array of bits instead of using conventional data structures to save the attributes for each agent. To address the high computational demands, the authors develop a parallel version of the model aiming multicore CPUs and GPUs architectures. According to the authors' findings, parallelization and the use of the bitstring technique considerably reduced both computational time and memory consumption. Computation time was reduced by 103.25%, and memory use was reduced by 41%.

In addition, [5] present a case study of how an existing agent-based model of Community Associated Methicillin Resistant Staphylococcus Aureus (CA-MRSA5) transmission in Chicago from 2001 to 2010 was parallelized and distributed to produce a scalable general epidemiological model. The authors migrated the model from Repast Simphony to Repast HPC. In comparison to the original approach, the parallelized strategy delivered a 1350% gain in run time performance.

The literature consistently has highlighted the critical role of human behavior in shaping the dynamics of disease spread and has exhibited the limitations of using agent-based models (ABMs) for modeling vector-borne diseases, primarily related to their computational complexity and the inability to conduct real-scale experiments, thus limiting our understanding of virus dynamics. Moreover, there is a gap in incorporating spatial characteristics into these models. Therefore, by building hybrid models within high-performance computing (HPC) environments we can enhance realism by scaling the geographical representation and population size, addressing these limitations.

## 3 Methodology

The transmission dynamics of vector-borne diseases are mostly the same regardless of whether Dengue, Zika, or Chikungunya is being discussed. It occurs from vector to human and from human to vector. For Zika virus, it can occur from human to human [3], but we will not consider this way of transmission.

Humans have four states of infection: Susceptible (S) which is when the human does not have the infection in their body and has not been bitten by a mosquito recently. Exposed (E) is when an infected mosquito has recently bitten the human but the virus can not be transmitted yet because it has not been incubated in the human body. The third stage of infection in the human is the Infected state (I), which is when the virus is in the human body and can be transmitted to other mosquitoes. In this state, is important to consider the infection period, which is the time the virus will be in the human body. When this time period is reached, the human will pass to the last state which is Recovered (R). The infection in the mosquito occurs when a susceptible mosquito bites an infected human. Unlike humans, mosquitoes can be in only 3 states: Susceptible (S), Exposed (E), and Infected (I) and they never recover because their lifespan is about 2 to 3 weeks, so it is not the longest enough to live until a recovery happen. In the model, we will not consider the birth or death of humans, due to the short simulation time contemplated, but it is considered the birth and death of mosquitoes because their lifespan is shorter and their reproduction affects the propagation of these viruses.

Agent-Based Modeling focuses on modeling individuals and the interactions between them and the environment. The behavior of the whole system is obtained as a result of these interactions [14]. Each individual in the model is called an agent and it has its own sets of decision and behavior rules. Agents can be modeled to represent various entities, such as individuals, animals, organizations, or even abstract concepts. They are typically characterized by their state, which includes variables representing their current conditions, characteristics, or attributes. Agents can perceive and sense their environment, gather information, make decisions, and take actions based on predefined rules. They can also interact and communicate with other agents, influencing each other's behavior and potentially leading to emergent properties and complex system-level behaviors. By simulating the actions and interactions of multiple agents, ABMs provides a framework to study the dynamics, patterns, and outcomes of complex systems. It allows researchers to explore how individual agent-level behaviors and interactions can collectively shape the behavior and outcomes of the entire system. ABMs has been widely used in various domains, including social sciences, biology, economics, transportation, and ecology, to gain insights into real-world phenomena and inform decision-making processes.

## 3.1 Conceptual Model

The model consists of two main components: humans and mosquitoes. Humans are represented as individual agents, whereas mosquitoes are modeled as "clouds of mosquitoes" distributed across the spatial area. The spatial representation is achieved using patches, where each patch holds a cloud of mosquitoes with counts for the number of mosquitoes in each state: susceptible, exposed, and infected. These patches represent static agents characterized by state variables indicating temperature and the number of mosquitoes in each state. Each human agent in the model represents one citizen of Bello, and the spatial representation encompasses the entire territorial area of the city of Bello in Antioquia, spanning 149 square kilometers. This spatial layout is organized as a 2D grid with individual squares, and each square contains a patch. The grid dimensions consist of $32 \times 32$ patches.

Each time step of the simulation, also known as a tick, represents one day. The model was run for 365 days. Each day the temperature and the number of mosquitoes in each patch change. Also, each human moves according to the activities that it has assigned, and at the end of the day, each human returns to its house. Each time a human makes a movement, the probability that this human gets infected is calculated and it is determined whether this human gets infected or not in this new place.

The input data for the model consisted of the minimum and maximum temperatures of Bello in 2019, which were used to assign a temperature value to each patch for every day. The model incorporates various parameters, as outlined in Table 1, with values sourced from [16]. To address the inherent parametric uncertainty in individual mosquito models, we adopted the authors' approach, which focuses on addressing heterogeneity in disease spread at the patch level rather than individual mosquito locations.
To determine the temperature of each patch on a given day, a uniform distribution with the minimum and maximum temperatures recorded in Bello during 2019 for that day was employed. To calculate the counts of susceptible, exposed, and infected mosquitoes in each patch at every time step, a system of continuous differential equations is solved using the 4th-order Runge-Kutta numerical method with a time step of $h = 0.1$.

**Table 1.** Parameters values

| Variable | Name | Value |
|---|---|---|
| $\mu_v$ | Per capita mosquito death rate | $\frac{1}{14}$ |
| $\psi_v$ | Per capita natural emergence rate of mosquitoes | 0.3 |
| $\beta_{vh}$ | Probability of transmission from an infectious human to a susceptible mosquito given that a contact between the two occurs | 0.333 |
| $\beta_{hv}$ | Probability of transmission from an infectious mosquito to a susceptible human given that a contact between the two occurs | 0.333 |
| $\sigma_v$ | Maximum number of bites per mosquito per unit time | 0.5 |
| $\sigma_h$ | Maximum number of bites a human can get per unit time | 19 |
| $K_v$ | Carrying capacity of the mosquitoes in the patch | 1000 |

## 3.2 Mosquitoes Behavior

This section presents the process of calculating the number of susceptible (${S_v}^k$), exposed (${E_v}^k$), and infected mosquitoes (${I_v}^k$) in each patch $k$ over time using a system of differential equations. These equations depend on the number of humans in the patch and the temperature of the patch. The equations used for these systems of differential equations were taken from [16] and [17] and are presented in 1.

$$\begin{aligned}\frac{d{S_v}^k}{dt} &= {h_v}^k - {\lambda_v}^k {S_v}^k - \mu_v {S_v}^k \\ \frac{d{E_v}^k}{dt} &= {\lambda_v}^k {S_v}^k - {v_v}^k {E_v}^k - \mu_v {E_v}^k \\ \frac{d{I_v}^k}{dt} &= {v_v}^k {E_v}^k - \mu_v {I_v}^k\end{aligned} \tag{1}$$

The subscript $v$ refers to the mosquito vector, the superscript $k$ refers to the patch, ${h_v}^k$ is the total birth rate of mosquitoes in patch k, ${\lambda_v}^k$ is the per capita rate of infection of mosquitoes in patch k, ${v_v}^k$ is the per capita rate of progression of mosquitoes from exposed state to the infectious state in patch k, and $\mu_v$ is the per capita death rate of mosquitoes (parameter).

The equations for calculating the total birth rate in patch k (${h_v}^k$), per capita rate of infection of mosquitoes in patch k (${\lambda_v}^k$), and per capita rate of progression of mosquitoes from exposed state to the infectious state in patch k (${v_v}^k$) are shown in Eq. 2.

$$
\begin{aligned}
&{h_v}^k = {N_v}^k \left( \psi_v - \frac{r_v * {N_v}^k}{K_v} \right) && r_v = \psi_v - \mu_v \\
&{\lambda_v}^k = {b_v}^k * \beta_{vh} * \left( \frac{{I_h}^k}{{N_h}^k} \right) && {b_v}^k = \frac{b^k}{{N_v}^k} \\
&b^k = \frac{\sigma_v * {N_v}^k * \sigma_h * {N_h}^k}{\sigma_v * {N_v}^k + \sigma_h * {N_h}^k} && {N_v}^k = {S_v}^k + {E_v}^k + {I_v}^k
\end{aligned}
\tag{2}
$$

$\psi_v$ refers to the natural per capita-emergence rate of mosquitoes (parameter), $r_v$ is the mosquito population growth rate, $K_v$ is the carrying capacity of the mosquitoes in a patch (parameter), and ${N_v}^k$ is the total number of mosquitoes in the patch k. The subscript $h$ refers to humans, ${N_h}^k$ is the total number of humans in the patch k, $\beta_{vh}$ is the probability of transmission from an infectious human to a susceptible mosquito given that a contact between the two occurs (parameter), and ${b_v}^k$ is the number of bites per mosquito per unit of time in the patch k. $b^k$ refers to the total number of contacts between humans and mosquitoes (bites) in the patch k, $\sigma_v$ is the maximum number of bites per mosquito per unit of time (parameter) and $\sigma_h$ is the number of bites a human can get per unit of time (parameter).

The equation for calculating the per capita rate of progression of mosquitoes from the exposed state to the infectious state in patch k (${v_v}^k$) is

$$
{v_v}^k = \frac{1}{{tinc_v}^k} \tag{3}
$$

where ${tinc_v}^k$ is the incubation time of the virus in the mosquito in patch $k$. We have made a modification to this equation to establish a dependency for the incubation time on the patch's temperature ($T^k$), with distinct modeling approaches applied for each of the viruses. For Zika, ${tinc_v}^k = 7 + \frac{0.667-0.378(T^k-26)}{0.299+0.027(T^k-26)}$ [22]; for Chikungunya, ${tinc_v}^k = 4 + e^{5.15-0.123T^k}$ [10]; and for Dengue: ${tinc_v}^k$ follows a uniform distribution, more specifically $U(10, 25)$ if $18 < T^k \leq 21$, $U(7, 10)$ if $21 < T^k \leq 26$ and $U(4, 7)$ if $26 < T^k < 31$

For Zika and Dengue viruses, if the temperature of the patch is less than 15 °C, the incubation time of the virus in the mosquitoes is not defined, and the rate of progression of mosquitoes from exposed to infected (${v_v}^k$) is equal to zero. Similarly, for Chikungunya, if the temperature in the patch is less than 12 °C, the mosquito incubation time is not defined in this patch and the rate of progression of mosquitoes from exposed to infected (${v_v}^k$) is equal to zero. This happens because, at low temperatures, the incubation time of the virus in mosquitoes is really long, so mosquitoes reach life expectancy and die before incubating the virus.

## 3.3 Human Behavior

This section focuses on representing human behavior within the model. Humans are assigned specific activities to perform during the day, and their movements are determined accordingly. As they move through different patches, certain variables related to infection state, time since a successful mosquito bite, and time since infection are updated.

At the beginning of each day, human individuals start at their respective homes. The first activity is determined by selecting the activity's coordinates from their list, and the humans are then moved to that specific location. As the probability of infection varies in each patch, the variables related to the human's infection state are updated with every movement they make. This updating process helps to determine if the human will get infected in the new location or not. Then, each human is moved to the coordinates of their second activity, and the relevant variables are updated again. Finally, each human returns to its home location, and the variables related to infection state, time since a successful mosquito bite, and time since infection are updated for the last time.

Unlike the mosquito behavior, which is modeled using differential equations to update state variables, the behavior of humans is modeled stochastically using probabilities. This probabilistic approach accounts for the uncertainties and randomness associated with human movement, infection, and recovery processes. In the model, a susceptible human in patch $k$ can transition to the exposed state with a probability of ${p_{SEh}}^k$. Once in the exposed state, a human can become infected with a probability of ${p_{EIh}}^k$, and an infected human can recover with a probability of ${p_{IRh}}^k$. The probability ${p_{SEh}}^k$ is determined by calculating the rate of infection of humans in patch $k$. On the other hand, both ${p_{SEh}}^k$ and ${p_{IR_h}}^k$ are represented by random variables.

The equation for calculating the probability of a human passing from susceptible to exposed in patch k (${p_{SE_h}}^k$) is the following: ${p_{SE_h}}^k = 1 - e^{-{\lambda_h}^k}$ where ${\lambda_h}^k$ is the rate of infection of humans in patch k and is defined as

$$ {\lambda_h}^k = {b_h}^k * \beta_{hv} * \left( \frac{{I_v}^k}{{N_v}^k} \right) \tag{4} $$

where ${I_v}^k$ is the number of infected mosquitoes in patch k, ${N_v}^k$ is the total number of mosquitoes in patch k, $\beta_{hv}$ is the probability of transmission from an infectious mosquito to a susceptible human given that a contact between the two occurs (parameter), and ${b_h}^k$ is the number of bites a human receives per unit time in the patch k and is defined in the following way: ${b_h}^k = \frac{b^k}{{N_h}^k}$, where $b^k$ is the total number of contacts between humans and mosquitoes (bites) in the patch k (as defined previously) and ${N_h}^k$ is the total number of humans in patch k.

As stated previously, the probability of a human passing from exposed to infected in patch k (${p_{EI_h}}^k$) is calculated by using a random variable approach. In this case the incubation time of the virus in patch k (${t_{inc_h}}^k$) is a random variable that follows a different probability distribution for each virus. For Zika it is $t_{inc_h} \sim \mathit{Weibull}\,(\alpha = 2.69, \beta = 6.70)$ [12]; for Chikungunya it is $t_{inc_h} \sim \mathit{Lognormal}\,(\mu = 1.099,\ \sigma = 0.139)$ [13] and [16]; and for Dengue it is $t_{inc_h} \sim \mathit{Gamma}\,(\alpha =\ 5.5,\ \beta = 1.12)$ [4].

${p_{EI_h}}^k$ is obtained by calculating the probability that the random variable(${t_{inc_h}}^k$) is exceeded by the time passed since the human received a successful bite (timeSinceSuccessfulBite) which is a state variable of the human. The equation for this probability calculation is described below.

$$ {p_{EI_h}}^k = p({t_{inc_h}}^k \leq timeSinceSuccessfulBite) \tag{5} $$

In this manner, every time a human moves, ${p_{EI_h}}^k$ is calculated in the patch k using the human's state variable timeSinceSuccessfulBite. Knowing this probability, it is then used to determine if the human gets infected or not.

The probability of a human passing from infected to recovered in patch k (${p_{IR_h}}^k$) is also calculated by using a random variable approach. In this case, the time of infection of the virus in patch k (${t_{inf_h}}^k$) is a random variable that follows a different probability distribution for each virus. For Zika $t_{inf_h} \sim \mathit{norm}\,(\mu = 6, \sigma = 1)$ [7]; for Chikungunya $t_{inf_h} \sim \mathit{Uniform}\,(a = 3, b = 7)$ [16] and for Dengue $t_{inf_h} \sim \mathit{Uniform}\,(a = 2, b = 7)$ [4].

${p_{IR_h}}^k$ is obtained by calculating the probability that the random variable (${t_{inf_h}}^k$) is exceeded by the time passed since the human got infected (timeSinceInfected) which is a state variable of the human. The equation for this probability calculation is presented below.

$$ {p_{IR_h}}^k = p({t_{inf_h}}^k \leq timeSinceInfected) \tag{6} $$

This probability ${p_{EI_h}}^k$ is then used to determine if the human gets recovered or not.

## 3.4 Computational Implementation

The computational implementations of the conceptual model that we described above, were done using the following two frameworks: Mesa-Geo library, which is an ABMs library implemented in Python, and the Repast Suite. Figure 1 shows the process overview of the model implementation.

Repast Suite is a free and open-source family of agent-based modeling and simulation platforms. It includes Repast Simphony, a Java toolkit that provides a range of features and tools to facilitate the creation, visualization, and analysis of complex systems using agent-based modeling. It also offers Repast HPC, a C++ toolkit that implements the core concepts of Repast Simphony and extends them into a parallel distributed environment.

Mesa-Geo provides a simple and intuitive interface for ABM simulations, making it easier for beginners to get started. As it is built using Python, a widely used and popular programming language known for its simplicity and readability, allows users to leverage the extensive Python ecosystem and easily integrate with other data analysis and visualization tools. It also includes built-in spatial analysis capabilities, allowing users to simulate and analyze geospatial patterns and interactions among agents. It provides tools for handling geographic data and visualizing spatial patterns. One of the disadvantages of Mesa-Geo is that it is primarily designed for small to medium-scale simulations. It may not perform optimally for large-scale simulations with thousands or millions of agents. On the contrary, Repast HPC is specifically designed to handle large-scale agent-based simulations efficiently. It also offers a wide range of advanced features and functionalities, including complex agent behaviors, network modeling, and distributed computing capabilities. However, integrating the geography of a region could be a challenging task that requires more advanced programming knowledge.

## 4 Results

Results obtained from the conducted simulations provide valuable insights into the dynamics of the transmission of the virus in an urban area. In this section, we present an analysis of the outcomes, highlighting key findings and their implications. The results not only shed light on the effectiveness and performance of the three models implemented but also offer valuable information on the impact on simulation capabilities.

The behavior shown by the three implementations reflects the typical behavior of SEIR models. Figure 2 shows the propagation of Dengue among humans over 100 days in the Python implementation. Figures 3 and 4 show the propagation of Dengue, Zika, and Chikungunya during a calendar year, using Repast Simphony and Repast HPC, respectively. For the Mesa-Geo model, we are not showing the results for the three diseases and overall the year due to the high computing time required to run it. Each virus has a peak, as real disease outbreaks do. After those peaks, no significant spread of the disease occurs and the number of susceptible humans stabilizes, which was also expected since we are not considering reinfection for any disease. Also, it is important to notice that we are simulating the viruses in a separate environment, i.e., agents are not exposed to Dengue, Zika, and Chikungunya at the same time.

### 4.1 Mesa-Geo Implementation

Mesa-Geo is a Python-based ABM framework providing built-in core components to easily create, visualize, and analyze simulations. It is one of the most used and actively supported ABM libraries, which exploits Python's popularity to provide ease of use and accessibility [2]. The model implemented in Mesa-Geo was initialized with a total number of humans of 1000 and an initial number of infected humans of 10 and the results are shown on Fig. 2.

Mesa-Geo implements a GeoSpace that can host GIS-based GeoAgents, which are similar to usual Agents, except that they have an attribute for its Coordinate Reference System. It allows to directly create arbitrary geometries or import them from a file. Mesa-Geo allows to create GeoAgents from any vector data file, GeoJSON objects or a GeoPandas GeoDataFrame [21].

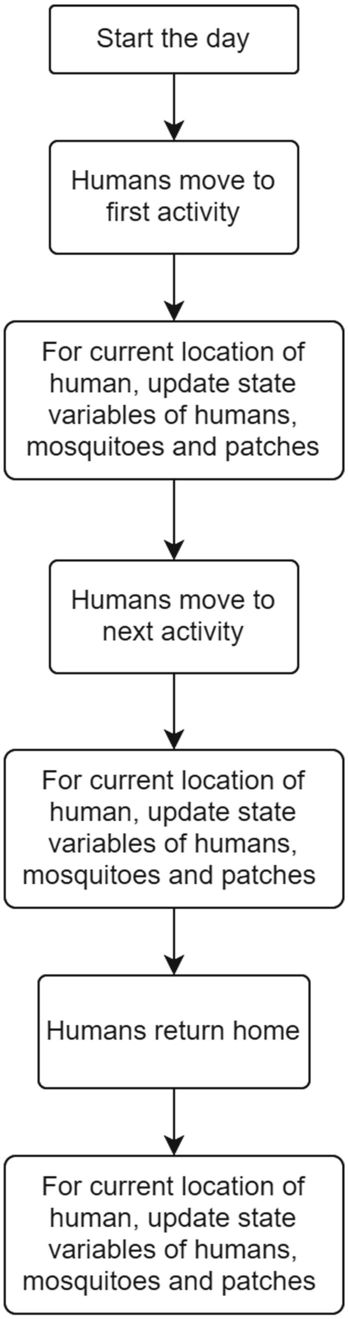

Start the day
Humans move to first activity
For current location of human, update state variables of humans, mosquitoes and patches
Humans move to next activity
For current location of human, update state variables of humans, mosquitoes and patches
Humans return home
For current location of human, update state variables of humans, mosquitoes and patches

**Fig. 1.** Process Overview

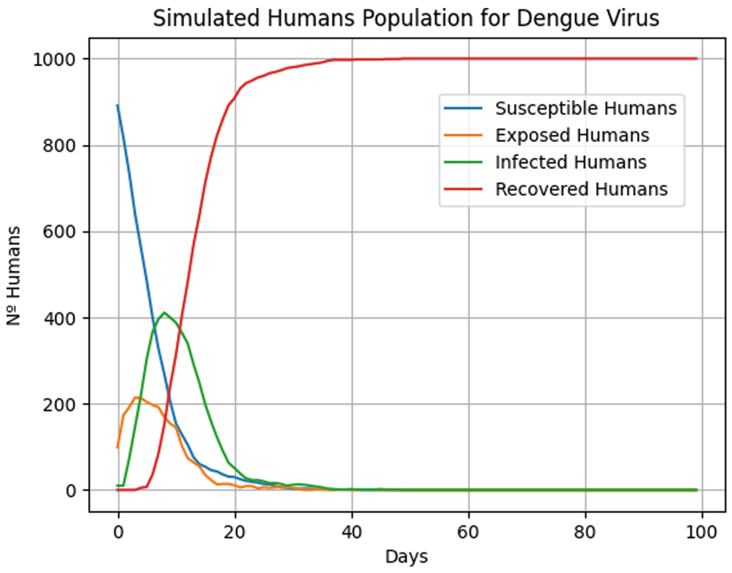


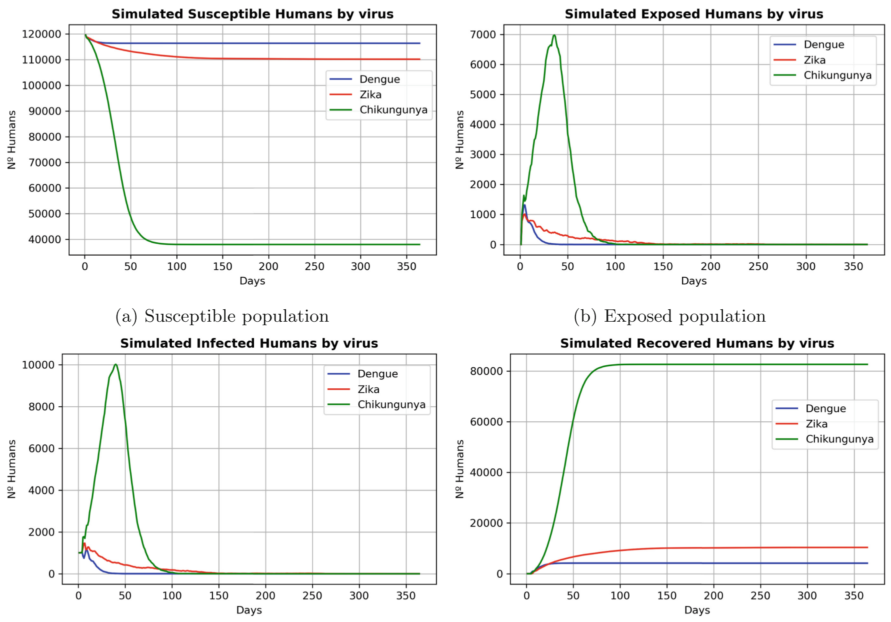
**Fig. 2.** Results of Python Model for Dengue.

Two key factors impact the performance of this model. In the first place, the model includes the representation of all individuals in the city and their respective states (susceptible, exposed, infected, and recovered) each time they move. This process involves updating the state of each individual, which incurs significant computational overhead. As the number of humans increases, the computational burden grows exponentially, leading to longer execution times. In addition to that, the graphical representation of humans as points on a map exacerbates the computational demands of the model. Visualizing each human's position and state in a city requires rendering graphics or updating a graphical user interface (GUI) in real-time. These graphical operations involve additional calculations and memory management, leading to increased computational complexity. Consequently, the model's runtime is significantly prolonged due to the computational cost of maintaining an up-to-date graphical representation of the city and its inhabitants.

## 4.2 Repast Simphony Implementation

Repast Simphony is a richly interactive and easy to learn Java-based modeling toolkit that is designed for use on workstations and small computing clusters. It provides automated methods to perform all the common tasks required in an ABMs simulation [2]. The model implemented in Repast Simphony was initialized with a total number of humans of 120571 and an initial number of infected humans of 1000 and the results are shown on Fig. 3.

Events in simulations are driven by a discrete-event scheduler, and each timestep is equivalent to a tick. Ticks do not necessarily represent clock-time but



(a) Susceptible population (b) Exposed population

(c) Infected population (d) Recovered population

**Fig. 3.** Results of Repast Simphony model for Dengue, Zika and Chikungunya.

rather the priority of its associated event. A context encapsulates the agents, contexts can have projections that establish relationships within agents. For instance, a grid projection allows agents to move in a matrix-type structure where each agent has a location.

## 4.3 Repast HPC Implementation

Repast HPC is an expert-focused C++-based distributed agent-based modeling toolkit that is designed for use on large computing clusters and supercomputers. This toolkit enables the execution of massive simulations containing hundreds of thousands of agents of very complex behavior whose execution requires high computational power [2]. The model implemented on Repast HPC framework was initialized with a total number of humans of 120571 and an initial number of infected humans of 1000 and the results are shown on Fig. 4. Because of the model's geography (which was divided in four sectors), the model was designed to be run with four processes.

Repast HPC is object-oriented. An agent's internal state (e.g. its age, wealth, level of hunger, etc.) is easily represented in an object's fields while the agent's behavior (e.g.eating, aging, acquiring and spending wealth, etc.) is modeled using an object's methods. Agent types are implemented as C++ classes [2].

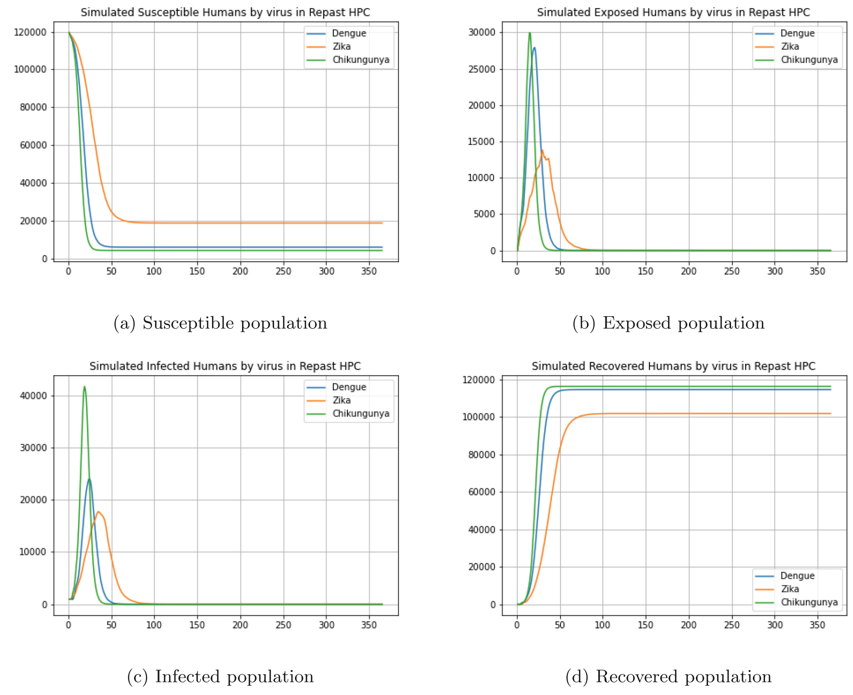


**Fig. 4.** Results of Repast HPC model for Dengue, Zika and Chikungunya.

This framework also incorporates Message Passing Interface (MPI), an industry-standard for message passing. The implementation of MPI allows sending and receiving messages between processes. This includes performing operations on data in transit and synchronizing information from the agents and the geographical context. In MPI applications, a process can not see the other processes' memory. Each process is responsible for the agents local to that process. When a process contains non-local agents, the copies must be updated with the latest state from the original. Movement from one process to another can be the result of grid movement and may be handled automatically by Repast HPC or, in certain cases, it may require manual synchronization.

The model's definition must allow different processes to run in parallel without sharing all memory. Our model fit this description because the Dengue, Zika, and Chikungunya do not have human-to-human transmission. Each subsection of the grid can make computations to update the number of susceptible, exposed, infected, and recovered humans with the information about the mosquitoes and the temperature in that patch. Thus, the parallelization of the model does not violate any assumptions regarding the behavior of the diseases.

An analysis of memory usage yielded an average 31% decrease in memory use over the Repast Simphony models. The improvement in memory usage is notable, however, part of the improvement may be due to the differences between programming languages. Notably, the Repast Simphony model required an increase in memory allocation to run the model. An analysis of the program revealed that the lack of memory was not due to a memory leak. Thus, it is related with the way Java handles memory and garbage collection causes excessive memory use. Unfortunately, the java memory management scheme is automated and is not subject to change. In contrast, it is possible to improve the results obtained in the Repast HPC models through more customized memory management.

## 4.4 Comparative Analysis

This section presents a comparative analysis of three different agent-based modeling implementations aimed at studying vector-borne disease dynamics in urban settings. The goal of this analysis is to gain valuable insights into the strengths and limitations of each implementation and its potential applicability in understanding disease dynamics and developing control strategies.

Table 2 provides a comparison of the three distinct agent-based modeling implementations. The comparison considers essential factors such as the programming language used, computer memory usage, and parallelization support. These factors play a critical role in determining the efficiency and scalability of the models. Additionally, we examine the ease of model development and the visualization capabilities, which are crucial for aiding non-experts, such as health decision-makers.

**Table 2.** Comparative Analysis of the Agent-Based Modeling Implementations.

| Aspect | Python | Repast Simphony | Repast HPC |
|---|---|---|---|
| Programming Language | Python | Java | C++ |
| Ease of Model Development | Easy | Moderate | High |
| Computer Memory Usage | High | High | Moderate |
| Parallelization Support | Limited | Limited | High |
| Scalability | Limited | Moderate | High |
| Visualization Capabilities | Yes | Limited | Limited |

Regarding programming language and ease of model development, Java, Python, and C++ are all free and open-source programming languages. Python's syntax is simpler and more concise compared to Java and C++. Java follows a class-based object-oriented approach and has a syntax similar to C and C++. Python stands out in ease of model development due to its straightforward process, whereas Repast Simphony and Repast HPC require a moderate and high level of expertise, especially for non-experienced programmers. Python is an easier language to learn, while Java and C++ require more experience and programming skills. Our aim for the model is to assist health decision makers. We seek to create a user-friendly and intuitive platform that allows non programming experts to use the model effectively, enabling them to gain valuable insights and make informed decisions. To use it on a day-to-day basis, Python could be more approachable due to its simplicity, readability, and extensive libraries and frameworks. Finally, Python has included in the library Mesa-Geo an easy way to visualize the spread of the disease in a graphical interface. Repast Simphony also allows a graphical interface but is not as easy to implement as in Python. Repast HPC does not include a graphical interface.

In terms of computer memory usage, Python and Repast Simphony exhibits relatively high consumption, while Repast HPC require moderate memory usage. The high memory consumption in Python and Repast Simphony can be attributed, on the one hand, to the graphical interface of the library, and on the other hand, to the memory management approach employed by these programming languages, which lack flexibility. Conversely, C++ offers a flexible memory management system that can be customized to meet the requirements of each processor.

The parallelization support aspect examines the capacity to enable multiple processes to be executed simultaneously [8]. Python and Repast Simphony offer limited support, while Repast HPC excels in high parallelization. Additionally, the scalability aspect assess the ability of an implementation to handle larger and more complex simulations. The key advantages of Repast HPC over Repast Simphony or Python-based ABMs are its focus on parallelization, efficient resource utilization, and compatibility with high-performance computing environments. This allows it to handle more extensive and complex simulations, making it a better choice for scalability when dealing with large-scale agent-based models.

All three implementations support the incorporation of spatial and temporal dimensions in the modeling approach, as well as agent heterogeneity and environmental interaction capabilities. Agent heterogeneity is represented by modeling agents with diverse characteristics and behaviors, and environmental interaction examines whether the implementations allow agents to interact with the environment.

Overall, this comparative analysis aids researchers in understanding the different features and performance characteristics of each implementation, enabling them to make informed decisions regarding the choice of agent-based modeling approach for studying vector-borne disease dynamics in urban settings, based on their specific requirements and constraints.

## 5 Conclusions and Future Research

In this paper, we explored the use of agent-based modeling and simulation (ABMs) to analyze the dynamics of vector-borne disease transmission, specifically focusing on the Aedes aegypti mosquito and its role in spreading diseases such as Dengue, Chikungunya, and Zika. We conducted an exploratory study in the city of Bello, Colombia, which possesses characteristics that make it suitable for examining the impact of dengue in densely populated urban areas. Our study aimed to address the limited research conducted on ABMs models that capture the complex dynamics of vector-borne disease transmission in urban areas. We compared three distinct agent-based modeling implementations using different programming languages and simulation toolkits: Python, Repast Simphony (Java), and Repast HPC. By evaluating factors such as computational capacity, computer memory usage, and implementation characteristics, we aimed to provide valuable insights into the strengths and limitations of each implementation and contribute to the advancement of agent-based modeling approaches for studying vector-borne disease dynamics in urban settings.

The results obtained from the three implementations showed the typical behavior of SEIR models, with each virus exhibiting a peak and the number of susceptible humans stabilizing over time. The Python implementation using Mesa-Geo provided a simple and intuitive interface, while the Repast Simphony implementation offered advanced features and tools for complex agent behaviors. The Repast HPC implementation demonstrated the scalability and efficiency of HPC in handling large-scale simulations.

We encountered computational difficulties while running simulations for the three models on a desktop computer, preventing us from experimenting with the actual population of the city of Bello. Although the HPC model showed satisfactory results in reducing memory usage, our next step is to test the models on a supercomputer since they have been specifically optimized for such devices. Furthermore, as part of our future work, we plan to explore the possibilities of parallelizing the Python model to enhance its performance and speed. By leveraging parallel computing techniques, we can achieve significant improvements in execution time. The substantial decrease in processing time will enable us to conduct numerous experiments, perform a comprehensive sensitivity analysis, estimate parameters, and validate the models. As a result, we will gain a deeper understanding of the transmission dynamics of the virus in an urban area.

In conclusion, our research contributes to bridging the gap in ABMs models for studying vector-borne disease transmission in urban areas. The comparative analysis of different implementations provides insights into their strengths and limitations, paving the way for further advancements in agent-based modeling approaches. By incorporating spatial, social, and climate characteristics and leveraging the computational power of HPC, researchers can gain valuable insights into disease dynamics, assess the effectiveness of control strategies, and make informed decisions for disease prevention and control efforts in urban settings.

## References

1. Alderton, S., et al.: A multi-host agent-based model for a zoonotic, vector-borne disease. a case study on trypanosomiasis in eastern province, Zambia. PLoS Negl. Trop. Dis. **10**(12), e0005252 (2016). https://doi.org/10.1371/journal.pntd.0005252
2. Antelmi, A., Cordasco, G., D'Ambrosio, G., De Vinco, D., Spagnuolo, C.: Experimenting with agent-based model simulation tools. Appl. Sci. **13**(1), 13 (2022). https://doi.org/10.3390/app13010013

3. Caminade, C., et al.: Global risk model for vector-borne transmission of zika virus reveals the role of el niño 2015. Proc. Natl. Acad. Sci. **114**(1), 119–124 (2017)
4. Chan, M., Johansson, M.A.: The incubation periods of dengue viruses. PLoS ONE **7**, 1–7 (2012). https://doi.org/10.1371/journal.pone.0050972
5. Collier, N., Ozik, J., Macal, C.M.: Large-scale agent-based modeling with repast HPC: a case study in parallelizing an agent-based model. In: Hunold, S., et al. (eds.) Euro-Par 2015. LNCS, vol. 9523, pp. 454–465. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27308-2_37
6. España, G., et al.: Exploring scenarios of chikungunya mitigation with a data-driven agent-based model of the 2014–2016 outbreak in Colombia. Sci. Rep. **8**(1), 12201 (2018). https://doi.org/10.1038/s41598-018-30647-8
7. Fontaine, A., de Laval, F., Belleoud, D., Briolant, S., Matheus, S.: Duration of zika viremia in serum. Clin. Infect. Dis. Off. Publ. Infect. Dis. Soc. Am. **67**, 1143–1144 (2018). https://doi.org/10.1093/cid/ciy261
8. Hager, G., Wellein, G.: Introduction to High Performance Computing for Scientists and Engineers. CRC Press, Boca Raton (2010). https://doi.org/10.1201/EBK1439811924
9. Jindal, A., Rao, S.: Agent-based modeling and simulation of mosquito-borne disease transmission. In: Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems, pp. 426–435 (2017)
10. Kakarla, S.G., et al.: Temperature dependent transmission potential model for chikungunya in India. Sci. Total Environ. **647**, 66–74 (2019). https://doi.org/10.1016/j.scitotenv.2018.07.461
11. Kofler, K., Davis, G., Gesing, S.: SAMPO: an agent-based mosquito point model in OpenCL. In: Proceedings of the 2014 Symposium on Agent Directed Simulation, pp. 1–10 (2014)
12. Krow-Lucal, E.R., Biggerstaff, B.J., Staples, J.E.: Estimated incubation period for zika virus disease. Emerg. Infect. Dis. **23**, 841–844 (2017). https://doi.org/10.3201/eid2305.161715
13. Leung, C.: Estimated incubation period for mosquito-borne disease-related Guillain-Barre syndrome. Clin. Epidemiol. Glob. Health **8**, 244–250 (2020). https://doi.org/10.1016/j.cegh.2019.08.007
14. Macal, C.M., North, M.J.: Agent-based modeling and simulation: abms examples. In: 2008 Winter Simulation Conference, pp. 101–112. IEEE (2008). https://doi.org/10.1109/WSC.2008.4736060
15. Mahmood, I., Jahan, M., Groen, D., Javed, A., Shafait, F.: Correction to: an agent-based simulation of the spread of dengue fever. In: Krzhizhanovskaya, V.V., et al. (eds.) ICCS 2020. LNCS, vol. 12139, pp. C1–C1. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50420-5_49
16. Manore, C.A., et al.: A network-patch methodology for adapting agent-based models for directly transmitted disease to mosquito-borne disease. J. Biol. Dyn. **9**(1), 52–72 (2015). https://doi.org/10.1080/17513758.2015.1005698
17. Mniszewski, S.M., Manore, C., Bryan, C., Del Valle, S.Y., Roberts, D.: Towards a hybrid agent-based model for mosquito borne disease. In: Summer Computer Simulation Conference: (SCSC 2014): 2014 Summer Simulation Multi-Conference: Monterey, California, USA, 6–10 July 2014. Summer Computer Simulation Conference (2014: Monterey, calif.), vol. 2014. NIH Public Access (2014)
18. Montes de Oca, E.S., Suppi, R., De Giusti, L.C., Naiouf, M.: Green high performance simulation for AMB models of Aedes aegypti. J. Comput. Sci. Technol. **20** (2020). https://doi.org/10.24215/16666038.20.e02

19. Rizzi, R.L., Kaizer, W.L., Rizzi, C.B., Galante, G., Coelho, F.C.: Modeling direct transmission diseases using parallel bitstring agent-based models. IEEE Trans. Comput. Soc. Syst. **5**(4), 1109–1120 (2018). https://doi.org/10.1109/TCSS.2018.2871625
20. Scheidegger, A.P.G., Banerjee, A.: An agent-based model to investigate behavior impacts on vector-borne disease spread. In: 2017 Winter Simulation Conference (WSC), pp. 2833–2844. IEEE (2017). https://doi.org/10.1109/WSC.2017.8248007
21. Wang, B., Hess, V., Crooks, A.: Mesa-geo: a GIS extension for the mesa agent-based modeling framework in python. In: Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoSpatial Simulation, pp. 1–10. GeoSim '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3557989.3566157
22. Winokur, O.C., Main, B.J., Nicholson, J., Barker, C.M.: Impact of temperature on the extrinsic incubation period of zika virus in aedes aegypti. PLoS Negl. Trop. Dis. **14**, 1–15 (2020). https://doi.org/10.1371/journal.pntd.0008047