

EPIGUI: Graphical User Interface for Simulating Epidemics on Networks

E. R. PINTO¹, E. G. NEPOMUCENO², J. KUSAK³ and A. S. L. O. CAMPANHARO^{4*}

Received on February 10, 2022 / Accepted on July 18, 2022

ABSTRACT. This paper presents a graphical interface called EPIGUI, which allows the study of the dynamics of an infectious disease spread using compartmental models in combination with complex networks. This interface aims at considering stochastic factors that govern the evolution of an infection in a network. Moreover, it provides simple tools to create networks of agents and to define the epidemiological parameters of outbreaks. There are six common infectious disease models (SI, SIS, SIR, SIRS, SEIR, and SEIRS) or a user can provide another model, combining compartments. Moreover, in the simulations the user can either include a synthetic network generated according to a network model (random, small-world, scale-free, modular, or hierarchical) or a real network. This approach can help understand the paths followed by outbreaks in a given community and design new strategies to prevent and to control them.

Keywords: agent-based model, complex networks, infectious diseases, computational graphical interface.

1 INTRODUCTION

Infectious diseases or communicable diseases have been a cause of panic since ancient times, decimating populations. They are usually caused by a biological agent (for example, virus or bacteria). Currently, the world is facing the impact of COVID-19 which, in addition to being a threat to public health, with the loss of countless lives, has brought enormous economic damage to all countries [11]. Another disease that terrified the world was the black death wiping out a quarter of the European population during the years 1347-1350 [7]. The number of deaths caused by the great epidemics is incomparably higher than the number of deaths caused by all wars together [2].

*Corresponding author: Andriana S. L. O. Campanharo – E-mail: andriana.campanharo@unesp.br

¹Department of Biostatistics, Institute of Biosciences, São Paulo State University (UNESP), Botucatu, São Paulo, Brazil – E-mail: eduardo.pinto@unesp.br <https://orcid.org/0000-0001-5057-6911>

²Centre for Ocean Energy Research and Department of Electronic Engineering, Maynooth University, Ireland – E-mail: erivelton.nepomuceno@mu.ie <https://orcid.org/0000-0002-5841-2193>

³Faculty of Computer Science, Electronics and Telecommunications, AGH University of Science and Technology, Krakow, Poland – E-mail: justynakusak17@gmail.com

⁴Department of Biostatistics, Institute of Biosciences, São Paulo State University (UNESP), Botucatu, São Paulo, Brazil – E-mail: andriana.campanharo@unesp.br <https://orcid.org/0000-0002-0501-5303>

Considering this scenario, numerous research fronts have been developed for the understanding and prevention of infectious diseases. Mathematical epidemiology has provided essential support to political and scientific authorities, inferring information about the epidemiological impact of a vaccine, identifying which and how many agents (or individuals) should be vaccinated, verifying whether the disease under study is epidemic or endemic and making projections for the number of infected people. [5] expresses the main reason for this interest when says: “The main reasons for studying mathematical models of disease proliferation is the hope that a better understanding of transmission mechanisms can provide strategies for more effective control.”

Mathematical epidemiology proved to be decisive in the policy for more restrictive isolation measures in countries in Europe, such as the United Kingdom. Scientists at Imperial College London made a set of mathematical predictions that contributed to convincing Prime Minister Boris Johnson to change the course of his policy [34]. In general, this document prepared epidemiological scenarios that can be synthesized taking into account the control action adopted and the expected number of deaths. In this sense, mathematical epidemiology has benefited from techniques and methodologies typical of Engineering and Exact Sciences.

With the increase in computational power, the use of software for simulating epidemics has become essential, and numerous graphic interfaces have emerged, such as, **GLEaMviz** [33], **STEM** [10], **OutbreakTools** [20] and **epidemix** [28]. The epidemiological modelling used in these programs is largely based on compartmental models, that is, groups of agents who share identical properties [2, 8]. In particular, the **sispread** [1] software and the **hybridModels** package in R [26] have attracted attention due to the use of complex networks. Unlike the classic models of differential equations, this approach does not use the assumption of a homogeneous mixture, in which the relationships between agents are randomized. An example in which the topology of a real network cannot be described by a purely random network, was demonstrated by the SARS outbreak, where the average number of secondary cases was about two for all infected agents [24].

It is worth mentioning that **sispread** and **hybridModels** software have some limitations. Firstly, they do not consider agents to be discrete and their local interactions. Secondly, the **sispread** software has only three epidemiological models (SI, SIS and SIR) and three complex networks models (Random, Scale-free and Kleinberg) that can be combined, not allowing the user to implement another model and use a network of your choice (real or synthetic). Thirdly, the **hybridModels** package does not consider interactions between agents, only between sub-populations. Furthermore, it requires some knowledge about the R language.

To overcome this lack, at least partially, the software proposed in this article implements a graphical user interface for simulating epidemics using the Agent-Based Model (ABM) - EPIGUI. Our method uses a bottom-up strategy and begins by creating rules for the behaviour of agents and, based on the interaction between these agents, tries to understand the properties that emerge from the system [16]. An advantage of agent-based models over traditional models is that they have a more flexible construction and can incorporate any number of agent-level mechanisms, using aspects that are difficult or impossible to represent in population-level differential equations. In

addition, stochasticity and the low profile of the population can play an important role in the evolution of the system. In a model described by an Ordinary Differential Equation (ODE), each process (for example, infection and recovery) has a rate. This approach assumes that the time evolution is continuous and deterministic and the number of agents can be a non-integer number for each time which is unreal [13]. Alternatively, instead of focusing on a process rate, models can be developed on agent reaction events that do not necessarily occur at regular intervals. The timing of these events is stochastic rather than deterministic [26].

The EPIGUI allows the user to simulate the spread of infectious diseases using compartmental models and complex networks. The software combines the advantages of ABM's stochasticity with the realistic contacts that complex network models or network databases can provide. Such models include random networks by [12]; small-world networks by [35]; scale-free networks by [3]; modular networks by [32] and hierarchical networks by [9]. Network databases are openly available in several repositories, such as the Koblenz Network Collection (KONECT) [22] and the Stanford large network dataset collection (SNAP Datasets) [23]. Finally, the interface is user-friendly, it does not require proficiency in any programming language and it is freely available on the internet.

The remainder of this paper is organized as follows: after this introduction, Sect. 2 describes the SI, SIS, SIR, SIRS, SEIR, and SEIRS compartment models, the Agent-Based Models for the compartment models previously described, as well several network models. The equivalence between each compartment model described in Sect. 2 and the corresponding Agent Based Model is built in Sect. 3. The installation, platform availability, and main elements of the EPIGUI software are presented in Sect. 4. Usage examples of the EPIGUI software are described in Sect. 5. The conclusions are presented in Sect. 6.

2 BACKGROUND

2.1 Compartmental models

2.1.1 SI

The SI model divides the population into two disjoint classes, ie, the number of susceptible agents (S) and number of agents infected (I) [18]. It is defined by a continuous system of two ordinary differential equations. This model (Eq. (2.1)) describes the temporal evolution of each epidemiological class, that is:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} - \mu S, & S(0) = S_0 \geq 0, \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \mu I, & I(0) = I_0 \geq 0, \end{cases} \quad (2.1)$$

where μ and β are birth/death and contact rates, respectively [18].

2.1.2 SIS

The SIS model admits that infected agents can recover and return to the susceptible state unlike the SI model, which assumes that agents remain infected for life [18]. The SIS model is defined by a continuous system of two ordinary differential equations. This model (Eq. (2.2)) describes the temporal evolution of each epidemiological class, that is:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} + \gamma I - \mu S, & S(0) = S_0 \geq 0, \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, & I(0) = I_0 \geq 0, \end{cases} \quad (2.2)$$

where μ and β are the same as described in Subsection 2.1.1 and γ is the recovery rate.

2.1.3 SIR

The SIR model admits that infected agents can recover from the infection and acquire lifetime immunity, moving to the recovered state (R) [21]. The classic SIR model is defined by a continuous system of three ordinary differential equations. This model (Eq. (2.3)) describes the temporal evolution of each epidemiological class, that is:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} - \mu S, & S(0) = S_0 \geq 0, \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, & I(0) = I_0 \geq 0, \\ \frac{dR}{dt} = \gamma I - \mu R, & R(0) = R_0 \geq 0, \end{cases} \quad (2.3)$$

where μ , β and γ are birth/death, contact and recovery rates, respectively.

2.1.4 SIRS

The SIRS model admits that recovered agents can return to the susceptible state after losing immunity over time [17, 19]. The SIRS model is defined by a continuous system of three ordinary differential equations. This model (Eq. (2.4)) describes the temporal evolution of each epidemiological class, that is:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} + \rho R - \mu S, & S(0) = S_0 \geq 0, \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, & I(0) = I_0 \geq 0, \\ \frac{dR}{dt} = \gamma I - \rho R - \mu R, & R(0) = R_0 \geq 0, \end{cases} \quad (2.4)$$

where μ , β and γ are the same as described in Subsection 2.1.3 and ρ is the proportion of agents that lose immunity.

2.1.5 SEIR

Several diseases have an intermediate state between the susceptible (S) and the infected states (I), called latent or exposed state (E), which can also be represented by a compartment. This new compartment was added to the SIR model due to the fact that an agent may be infected, but not yet developed the disease and infected other agents [25]. The SEIR model is defined by a continuous system of four ordinary differential equations. This model (Eq. (2.5)) describes the temporal evolution of each epidemiological class, that is:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} - \mu S, & S(0) = S_0 \geq 0, \\ \frac{dE}{dt} = \frac{\beta IS}{N} - \gamma E - \mu E, & E(0) = E_0 \geq 0, \\ \frac{dI}{dt} = \gamma E - \rho I - \mu I, & I(0) = I_0 \geq 0, \\ \frac{dR}{dt} = \rho I - \mu R, & R(0) = R_0 \geq 0, \end{cases} \quad (2.5)$$

where μ , β , γ , and ρ are birth/death, contact, latency and recovery rates, respectively.

2.1.6 SEIRS

Unlike the SEIR model, the SEIRS model admits that recovered agents can return to the susceptible state after losing immunity over time. The SEIRS model is defined by a continuous system of four ordinary differential equations [25]. This model (Eq. (2.6)) describes the temporal evolution of each epidemiological class, that is:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} + \varepsilon R - \mu S, & S(0) = S_0 \geq 0, \\ \frac{dE}{dt} = \frac{\beta IS}{N} - \gamma E - \mu E, & E(0) = E_0 \geq 0, \\ \frac{dI}{dt} = \gamma E - \rho I - \mu I, & I(0) = I_0 \geq 0, \\ \frac{dR}{dt} = \rho I - \varepsilon R - \mu R, & R(0) = R_0 \geq 0, \end{cases} \quad (2.6)$$

where μ , β , γ and ρ are the same as described in Subsection 2.1.5 and ε is the proportion of agents that lose immunity.

2.2 Agent-Based Model (ABM)

Agent-Based Models simulate populations and communities by following agents and their properties [15]. These models are developed to address important features such as agent variability, local interactions and adaptive behaviours. In the ABM, each agent in the population has a set of characteristics such as state or behaviours. According to [14], “each agent is treated as a unique

and discrete entity that has age and at least one more property that changes over the life cycle, such as weight, social position, among others”.

The ABM can be expressed by a $P_{m \times n}$ matrix, where each of its rows represents an agent and each of its columns represents a characteristic. For each time step t , its matrix (population) can be represented by:

$$P_t = \begin{bmatrix} C_{1,1}^t & C_{1,2}^t & C_{1,3}^t & \cdots & C_{1,n}^t \\ C_{2,1}^t & C_{2,2}^t & C_{2,3}^t & \cdots & C_{2,n}^t \\ C_{3,1}^t & C_{3,2}^t & C_{3,3}^t & \cdots & C_{3,n}^t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{m,1}^t & C_{m,2}^t & C_{m,3}^t & \cdots & C_{m,n}^t \end{bmatrix}. \tag{2.7}$$

As previously mentioned, this formulation is quite general, allowing to incorporate several attributes of agents. For the compartmental models SI, SIS, SIR, SIRS, SEIR, and SEIRS described in Section 2.1, and already implemented in EPIGUI, the characteristics are based on the work of [31].

2.2.1 SI and SIS model

In this work, the ABM for the SI model with vital dynamic consists of the following rules:

- C_1 is the state of an agent regarding the epidemic (susceptible (0) and infected (1)).
- C_2 is the age of an agent (in units of time). At $t = 0$, $C_2 = 0$ and for each iteration, Δt is added to the age.
- C_3 is the maximum age an agent can live. At the birth time, this age is obtained by:

$$C_3 = -\mu_{abm} \log(a_u), \tag{2.8}$$

where μ_{abm} is the life expectancy of the population and a_u is a random variable uniformly distributed between 0 and 1.

The ABM for the SIS model has the same characteristics and transition rules of the SI model, except that, for SIS, two new features are added:

- C_4 is the time (in units of time) that an agent gets infected. At $t = 0$, $C_4 = 0$.
- C_5 is the maximum time that an agent remains infected and it is obtained by:

$$C_5 = -\gamma_{abm} \log(a_u), \tag{2.9}$$

where γ_{abm} is the infection period.

2.2.2 SIR and SIRS model

In this work, the ABM for the SIR model with vital dynamic consists of the following rules:

- C_1 is the state of an agent regarding the epidemic (susceptible (0), infected (1) and recovered (2)).
- C_2 is the age of an agent (in units of time). At $t = 0$, $C_2 = 0$ and for each iteration, Δt is added to the age.
- C_3 is the maximum age an agent can live. At the birth time, this age is obtained by:

$$C_3 = -\mu_{abm} \log(a_u), \quad (2.10)$$

where μ_{abm} is the life expectancy of the population and a_u is a random variable uniformly distributed between 0 and 1.

- C_4 is the time (in units of time) that an agent gets infected. At $t = 0$, $C_4 = 0$.
- C_5 is the maximum time that an agent remains infected and it is obtained by:

$$C_5 = -\gamma_{abm} \log(a_u), \quad (2.11)$$

where γ_{abm} is the infection period.

The ABM for the SIRS model has the same characteristics and transition rules of the SIR model, except that, for SIRS, two new features are added:

- C_6 is the time (in units of time) that an agent gets recovered. At $t = 0$, $C_6 = 0$.
- C_7 is the maximum time that an agent remains recovered and it is obtained by

$$C_7 = -\rho_{abm} \log(a_u), \quad (2.12)$$

where ρ_{abm} is the recovered period.

By definition, in the SIR and SIRS models, for susceptible and recovered agents $C_4 = 0$ and $C_5 = 0$ for all $t \geq 0$.

2.2.3 SEIR and SEIRS model

In this work, the ABM for the SEIR model with vital dynamic consists of the following rules:

- C_1 is the state of an agent regarding the epidemic (susceptible (0), exposed (1), infected (2), and recovered (3)).
- C_2 is the age of an agent (in units of time). At $t = 0$, $C_2 = 0$ and for each iteration, Δt is added to the age.

- C_3 is the maximum age an agent can live. At the birth time, this age is obtained by:

$$C_3 = -\mu_{abm} \log(a_u), \tag{2.13}$$

where μ_{abm} is the life expectancy of the population and a_u is a random variable uniformly distributed between 0 and 1.

- C_4 is the time (in units of time) that an agent gets exposed. At $t = 0$, $C_4 = 0$.
- C_5 is the maximum time that an agent remains exposed and it is obtained by:

$$C_5 = -\gamma_{abm} \log(a_u), \tag{2.14}$$

where γ_{abm} is the exposed period.

- C_6 is the time (in units of time) that an agent gets infected. At $t = 0$, $C_6 = 0$.
- C_7 is the maximum time that an agent remains infected and it is obtained by:

$$C_7 = -\rho_{abm} \log(a_u), \tag{2.15}$$

where ρ_{abm} is the infected period.

The ABM for the SEIRS model has the same characteristics and transition rules of the SEIR model, except that, for SEIRS, two new features are added:

- C_8 is the time (in units of time) that an agent gets recovered. At $t = 0$, $C_6 = 0$.
- C_9 is the maximum time that an agent remains recovered and it is obtained by:

$$C_9 = -\varepsilon_{abm} \log(a_u), \tag{2.16}$$

where ε_{abm} is the recovered period.

For the SEIR and SEIRS models, the characteristics C_4 to C_7 are unnecessary for susceptible and recovered agents. For all models presented, at time t the numbers of susceptible, exposed, infected, and recovered agents are indicated by $S(t)$, $E(t)$, $I(t)$ and $R(t)$, respectively. Moreover, the population has a constant size, that is, $N = S(t) + E(t) + I(t) + R(t)$ for all $t \geq 0$.

2.3 Complex networks

A complex network can be described by $R = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ represents a set of N vertices or nodes and $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$ represents set of M connections or edges which connect pairs of vertices. Complex network theory relies on the use of mathematical metrics that are able to quantify different features of the network's topology, such as, the average degree ($\langle k \rangle$), the clustering coefficient ($\langle CC \rangle$), the shortest path length (ℓ), and the global efficiency (E) [6].

2.3.1 Complex network models

Several complex networks models have been proposed in the literature with the aim of reproducing patterns of connections found in real networks [29]. In this work, random, small-world, scale-free, modular and hierarchical models were used to generate undirected and unweighted networks to better understand the implications of these patterns in the propagation of a disease.

Random model (RAN) Paul Erdős and Alfred Rényi proposed the first model of complex networks (or graphs), which became known as random graphs of Erdős and Rényi [12]. According to this model, N disconnected vertices are connected with a probability p between $[0, 1]$. For $p = 0$ the network is disconnected, for $p = 1$ the network is fully connected, and for any $0 < p < 1$ the network is random.

Small-world model (SW) Duncan Watts and Steven Strogatz analyzed empirical data of several real networks, such as the neuron network of *Caenorhabditis elegans* and the United States power distribution network, and noted that the connections were not completely regular or completely random, however it was situated between these two extremes [35]. Thus, they proposed a model that consists of an interpolation between a regular and a random network. Networks procuded by this model are highly clustered, as regular networks, but with a short path length between their nodes, as random graphs.

Scale-free model (SF) Real-world networks are often claimed to be scale-free, meaning that the degree distribution follows $P(k) \approx k^{-\gamma}$, where $2 \leq \gamma \leq 3$. The scale-free model used in this work, proposed by Albert-Laszlo Barabasi and Reka Albert, is capable of generating scale-free networks based on the the concepts of growth and preferential attachment [3].

Modular model (MOD) A modular network model is able to produce networks with a modular structure, that is, with edges densely distributed between vertices belonging to the same group (module) and sparingly distributed between vertices belonging to different groups. In this work, the algorithm proposed by Pratha Sah, known as random modular model, is used to create modular networks [32].

Hierarchical model (HIE) The scale-free model previously described lacks the small-world effect that can be found in most real complex networks, i.e. its clustering coefficient is too small while the average length path remains relatively large. Hierarchical models are able to overcome this deficiency, creating networks that have similar characteristics of scale-free and small-world networks at the same time. In this work, the algorithm proposed by Peter Dodds *et al.* is used to create hierarchical networks [9].

3 REFORMULATION OF THE ABM

The set of Eqs. (3.1) is defined to guarantee the discrete nature of ABM and to follow the parameters of the models previously described in Sec. 2.1. Using Eqs. (3.1), it is possible to build the equivalence between an ODE model and an ABM, in such a way that, on average, their solutions present similar behaviours [31].

$$\left\{ \begin{array}{l} \beta_{abm} = \beta_{ode}\Delta t, \\ \gamma_{abm} = 1/\gamma_{ode}, \\ \mu_{abm} = 1/\mu_{ode}, \\ \rho_{abm} = 1/\rho_{ode}, \\ \epsilon_{abm} = 1/\epsilon_{ode}. \end{array} \right. \quad (3.1)$$

Figure 1 presents the results of the ABM (dashed lines) in comparison with its equivalent compartmental model (solid lines), over 100 realizations of the ABM. The general parameters are set to show an epidemic state as follows. Fig. 1a: $\mu_{ode} = 0.0143$, $\beta_{ode} = 0.125$, $N = 1,000$, $S(0) = 800$, $I(0) = 200$, $\Delta t = 0.1$ and $t_f = 100$; Fig. 1b: $\mu_{ode} = 0.0143$, $\beta_{ode} = 0.125$, $\gamma_{ode} = 0.1$, $N = 1,000$, $S(0) = 800$, $I(0) = 200$, $\Delta t = 0.1$ and $t_f = 100$; Fig. 1c: $\mu_{ode} = 0.0143$, $\beta_{ode} = 1.5$, $\gamma_{ode} = 0.0278$, $N = 1,000$, $S(0) = 800$, $I(0) = 200$, $R(0) = 0$, $\Delta t = 0.1$ and $t_f = 100$; Fig. 1d: $\mu_{ode} = 0.0143$, $\beta_{ode} = 1.5$, $\gamma_{ode} = 0.0278$, $\rho_{ode} = 0.1$, $N = 1,000$, $S(0) = 800$, $I(0) = 200$, $R(0) = 0$, $\Delta t = 0.1$ and $t_f = 100$; Fig. 1e: $\mu_{ode} = 0.2$, $\beta_{ode} = 45$, $\gamma_{ode} = 3$, $\rho_{ode} = 4$, $N = 1,000$, $S(0) = 800$, $E(0) = 0$, $I(0) = 200$, $R(0) = 0$, $\Delta t = 0.01$ and $t_f = 10$; Fig. 1f: $\mu_{ode} = 0.2$, $\beta_{ode} = 45$, $\gamma_{ode} = 3$, $\rho_{ode} = 4$, $\epsilon_{ode} = 3$, $N = 1,000$, $S(0) = 800$, $E(0) = 0$, $I(0) = 200$, $R(0) = 0$, $\Delta t = 0.01$ and $t_f = 10$. It is worth mentioning that although the model and parameters above described are not realistic, they were used to illustrate the potential of the proposed approach.

4 INSTALLATION AND PLATFORM AVAILABILITY

The EPIGUI software is freely available on Linux and Windows operating systems. The installation size depends on the system. On Linux, it is currently around 76.8 MB (megabyte) compressed. On Windows, it is currently around 126 MB compressed. The software is available for Linux at: https://www.dropbox.com/s/a17n6pkr8ofixi0/EPIGUI_LINUX.zip?dl=0 and for Windows at: https://www.dropbox.com/s/x7thdbouojonjhn/EPIGUI_WINDOWS.zip?dl=0. It only needs to be uncompressed to any user-writable folder and can be run without any other installation steps (as shown in the Appendix A). EPIGUI runs as a standalone application, written in Python and C languages.

4.1 Main elements of the user interface

The main user elements and features of the EPIGUI interface are presented as follows. First, there is a dialog box that offers the user the choice between writing a system or using a predefined one. If the user chooses to write the system, then the number n of epidemiological states in the model

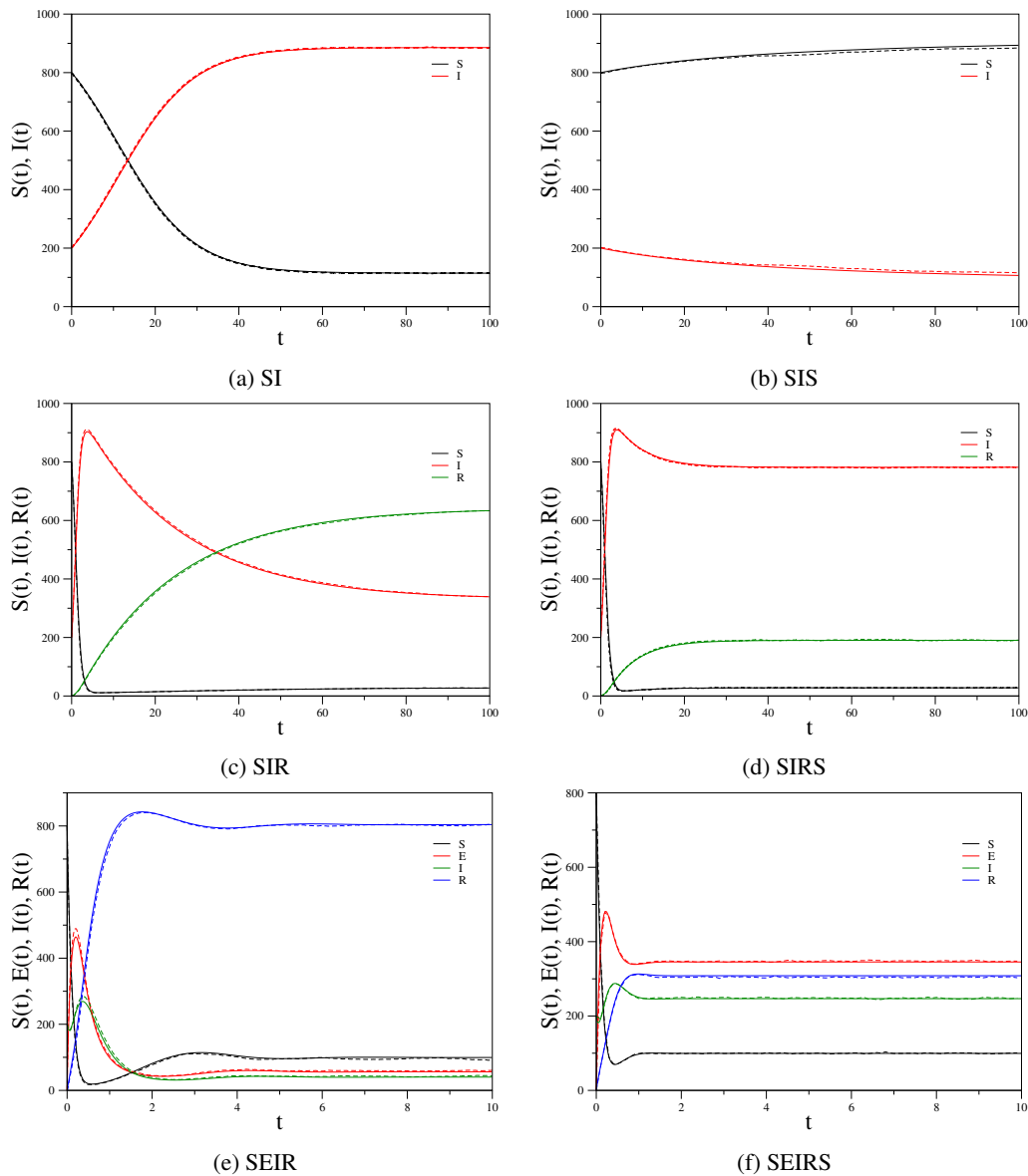


Figure 1: Comparison of the ODE systems solutions (solid lines) and the corresponding ABM solutions (dashed lines), averaged over 100 different simulations for the (a) SI, (b) SIS, (c) SIR, (d) SIRS, (e) SEIR, and (f) SEIRS models, with $t_f = 100$. ((a), (b), (c), and (d)) and $t_f = 10$ ((e) and (f)).

must be informed. After that, n boxes will appear to be named. The number of parameters m can be chosen and named. Below the parameter descriptions, there is a “Writer System” button in which the user can write the model of equations, the initial conditions and the values of the parameters, as shown in Figure 2. Note that each variable, each parameter and each arithmetic operation written in the writer system must be separated with single spaces.



Figure 2: The EPIGUI interface. In this interface, the user is able to set up the number of compartments, parameters and write the correspondent equation. Additionally, the type of network and respective parameters are also disposable to be configured.

The epidemic model with vital dynamics can be considered. In this case, a mortality rate μ is added to all compartments, and after reaching the maximum age, agents return to the first compartment. After, the number of simulations, the time spacing and the final time must be informed. The interface allows the user to change all parameters, to run it and to see the results in plot graphs. However, the interface does not check whether the parameters values are reasonable for the simulation.

The last group of parameters to be defined corresponds to the complex network model. For illustration, the random network model with $p = 1$ was chosen. Other options include the small world, scale-free, modular and hierarchical network models. As previously described in Section 2.3.1, each model has at least one box to be filled out with the corresponding parameter value. Each parameter has a tooltip with its description, which is activated by placing the mouse pointer on it. At the bottom of Figure 2 there is a box with an article reference of the corresponding network model.

Finally, the user can click on the green box called “Run simulation” and a graph plot is displayed (Fig. 3a). In the example, the graph plot corresponds to the average number of agents (susceptible,

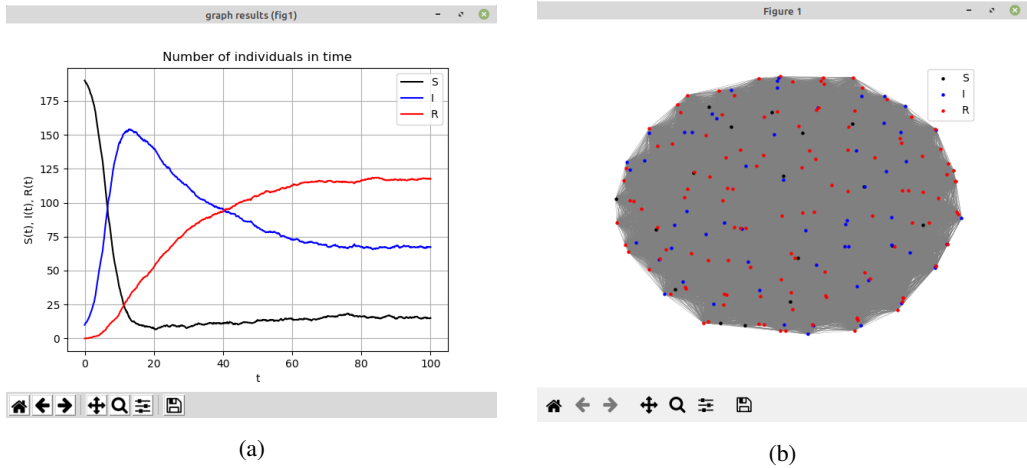









Figure 3: (a) Time series for susceptible, infected and recovered agents and (b) network graph, which each node represents an agent coloured according to its compartment.

infected and recovered) as a function of time. More specifically, the ABM solution based on the SIR model for an epidemic state (basic reproductive number $\mathcal{R}_0 > 1$) with $\mu = 0.0143$, $\gamma = 0.0278$, $\beta = 0.5$, $N = 200$, $S(0) = 190$, $I(0) = 10$, $R(0) = 0$, $\Delta t = 0.1$, $t_f = 100$, and $p = 1$, over 10 different simulations. After closing the graph plot, a button named “Show graph” will be enabled, allowing the user to visualize the layout of the corresponding network in its final time step (Fig. 3b). In this example, the susceptible agents (black nodes), infected agents (blue nodes) and recovered agents (red nodes) for $t_f = 100$ and $p = 1$.

At the bottom of Figs. 3 (a) and (b) can be found the following options:

-  Return to the initial settings, ie, auto-scales the graph to contain all data points for all visible sets.
-  Back to the previous view.
-  Forward to the next view.
-  Pan mode for axes (left mouse button) and move mode for axes (right mouse button).
-  Zoom in the axis view.
-  Configure subplots.
-  Save the image in one of the following formats: EPS, PN, JPEG, JPG, PDF, PS, RAW, RGBA, SVG, SVGZ, TIF, or TIFF.

After closing the network plot, a button named “Show measures” will be enabled, allowing the user to visualize six different network measures (Fig. 4) of the corresponding network (Fig. 3b).

At the bottom of Figure 4 there is a box with an article reference of the corresponding network measures.

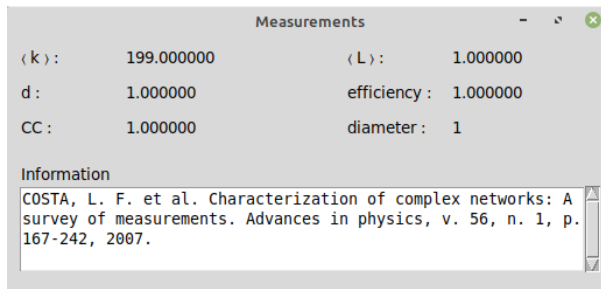


Figure 4: EPIGUI allows the user to show some basic measures of the network, which calculations are based on reference illustrated.

5 EXAMPLES

5.1 SEI model

The SEI model can be used for modelling the tuberculosis spread [30]. The population is divided into three compartmental classes, that is, agents who have never had contact with the bacillus, called susceptible (S); agents who have the bacillus but do not develop the disease, called exposed or latent (E) and agents who develop pulmonary tuberculosis, responsible for the transmission of the disease, called infected (I).

Figure 5 presents the interaction flowchart between agents in the classes S , E and I . Parameters μ , β , δ , and ρ represent the rates of mortality, infection, exogenous infection, and relapse, respectively. A portion q of susceptible agents who contract the bacillus agent, does not develop tuberculosis in a period γ^{-1} after infection, while $(1 - q)$ develops pulmonary tuberculosis.

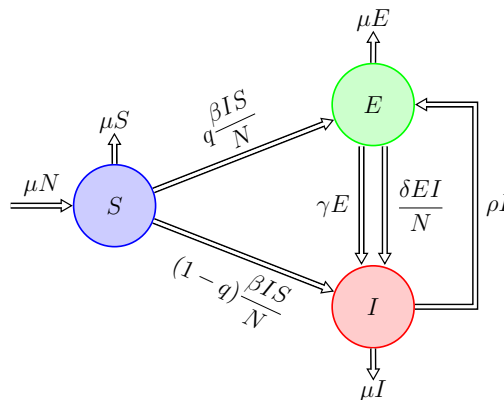


Figure 5: Flowchart of susceptible, exposed and infected agents for the SEI model.

The following ODE system can be written based on the flowchart (Fig.5), that is [27]:

$$\begin{cases} \frac{dS}{dt} = \mu N - \frac{\beta IS}{N} - \mu S, \\ \frac{dE}{dt} = q \frac{\beta IS}{N} - \left(\gamma + \frac{\delta I}{N} \right) E + \rho I - \mu E, \\ \frac{dI}{dt} = (1 - q) \frac{\beta IS}{N} + \left(\gamma + \frac{\delta I}{N} \right) E - \rho I - \mu I. \end{cases} \quad (5.1)$$

Figure 6 shows the configuration of the EPIGUI based on Eq.(5.1), using a real infection network.¹ Note that $S(0) + E(0) + I(0)$ has to be equal to the number of vertices in the network to properly run the interface. Figure 7a shows the average number of susceptible, exposed and infected agents, over 10 different simulations as a function of time. More specifically, the ABM solution based on Eq.(5.1) with $N = 410$, $\Delta t = 0.1$, $q = 0.9$, $\gamma = 0.0027$, $\rho = 0.0083$, $\mu = 0.0143$, $\delta = 0.0867$, $\beta = 4.3$, $S(0) = 400$, $E(0) = 0$ and $I(0) = 10$. Figure 7b shows the corresponding network layout in the final time step $t_f = 500$. Black, blue and red vertices correspond to susceptible, exposed and infected agents, respectively. Figure 8 shows the network measures of the corresponding network for the ABM.

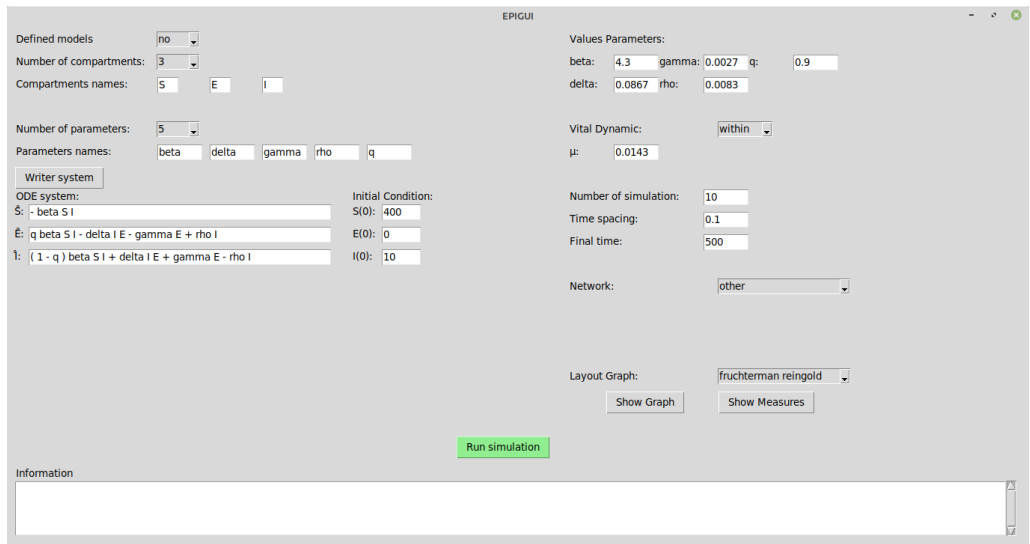


Figure 6: Configuration of the SEI model in the EPIGUI interface.

¹The infection network file can be found in the Appendix B.

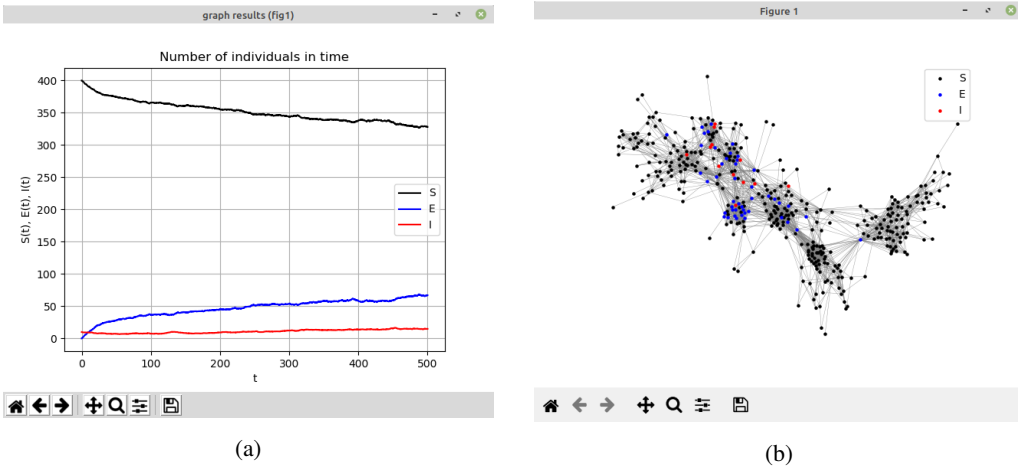


Figure 7: (a) Average number of susceptible, exposed and infected agents as a function of time, over 10 different simulations and (b) Network layout in the final time step $t_f = 500$.

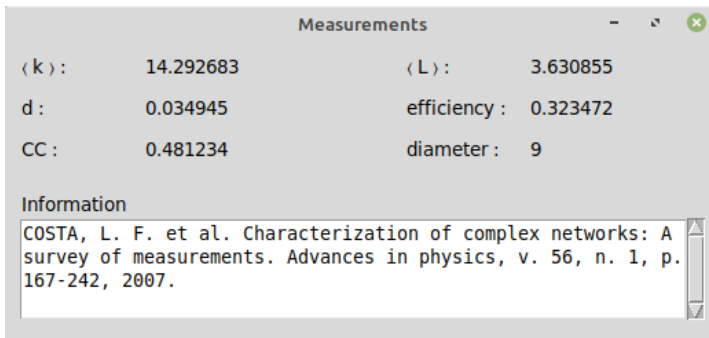


Figure 8: Network measures of the real infection network for the ABM of the SEI model and $t_f = 500$.

5.2 SIRV model

The SIR model with vaccination (SIRV) is considered as a way of reducing epidemic incidences. In this model, there is a class of vaccinated agents (V), that is, agents who are not infected, are vaccinated and acquire temporary immunity, besides the susceptible (S), infected (I) and recovered (R) classes. The SIRV model [4], defined by a continuous system of four ordinary differential equations considers that the distribution of agents is spatially and temporally homogeneous.

Figure 9 presents the interaction flowchart between agents in the classes S, I, R , and V . Parameters μ, β and γ represent the rates of mortality, infection and recovery, respectively. Note that a

portion p_v of susceptible agents are vaccinated and acquire immunity by moving to the vaccinated compartment (V).

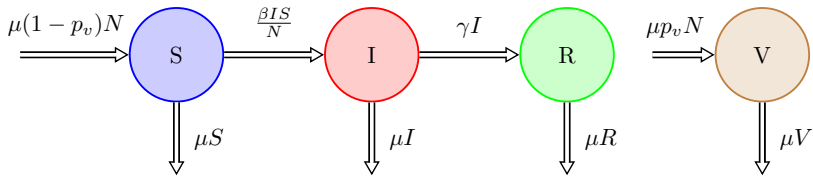


Figure 9: Flowchart of susceptible, infected, recovered, and vaccinated agents for the SIRV model.

The following ODE system can be written based on the flowchart (Fig.9), that is [4]:

$$\begin{cases} \frac{dS}{dt} = \mu(1 - p_v)N - \frac{\beta IS}{N} - \mu S, \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I, \\ \frac{dR}{dt} = \gamma I - \mu R, \\ \frac{dV}{dt} = \mu p_v N - \mu V. \end{cases} \quad (5.2)$$

The flowchart in Fig. 9 can be redesigned taking into account the following rules: $\alpha = \mu(1 - p_v)$, $\rho = \mu p_v$ and $\mu = \alpha + \rho$ (Fig. 10). Therefore, the system previously described in Eq. (5.2) can be rewritten (Eq.(5.3)) [31].

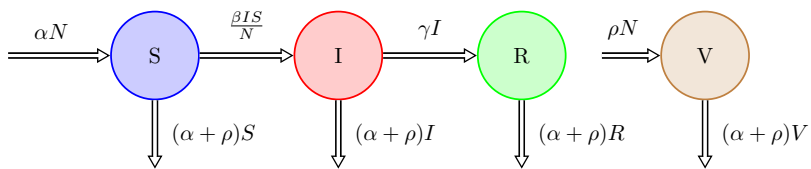


Figure 10: Flowchart of susceptible, infected, recovered, and vaccinated agents for the modified SIRV model.

$$\left\{ \begin{aligned} \frac{dS}{dt} &= \alpha N - \frac{\beta IS}{N} - (\alpha + \rho)S, \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I - (\alpha + \rho)I, \\ \frac{dR}{dt} &= \gamma I - (\alpha + \rho)R, \\ \frac{dV}{dt} &= \rho(S + I + R) - \alpha V. \end{aligned} \right. \quad (5.3)$$

Figure 11 shows the configuration of the EPIGUI based on Eq.(5.3), using a small-world network with $n_0 = 20$ and $p = 0.01$. Note that $S(0) + E(0) + I(0)$ has to be equal to the number of vertices in the network to properly run the interface. Figure 12a shows the average number of susceptible, infected, recovered, and vaccinated agents, over 10 different simulations as a function of time. More specifically, the ABM solution based on Eq.(5.3) with $N = 200$, $\Delta t = 0.1$, $p_v = 0.9$, $\beta = 0.5$, $\gamma = 0.0278$, $\mu = 0.0143$, $\alpha = 0.00143$, $\rho = 0.01287$, $S(0) = 190$, $I(0) = 10$, $R(0) = 0$, and $V(0) = 0$. Figure 12b shows the corresponding network layout in the final time step $t_f = 100$. Black, blue, red, and green vertices correspond to susceptible, infected, recovered, and vaccinated agents. Figure 13 shows the network measures of the corresponding network for the ABM. It is worth mentioning that although the models and parameters previously described (Sec. 5.1 and 5.2) are not realistic, they were used to illustrate the potential of the proposed approach.

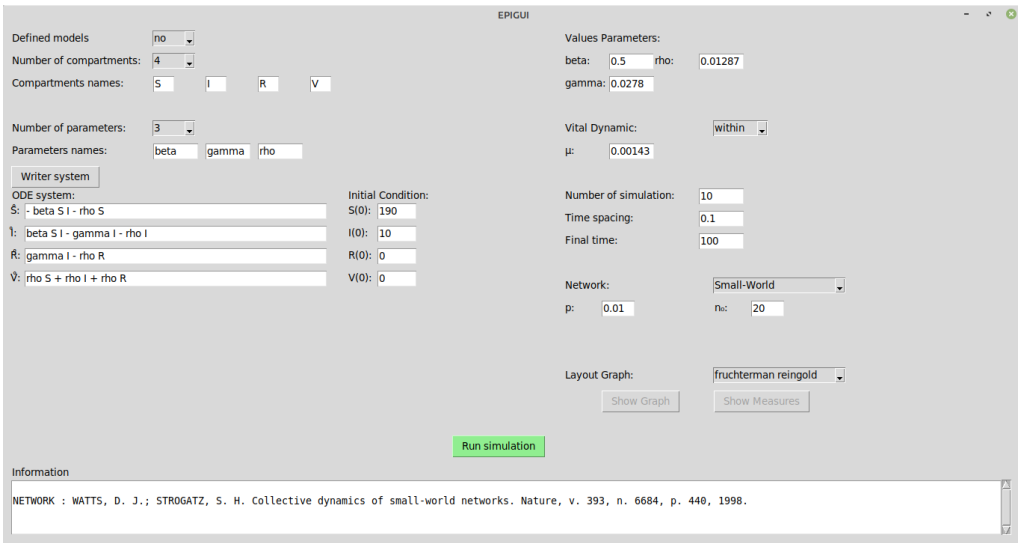


Figure 11: Configuration of the *SIRV* model in the EPIGUI interface.

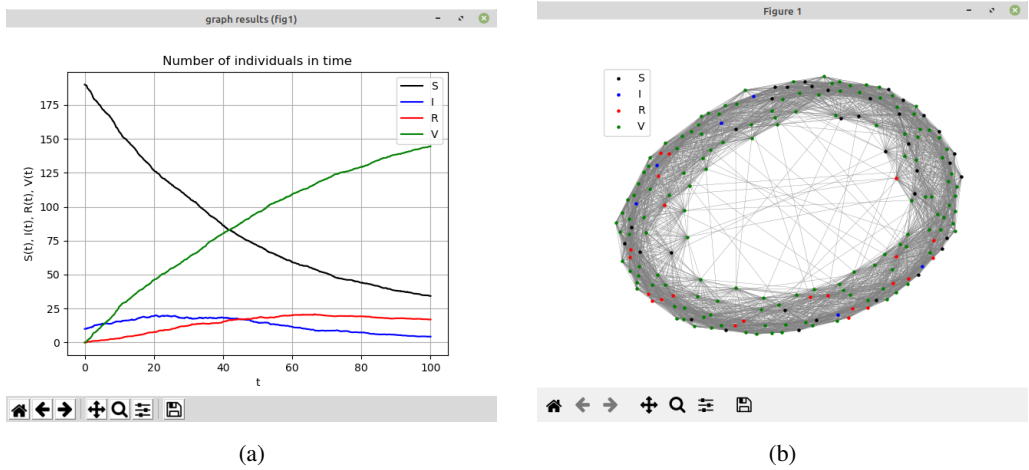


Figure 12: (a) Average number of susceptible, infected, recovered, and vaccinated agents as a function of time, over 10 different simulations and (b) Network layout in the final time step $t_f = 100$.

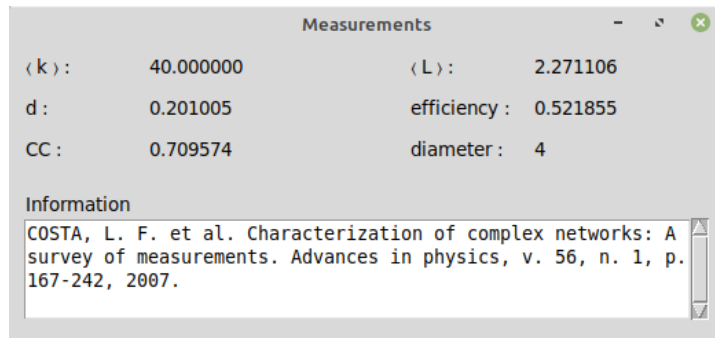


Figure 13: Network measures of the small-world network for the ABM of the SIRV model and $t_f = 100$.

6 CONCLUSIONS

This paper has introduced the EPIGUI interface for simulating epidemics. Although there is a significant number of epidemiological packages in public domain [10, 20, 28, 33] to model epidemiological systems using compartmental models, less attention has been given to the discrete aspect of the population or the impact of network topology of population. The EPIGUI interface was developed to fill this gap by incorporating stochasticity in real epidemics together with agent contact networks. This approach is certainly of great help to understand paths followed by outbreaks in a given community and to design new strategies to prevent and control them.

Our interface was developed for users, proficient or not in any programming language, to study infectious disease spread using compartmental models in combination with complex networks. In the disease outbreak simulation, the user can combine one of the six compartmental models widely known in the literature (SI, SIS, SIR, SIRS, SEIR, or SEIRS) or another provided by the user with one of the five complex network models (random, small-world, scale-free, modular, or hierarchical) or a real network openly available in a database. Unified access to different modelling approaches under the same visualization platform allows users to compare assumptions, parameters and differences in results under different conditions. Thus, with the use of the EPIGUI interface users may better understand main concepts of the dynamics of infectious diseases and also appreciate the impact of various modelling assumptions and control interventions.

Acknowledgments

E. R. Pinto acknowledges the support of Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), grant 1770124. A. S. L. O. Campanharo acknowledges the support of Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grant 2018/25358-9.

REFERENCES

- [1] F. Alvarez, P. Crépey, M. Barthélemy & A.J. Valleron. sisspread: A software to simulate infectious diseases spreading on contact networks. *Methods of information in medicine*, **46** (2007).
- [2] R.M. Anderson & R.M. May. “Infectious Diseases of Humans: Dynamics and Control”. Oxford University Press (1992).
- [3] A.L. Barabási & R. Albert. Emergence of scaling in random networks. *Science*, **289** (1999).
- [4] S. Chauhan, O.P. Misra & J. Dhar. Stability analysis of SIR model with vaccination. *American Journal of computational and applied mathematics*, **4** (2014).
- [5] D. Clancy. Optimal intervention for epidemic models with general infection and removal rate functions. *Journal of Mathematical Biology*, **39** (1999).
- [6] L.F. Costa, F.A. Rodrigues, G. Travieso & P.R. Villas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, **56** (2007).
- [7] N.J. Cox, S.E. Tamblin & T. Tam. Influenza pandemic planning. *Vaccine* **21**, **16** (2003).
- [8] O. Diekmann & J.A.P. Heesterbeek. “Mathematical epidemiology of infectious diseases: model building, analysis and interpretation”, volume 5. John Wiley & Sons (2000).
- [9] P.S. Dodds, D.J. Watts & C.F. Sabel. Information exchange and the robustness of organizational networks. *PNAS*, **100** (2003).
- [10] J.V. Douglas, S. Bianco, S. Edlund, T. Engelhardt, M. Filter, T. Günther, K. Hu, E.J. Nixon, N.L. Sevilla, A. Swaid *et al.* Stem: An open source tool for disease modeling. *Health security*, **17** (2019).
- [11] A. Du Toit. Outbreak of a novel coronavirus. *Nature Reviews Microbiology*, **18** (2020).

- [12] P. Erdős & A. Rényi. On random graphs. *Publicationes Mathematicae*, **6** (1959).
- [13] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of physical chemistry*, **81** (1977).
- [14] V. Grimm. Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological modelling*, **115** (1999).
- [15] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S.K. Heinz, G. Huse *et al.* A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, **198** (2006).
- [16] V. Grimm & S.F. Railsback. “Individual-based modeling and ecology”, volume 8. Princeton university press (2005).
- [17] H.W. Hethcote. Qualitative analyses of communicable disease models. *Mathematical Biosciences*, **28** (1976).
- [18] H.W. Hethcote. Three basic epidemiological models. In “Applied mathematical ecology”. Springer (1989).
- [19] H.W. Hethcote, H.W. Stech & P. Van Den Driessche. Nonlinear oscillations in epidemic models. *SIAM Journal on Applied Mathematics*, **40** (1981).
- [20] T. Jombart, D.M. Aanensen, M. Baguelin, P. Birrell, S. Cauchemez, A. Camacho, C. Colijn, C. Collins, A. Cori, X. Didelot *et al.* OutbreakTools: a new platform for disease outbreak analysis using the R software. *Epidemics*, **7** (2014).
- [21] W.O. Kermack & A.G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, **115** (1927).
- [22] J. Kunegis. Konect: the koblenz network collection. In “Proceedings of the 22nd international conference on World Wide Web” (2013), p. 1343–1350.
- [23] J. Leskovec & A. Krevl. SNAP Datasets: Stanford large network dataset collection (2014).
- [24] J.O. Lloyd-Smith, S.J. Schreiber, P.E. Kopp & W.M. Getz. Superspreading and the effect of individual variation on disease emergence. *Nature*, **438** (2005).
- [25] I. Longini & N. Bailey. THE GENERALIZED DISCRETE-TIME EPIDEMIC MODEL WITH IMMUNITY. In “Vol. 2 Mathematical Statistics Theory and Applications”. De Gruyter (2020).
- [26] F.S. Marques, M. Amaku & J. Grisi-Filho. hybridModels: An R Package for Stochastic Simulation of Disease Spreading in Dynamic Networks. *R package version 0.2*, **15** (2017).
- [27] V. Moreno, B. Espinoza, M. Paredes, D. Bichara, M. A. & C. CastilloChavez. The role of mobility and health disparities on the transmission dynamics of Tuberculosis. *Theoretical Biology and Medical Modeling*, **143** (2017).
- [28] U. Muellner, G. Fournié, P. Muellner, C. Ahlstrom & D.U. Pfeiffer. epidemix—An interactive multi-model application for teaching and visualizing infectious disease transmission. *Epidemics*, **23** (2018).

- [29] M.E.J. Newman. “Networks: An Introduction”. Oxford University Press (2010).
- [30] E.R. Pinto, E.G. Nepomuceno & A.S.L.O. Campanharo. Influence of Contact Network Topology on the Spread of Tuberculosis. In “Latin American Workshop on Computational Neuroscience”. Springer (2019).
- [31] E.R. Pinto, E.G. Nepomuceno & A.S.L.O. Campanharo. Impact of network topology on the spread of infectious diseases. *Trends in Computational and Applied Mathematics*, **21** (2020).
- [32] P. Sah, L.O. Singh, A. Clauset & S. Bansal. Exploring community structure in biological networks with random graphs. *BMC Bioinformatics*, **15** (2014).
- [33] W. Van den Broeck, C. Gioannini, B. Gonçalves, M. Quaggiotto, V. Colizza & A. Vespignani. The GLEaMviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale. *BMC infectious diseases*, **11** (2011).
- [34] P.G.T. Walker, C. Whittaker, O. Watson, M. Baguelin, K.E.C. Ainslie, S. Bhatia, S. Bhatt, A. Boonyasiri, O. Boyd, L. Cattarino, Z. Cucunubá, G. Cuomo-Dannenburg, A. Dighe, C.A. Donnelly, I. Dorigatti, S. Van Elsland, R. Fitzjohn, S. Flaxman, H. Fu, K. Gaythorpe, L. Geidelberg, N. Grassly, W. Green, A. Hamlet, K. Hauck, D. Haw, S. Hayes, W. Hinsley, N. Imai, D. Jorgensen, E. Knock, D. Laydon, S. Mishra, G. Nedjati-Gilani, L.C. Okell, S. Riley, H. Thompson, J. Unwin, R. Verity, M. Vollmer, C. Walters, W. Wang, Y. Wang, P. Winskill, X. Xi, N.M. Ferguson & A.C. Ghani. The Global Impact of COVID-19 and Strategies for Mitigation and Suppression. *Imperial College COVID-19 Response Team*, (2020).
- [35] D.J. Watts & S.H. Strogatz. Collective Dynamics of “Small World” Networks. *Letters to Nature*, **393** (1998).



APPENDIX A SOFTWARE DOWNLOAD

The EPIGUI software is available on the Linux and Windows operating systems. The installation size depends on the system. On Linux, it is currently around 76.8 MB (megabyte) compressed. On Windows, it is currently around 126 MB compressed.

A.1 Linux

1. Download the EPIGUI_LINUX.zip file accessing the web address:
https://www.dropbox.com/s/a17n6pkr8ofixi0/EPIGUI_LINUX.zip?dl=0
2. Unzip the file EPIGUI_LINUX.zip.
3. Open the EPIGUI_LINUX folder.
4. Execute install.sh from command prompt or double-clicking it.
5. Run the executable "EPIGUI" from command prompt or double-clicking it.

A.1 Windows

1. Download the EPIGUI_WINDOWS.zip file accessing the web address:
https://www.dropbox.com/s/x7thdbouojonjhn/EPIGUI_WINDOWS.zip?dl=0
2. Unzip the file EPIGUI_WINDOWS.zip.
3. Open the EPIGUI_WINDOWS folder.
4. Run the executable EPIGUI.exe.

APPENDIX B REPLICATION RESULTS

Figure 1a

1. Select the **yes** option in the **Defined models** dialog box;
2. Select the **SI** option in the **Choose the model** dialog box;
3. Write the initial system conditions under the **Initial Condition**, ie,
 first box (**S(0)**): 800
 second box (**I(0)**): 200
4. Write the parameter value under **Values Parameters**, ie, **beta**: 0.125
5. Select the **within** option in the **Vital dynamic** dialog box;
6. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.0143;
7. Write the simulation parameters, ie,
 Number of simulation: 100
 Time spacing: 0.1
 Final time: 100

8. Select the **Random** option in the **Network** dialog box;
9. Write the parameter of the random network, that is, $p = 1$;
10. Click in the **Run simulation** button.

Figure 1b

1. Select the **yes** option in the **Defined models** dialog box;
2. Select the **SIS** option in the **Choose the model** dialog box;
3. Write the initial system conditions under the **Initial Condition**, ie,
first box (**S(0)**): 800
second box (**I(0)**): 200
4. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 0.125
second box (**gamma**): 0.1
5. Select the **within** option in the **Vital dynamic** dialog box;
6. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.0143;
7. Write the simulation parameters, ie,
Number of simulation: 100
Time spacing: 0.1
Final time: 100
8. Select the **Random** option in the **Network** dialog box;
9. Write the parameter of the random network, that is, $p = 1$;
10. Click in the **Run simulation** button.

Figure 1c

1. Select the **yes** option in the **Defined models** dialog box;
2. Select the **SIR** option in the **Choose the model** dialog box;
3. Write the initial system conditions under the **Initial Condition**, ie,
first box (**S(0)**): 800
second box (**I(0)**): 200
third box (**R(0)**): 0
4. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 1.5
second box (**gamma**): 0.0278
5. Select the **within** option in the **Vital dynamic** dialog box;

6. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.0143;
7. Write the simulation parameters, ie,
 Number of simulation: 100
 Time spacing: 0.1
 Final time: 100
8. Select the **Random** option in the **Network** dialog box;
9. Write the parameter of the random network, that is, $p = 1$;
10. Click in the **Run simulation** button.

Figure 1d

1. Select the **yes** option in the **Defined models** dialog box;
2. Select the **SIRS** option in the **Choose the model** dialog box;
3. Write the initial system conditions under the **Initial Condition**, ie,
 first box (**S(0)**): 800
 second box (**I(0)**): 200
 third box (**R(0)**): 0
4. Write the parameter values under **Values Parameters**, ie,
 first box (**beta**): 1.5
 second box (**gamma**): 0.0278
 third box (**rho**): 0.1
5. Select the **within** option in the **Vital dynamic** dialog box;
6. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.0143;
7. Write the simulation parameters, ie,
 Number of simulation: 100
 Time spacing: 0.1
 Final time: 100
8. Select the **Random** option in the **Network** dialog box;
9. Write the parameter of the random network, that is, $p = 1$;
10. Click in the **Run simulation** button.

Figure 1e

1. Select the **yes** option in the **Defined models** dialog box;
2. Select the **SEIR** option in the **Choose the model** dialog box;

3. Write the initial system conditions under the **Initial Condition**, ie,
first box (**S(0)**): 800
second box (**E(0)**): 0
third box (**I(0)**): 200
fourth box (**R(0)**): 0
4. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 45
second box (**gamma**): 3
third box (**rho**): 4
5. Select the **within** option in the **Vital dynamic** dialog box;
6. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.2;
7. Write the simulation parameters, ie,
Number of simulation: 100
Time spacing: 0.01
Final time: 10
8. Select the **Random** option in the **Network** dialog box;
9. Write the parameter of the random network, that is, $p = 1$;
10. Click in the **Run simulation** button.

Figure 1f

1. Select the **yes** option in the **Defined models** dialog box;
2. Select the **SEIRS** option in the **Choose the model** dialog box;
3. Write the initial system conditions under the **Initial Condition**, ie,
first box (**S(0)**): 800
second box (**E(0)**): 0
third box (**I(0)**): 200
fourth box (**R(0)**): 0
4. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 45
second box (**gamma**): 3
third box (**rho**): 4
fourth box (**epsilon**): 3
5. Select the **within** option in the **Vital dynamic** dialog box;
6. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.2;

7. Write the simulation parameters, ie,
Number of simulation: 100
Time spacing: 0.01
Final time: 10
8. Select the **Random** option in the **Network** dialog box;
9. Write the parameter of the random network, that is, $p = 1$;
10. Click in the **Run simulation** button.

Figure 3a

1. Select the **no** option in the **Defined models** dialog box;
2. Select the second option (3 compartments) in the **Number of compartments** dialog box;
3. Write the names of the compartments **S**, **I** and **R** in the three boxes, respectively;
4. Select the second option (2 parameters) in the **Number of parameters** dialog box;
5. Write the names of the parameters **beta** and **gamma** in the two boxes, respectively;
6. Click in **Writer system**;
7. Write the system of differential equations in the boxes below the **ODE system** (each parameter, variable and sign must be separated by spaces), ie,
first box (**S**): - beta S I
second box (**I**): beta S I - gamma I
third box (**R**): gamma I
8. Write the initial system conditions under the **Initial Condition**, ie,
first box (**S(0)**): 190
second box (**I(0)**): 10
third box (**R(0)**): 0
9. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 0.5
second box (**gamma**): 0.0278
10. Select the **within** option in the **Vital dynamic** dialog box;
11. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.0143;
12. Write the simulation parameters, ie,
Number of simulation: 10
Time spacing: 0.1
Final time: 100
13. Select the **Random** option in the **Network** dialog box;
14. Write the parameter of the random network, that is, $p = 1$;
15. Click in the **Run simulation** button.

Figure 3b

1. Close the graph plot (graph results);
2. Click in **Show Graph**.

Figure 4

1. Close the Figure 1 (network);
2. Click in **Show Measures**.

Figure 7a

1. Download the “infectious.dat” file² accessing the web address: <https://www.dropbox.com/s/e7utgi2n54asxvr/infectious.dat?dl=0>.
2. Select the **no** option in the **Defined models** dialog box;
3. Select the second option (3 compartments) in the **Number of compartments** dialog box;
4. Write the names of the compartments **S**, **E** and **I** in the three boxes, respectively;
5. Select the fifth option (5 parameters) in the **Number of parameters** dialog box;
6. Write the names of the parameters **beta**, **delta**, **gamma**, **rho** and **q** in the five boxes, respectively;
7. Click in **Writer system**;
8. Write the system of differential equations in the boxes below the **ODE system** (each parameter, variable and sign must be separated by spaces), ie,
first box (\dot{S}): - beta S I
second box (\dot{E}): q beta S I - delta I E - gamma E + rho I
third box (\dot{I}): (1 - q) beta S I + delta I E + gamma E - rho I
9. Write the initial system conditions under the **Initial Condition**, ie,
first box (**S(0)**): 400
second box (**E(0)**): 0
third box (**I(0)**): 10
10. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 4.3
second box (**delta**): 0.0867
third box (**gamma**): 0.0027
fourth box (**rho**): 0.0083
fifth box (**q**): 0.9
11. Select the **within** option in the **Vital dynamic** dialog box;
12. Write the value of the system’s birth/mortality rate in the μ box parameter, ie, 0.0143;

²This file is part of the data openly available at <http://konect.cc/networks/sociopatterns-infectious/>

13. Write the simulation parameters, ie,
 Number of simulation: 10
 Time spacing: 0.1
 Final time: 500
14. Select the **other** option in the **Network** dialog box;
15. Select the “infectious.dat” file;
16. Click in the **Run simulation** button.

Figure 7b

1. Close the graph plot (graph results);
2. Click in **Show Graph**.

Figure 8

1. Close the Figure 1 (network);
2. Click in **Show Measures**.

Figure 12a

1. Select the **no** option in the **Defined models** dialog box;
2. Select the third option (4 compartments) in the **Number of compartments** dialog box;
3. Write the names of the compartments **S**, **I**, **R** and **V** in the four boxes, respectively;
4. Select the third option (3 parameters) in the **Number of parameters** dialog box;
5. Write the names of the parameters **beta**, **gamma** and **rho** in the three boxes, respectively;
6. Click in **Writer system**;
7. Write the system of differential equations in the boxes below the **ODE system** (each parameter, variable and sign must be separated by spaces), ie,
 first box (\dot{S}): - beta S I - rho S
 second box (\dot{I}): beta S I - gamma I - rho I
 third box (\dot{R}): gamma I - rho E
 fourth box (\dot{V}): rho S + rho I + rho R
8. Write the initial system conditions under the **Initial Condition**, ie,
 first box (**S(0)**): 190
 second box (**I(0)**): 10
 third box (**R(0)**): 0
 fourth box (**V(0)**): 0

9. Write the parameter values under **Values Parameters**, ie,
first box (**beta**): 0.5
second box (**gamma**): 0.0278
third box (**rho**): 0.01287
10. Select the **within** option in the **Vital dynamic** dialog box;
11. Write the value of the system's birth/mortality rate in the μ box parameter, ie, 0.00143;
12. Write the simulation parameters, ie,
Number of simulation: 10
Time spacing: 0.1
Final time: 100
13. Select the **Small-World** option in the **Network** dialog box;
14. Write the parameters of the small-world network, that is, $p = 0.01$ and $n_0 = 20$;
15. Click in the **Run simulation** button.

Figure 12b

1. Close the graph plot (graph results);
2. Click in **Show Graph**.

Figure 13

1. Close the Figure 1 (network);
2. Click in **Show Measures**.