EMULSION: transparent and flexible multiscale stochastic models in epidemiology

Sébastien Picault[1,2*], Yu-Lin Huang[1#a], Vianney Sicard[1], Sandie Arnoux[1], Gaël Beaunée[1] and Pauline Ezanno[1]

[1]BIOEPAR, INRA, Oniris, 44307 Nantes, France

[2]Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France

[#a]Current Address: Univ. Artois, EA 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), Béthune, France

* Corresponding author

E-mail: sebastien.picault@inra.fr

Abstract

Stochastic mechanistic epidemiological models largely contribute to better understand pathogen emergence and spread, and assess control strategies at various scales (from within-host to transnational scale). However, developing realistic models which involve multi-disciplinary knowledge integration faces three major challenges in predictive epidemiology: lack of readability once translated into simulation code, low reproducibility and reusability, and long development time compared to outbreak time scale. We introduce here EMULSION, an artificial intelligence-based software intended to address those issues and help modellers focus on model design rather than programming. EMULSION defines a domain-specific language to make all components of an epidemiological model (structure, processes, parameters...) explicit as a structured text file. This file is readable by scientists from other fields (epidemiologists, biologists, economists), who can contribute to validate or revise assumptions at any stage of model development. It is then automatically processed by EMULSION generic simulation engine, preventing any discrepancy between model description and implementation. The modelling language and simulation architecture both rely on the combination of advanced artificial intelligence methods (knowledge representation and multi-level agent-based simulation), allowing several modelling paradigms (from compartment- to individual-based models) at several scales (up to metapopulation). The flexibility of EMULSION and its capability to support iterative modelling are illustrated here through examples of progressive complexity, including late revisions of core model assumptions. EMULSION is also currently used to model the spread of several diseases in real pathosystems. EMULSION provides a command-line tool for checking models, producing model diagrams, running simulations, and plotting outputs. Written in Python 3, EMULSION runs on Linux, MacOS, and Windows. It is released under Apache-2.0 license. A comprehensive documentation with installation instructions, a tutorial and many examples is available from: `https://sourcesup.renater.fr/emulsion-public`.

Introduction

Understanding and predicting the spread of pathogens at several scales (from individuals to territories) under various scenarios (control measures, climate, etc.) relies on realistic mechanistic models [1–5]. Yet, predictive epidemiology is currently facing key methodological challenges which

could be faced mobilizing Artificial Intelligence (AI) technologies. First, the lack of model readability often hinders the capability of scientists from life sciences to discuss or revise model assumptions, once translated into simulation code, impairing the users' confidence as well as critical mind towards model predictions. Second, epidemiological models highly depend on implementation details, which affects reusability in even slightly different contexts. Third, developing mechanistic models for complex systems is an iterative process, rendering it difficult to get reliable forecasts in a duration relevant w.r.t. on-field needs (e.g. while fighting outbreaks). Fourth, the complexity of realistic models often leads to misinterpretations coming from programming biases [6], thus it is difficult to ensure that model assumptions are correctly implemented in the simulation code. Fifth, model granularity is likely to change throughout the model development process, either to account for increased detail level, or conversely to keep only relevant features at a broader scale. Changes in core model assumptions can also occur late in the modelling process, for instance because a feature first considered negligible must finally be accounted for. This requires a capability to move back and forth between modelling paradigms (stochastic vs. deterministic, compartments vs. individuals) and between scales (individuals, groups, populations, metapopulations) without having to recode large parts of the models.

To address such challenges, the implementation of epidemiological models must first gain transparency, in order to make all assumptions as explicit as possible. Models are a deliberately simplified representation of reality, thus it is crucial to present the underlying assumptions and their implications, to help understand their benefits and limitations and assess their relevance. Once models are translated into simulation code (which is the usual process), scientists without programming skills, involved either in the co-design of the model or in a peer-reviewing process, cannot develop a thorough understanding of the models nor make an informed judgment. Instead, models should be described in a readable form. The ODD protocol (Overview, Design concepts, Details) [7] is a substantial effort to obtain a comprehensive description of model components within a textual template; however, "the lack of real specification" [8] prevents to translate ODD models straight into simulation code. On the other hand, enhancing reusability requires reproducibility (the capability for others to recode a model) and flexibility (the possibility to adapt it). Though necessary, best practices in software engineering [9] cannot settle those issues alone: ad hoc codes can be documented, tested, reproducible, and yet wrong [10].

A first step to reduce development time and errors is to use simulation platforms, which provide high-level built-in features. In general-purpose platforms such as GAMA [11], NetLogo [12] or Repast [13], a substantial programming effort is still required as algorithms involved in epidemiological models are not provided as platform features. Simulation libraries and platforms dedicated to epidemiological issues are rising, e.g. SimInf [14], a R library for data-driven compartment-based models; MicroSim [15], an agent-based platform for several kinds of diseases; or GLEaMviz [16], a metapopulation-oriented platform. To our knowledge, the most advanced approach in terms of diversity of modelling paradigms is Broadwick [17], a Java framework for compartment- and individual-based models with interaction networks, which nevertheless still requires writing large portions of code to derive specific classes and carry out simulations on practical cases.

Artificial intelligence (AI) can help going further, as demonstrated in a promising approach, KENDRICK, which defines a domain-specific language (DSL, [18]) for compartment-based models [19]. This DSL allows to describe models as text files rather than executable code, enforces a clear separation of concerns (infection, demography, etc.) to facilitate model assessment by scientists from several domains, and is used to generate an optimized simulation code, which guarantees that model features are translated without bias into the program.

To the best of our knowledge, none of existing solutions address those methodological challenges simultaneously, the most advanced approaches providing either a flexibility in modelling paradigms at the expense of software development efforts, or an enhanced readability through a DSL limited to a specific modelling paradigm.

This article introduces EMULSION, an AI-based framework which tackles those challenges in an integrated approach. EMULSION intends to help modellers develop mechanistic stochastic models of complex systems in epidemiology at several scales using multiple paradigms, and to facilitate the co-construction and assessment of model components (biological assumptions, model structure, parameters, scenarios, etc.) with epidemiologists or scientists from other relevant fields, throughout model development. This makes our software an outstanding contribution towards reliable, reactive and transparent predictive epidemiology.

Design and implementation

To address the challenges presented above, EMULSION combines two AI methods [20]. The first one is a DSL designed for the description of all components of an epidemiological model, to make them explicit in a human-readable form as a structured text file, so that scientists from different fields can better interact with modellers throughout the modelling process, and discuss, assess or revise model structure, assumptions, parameters at any moment without having to read or write any line of simulation code. The second one is the use of a generic simulation engine, whose core architecture relies upon a multi-level agent-based system [21]. This allows several scales (individuals, groups, populations, metapopulations) and modelling paradigms (compartment- or individual-based models) to be encompassed within a homogeneous software interface, as agents act as wrappers which can be dynamically combined regardless of what they have to compute and the scale at which they operate. To run an experiment, the simulation engine reads the DSL file containing the model description, assembles the agents required for a particular type of model and a specific scale, initializes parameters, functions, processes specified in the DSL file, and make them interact to produce simulation outputs (Fig. 1).

Representing processes: from flow diagrams to state machines

Processes occurring in the pathosystem (infection, demography, migrations…) are a core component of epidemiological models, often described by flow diagrams [22,23] with nodes denoting state variables (amount of individuals in each state), and transitions labelled with rates. Though flow diagrams can be easily derived into ordinary differential equations or into stochastic difference equations, they often mix several concerns in a unique, monolithic representation, involve implicit computational assumptions (e.g. an exponential distribution of durations in states), and cannot capture additional information (e.g. conditions, actions) which have to be introduced later at implementation. EMULSION relies upon a formalism close to flow diagrams but more accurate: finite state machines [24], classically used in computer science.

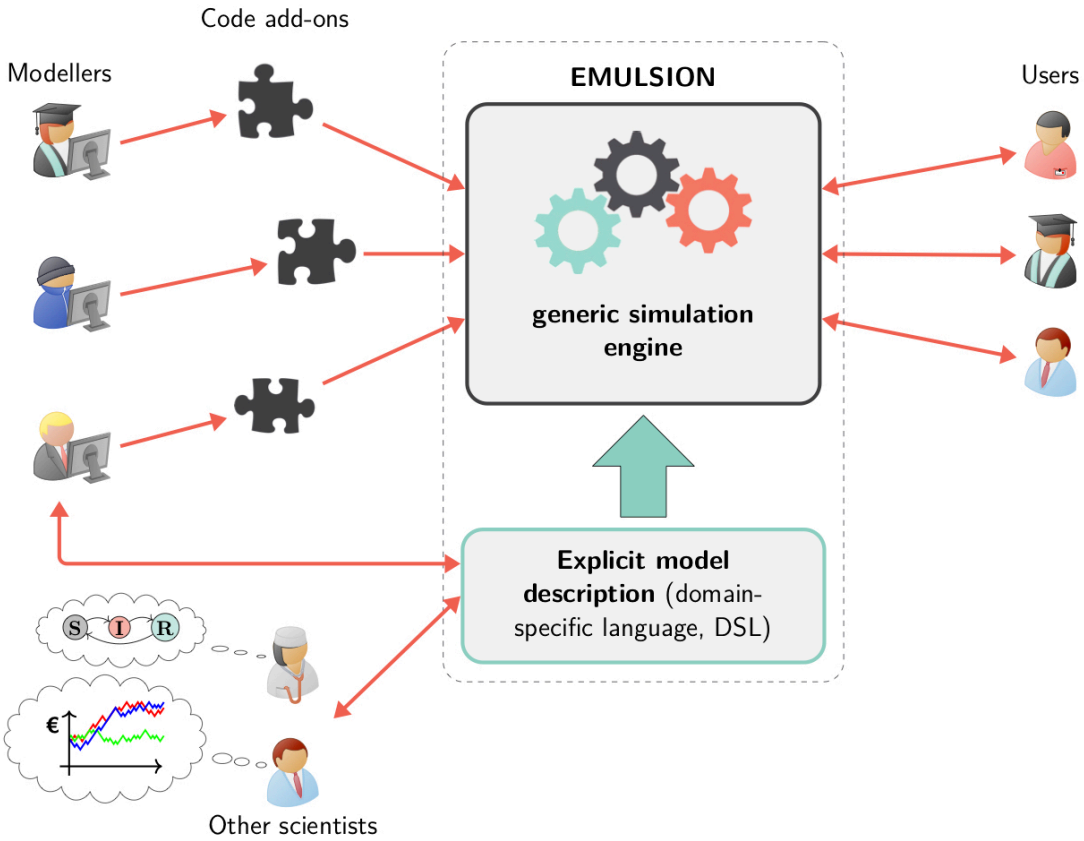

Fig 1. Approach enforced in the EMULSION framework. A generic simulation engine is coupled to a domain-specific modelling language (DSL), reinforcing interactions between modellers and scientists from other fields. Knowledge involved in epidemiological models is kept explicit, understandable and revisable as a structured text file. A few specific software add-ons can be written to complement the simulation engine if needed.

Compared to flow diagrams, state machines describe the evolution of one individual instead of a population, and one state machine represents one single process, so that a complex flow diagram may be split into several simpler state machines. Besides, states can be endowed with additional properties, such as a duration distribution specifying how long an individual is expected to stay in the current state, and actions performed by individuals when entering, being in, or leaving the state. Transitions are labelled with either a rate, a probability or an absolute amount (rates are automatically converted into probabilities). They can also specify: 1) calendar conditions to indicate time periods when transitions are available; 2) escape conditions allowing to free from state duration constraints; 3) individual conditions to filter which agents are allowed to cross the transition; 4) actions performed by individuals crossing the transition, i.e. after leaving their current state and before entering their new one (Fig 2).

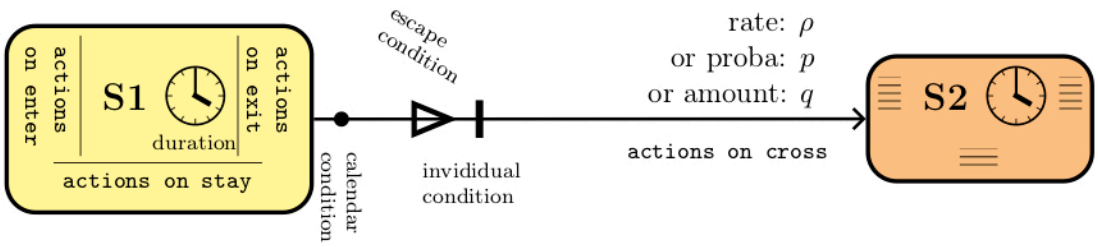

Fig 2. Structure of a transition between two states in state machines. Individuals can be given a duration and actions when entering, staying in, or leaving the state. Transitions feature a rate, probability, or amount, and can be associated with actions performed on crossing, time-dependent ("calendar") conditions, or individual conditions restricting the capability to cross the transition, and escape conditions allowing individuals to leave their state before the nominal duration.

A multi-level agent-based simulation architecture to encompass paradigms and scales

Multi-agent systems are composed of autonomous entities (agents) endowed with a behaviour and interacting in a shared environment. In the last decade, multi-level agent-based systems emerged using agents to explicitly represent intermediary abstraction levels (groups, sub-populations, organizations…) with behaviours of their own, between individuals and the whole system [25–28]. Recent advances in this field [21] led to design patterns, i.e. systematic solutions for recurrent modelling issues. They were used in EMULSION to build the architecture of the generic simulation engine, in which nested agents are in charge of implementing a specific modelling paradigm at a given scale. Agents currently defined in EMULSION allow to implement the main paradigms used in epidemiological models: 1) compartment-based models [29], where state variables represent aggregate amounts of individuals which only differ by few key variables, such as health state or age group; and 2) individual-based models [30] necessary for finer grained representations. Besides, EMULSION provides a hybrid approach which combines the capability of representing detailed information through individuals, with an adaptive grouping of individuals based on their state, to optimize computation (Fig 3). The same approach allows to wrap different scales within agents to build either groups, populations, or metapopulations [31,32] which can handle region-wide models at a moderate computational cost. Scales and paradigms can be chosen independently: hence, a metapopulation at

regional scale can rely at the local scale upon compartment-based, individual-based or hybrid models, depending on the required detail level.

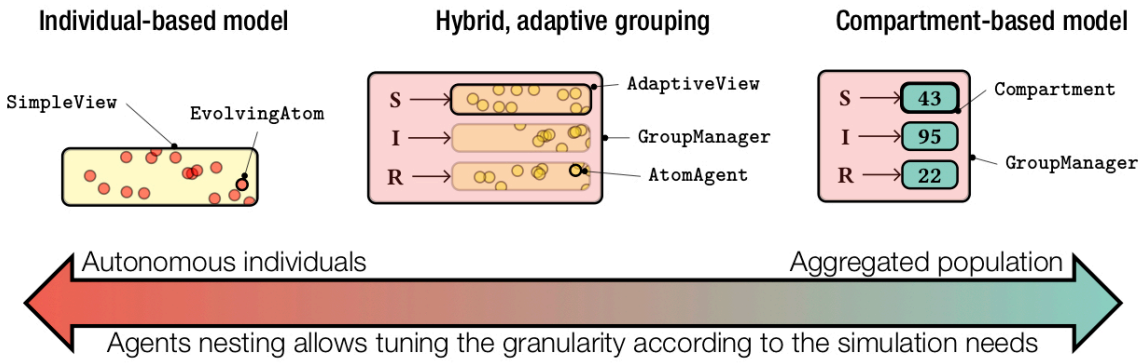

Fig 3. Diversity of modelling paradigms in EMULSION models. EMULSION allows to represent within a same formalism (nested agents) several modelling paradigms, from the finest grained (individuals) to aggregations (compartments), including intermediary representations as a trade-off between computation time and preservation of individual details. The chosen modelling paradigm is associated with the appropriate combinations of agents.

A domain-specific language for epidemiological models

The main benefits of the DSL defined in EMULSION are a modular decomposition of the model, which reduces dependencies between model components and facilitates further extensions, and a high readability by non-modeller scientists. When EMULSION simulation engine reads the DSL, it selects the agent classes corresponding to the chosen paradigm and scales, instantiate agents depending on initial conditions, transforms expressions into Python functions (using SymPy, a library for symbolic computation), builds the state machines, and runs the simulations. Hence, the execution of simulations based on an EMULSION model is univocal, all required information being directly usable (yet human-readable) by the simulation engine.

EMULSION comes with several built-in actions (changing the values of variables, creating new individuals, etc.) which can be used in states and transitions. When the user defines levels, groups, or states, EMULSION automatically provides associated variables (e.g. population counts, individual tests for each state) to make model description as self-sufficient as possible in model files. In very specific

cases, processes or actions may require the definition of small code add-ons, as demonstrated below for data-driven events in a metapopulation. But in the majority of situations, no additional code is required.

An EMULSION model is a text file based on the YAML format [33], which is mainly based on nested lists or key-value pairs, ending with strings, numbers or booleans (yes/no). Each model is composed of several first-level keys (called "sections" in what follows) to provide information by topic: model name, modelling paradigms and scales, processes occurring at each level, structure of state machines, list of parameters and expressions, individual actions, initial conditions, outputs. Section and subsection names are chosen to be as self-evident as possible, and a substantial part of the documentation is dedicated to explaining the syntax of the DSL in detail, with comprehensive examples and tutorials.

EMULSION models can be built by adapting existing examples (changing only a few sections) rather than starting from scratch. Especially, the main part of a model consists in section "state_machines", which defines the state machines underlying the processes occurring in the pathosystem, and section "parameters", which provides a comprehensive, commented list of parameters, distributions and functions used in the calculations. Almost all items have to be accompanied by a textual description of their meaning and role. A full example of a SIR model implemented with hybrid modelling paradigm is provided in S1 Text. Also, a comparison of modelling paradigms (compartment, hybrid and individual-based) on two SIR models (one with constant population, one with birth and death processes) with the corresponding simulation outcomes and state machine diagrams, is detailed in S1 Appendix.

Results

Incremental design of complex epidemiological models

EMULSION enables to build non-trivial models incrementally. We illustrate this below with an example from within-population to between-population scales, and demonstrate how easy it is to operate late changes in core model assumptions.

First, a classical SIR model is developed (S: susceptible, I: infectious, R: resistant), assuming a frequency-dependent force of infection. In this example, we use the hybrid modelling paradigm, where individuals are grouped according to similar health states. Transitions (infection and recovery) are driven by a state machine ("health_state") which is associated with a way of grouping individuals (all

individuals of similar health states are grouped). The corresponding YAML file with full description is provided in S1 Text.

Second, we introduce demography and two age groups: juveniles (J) and adults (A). We assume that juveniles become adults at rate "maturation", that adults produce juveniles at rate "birth" and that both die at rate "mortality". An additional process is required in the model, hence a new state machine (called "age_group"), also associated to a grouping. The differences introduced compared to the first case are highlighted in S2 Text.

Third, the population model is upscaled at the metapopulation scale, assuming that pathogens spread among populations due to a contact network, which is described by observed data. This part involves two code add-ons: one for reading a CSV file which gives the dates, age groups, quantity, source and destination for each move, another to perform the relevant migrations at each time step. The differences introduced in the previous model are highlighted in S3 Text, and Python code add-ons required to handle data are provided in S4 Text.

After being used to better understand pathogens spread in a population, epidemiological models are often extended to assess control strategies. Let now assume that the user wants to assess the effect of targeted vaccination, e.g. of juveniles in spring. This obviously involves substantial changes in the infection process, to incorporate a vaccinated state and associated transitions. The modularity of EMULSION allows modellers to change the "health_state" state machine without any impact on the metapopulation structure. First, a new health state "vaccinated" (V) must be added. A calendar is introduced in the "time_info" section to define when "spring" occurs. Then, the transition from "S" to "V" has to specify that it can only be used during spring ("when: spring") and that it concerns only juveniles ("cond: is_J"). The differences introduced in the previous model are highlighted in S5 Text.

Application to real diseases

EMULSION is currently used to understand pathogen spread within structured populations and metapopulations, and to assess control strategies. Being developed at INRA (French National Institute for Agricultural Research), real disease applications mainly come from veterinary epidemiology, but human and plant diseases could also be addressed.

A first study was conducted on Q fever, a zoonosis caused by the bacteria *Coxiella burnetii* and affecting mainly domestic ruminants. To better understand the interplay between within- and between-

herd levels in bovine dairy herds, and identify the contributions of transmission pathways at regional scale, two existing models (respectively a within- [34] and a between-herd [35] model) were re-implemented using EMULSION. Assumptions of the within-herd model were simplified, keeping relevant assumptions and removing those less crucial in the perspective of the between-herd dynamics [36], which was easily tackled due to EMULSION modularity.

EMULSION was also used to assess strategies for the control of Bovine Respiratory Diseases in young bulls at fattening. In a perspective to reduce antibiotic use, metaphylaxis (collective treatment triggered after first cases are detected) was compared to early detection methods, based on sensors measuring hyperthermia durations. Three tightly coupled processes were modelled: infection, hyperthermia, and treatment, and scenarios combining metaphylaxis policies with detection methods were assessed w.r.t. reduction of antibiotics doses, infection duration, and false positives [37].

Availability and future directions

The EMULSION package documentation, with installation instructions (either from PyPI or by cloning the GIT source repository), tutorial and examples, can be found on SourceSup (a forge dedicated to French public research institutes) at: `https://sourcesup.renater.fr/emulsion-public`. The package comes with 29 examples to illustrate features of EMULSION modelling language, from very simple SIR models declined in compartmental, individual-based, or hybrid approaches, to epidemiological models in structured populations mixing infectious process, demography, seasonality, forcing external variables, and non-exponential distributions of state durations. Examples also demonstrate how to move from within- to between-population models using data-driven movements. Updates will be released regularly based on the integration on new modelling features or code optimizations.

As demonstrated, EMULSION is flexible enough to cover a broad diversity of stochastic and mechanistic epidemiological models at several scales. Current version of the software is fully usable by modellers and currently deployed to develop models for real diseases assuming various propagation pathways (direct contacts, environment contamination, airborne or vector-borne transmission, vertical transmission), demographic and herd management processes, detection methods (observation of clinical signs, sensors, tests), and control measures (vaccination, antibiotics, tests at purchase, selective culling).

Specific features in those epidemiological models, requiring for now dedicated add-ons, will be incorporated to the DSL and to the simulation engine as generic modules if recurrent enough among models, to help encompass a broad range of mechanisms within EMULSION.

We identified three priorities for future work to be carried out in the short term. First, we aim at better automating data integration. Models rely on observed commercial, meteorological, and epidemiological data, either to define parameter values or to confront model predictions with observations. The DSL should thus allow to specify how to inform parameter values from data and how to define data-driven processes. Second, we will address spatially explicit models. For now, discrete metapopulations are possible, relevant for network contact representation but less for proximity contacts (airborne transmission, neighbourhood contacts, vector-borne transmission, etc.). Third, we plan to allow a model to combine several modelling paradigms, e.g. using for vector-borne disease compartments for vectors and individuals for hosts. In the longer term, we plan to use the DSL for optimized code generation (e.g. in C/C++). The choice of Python for implementing EMULSION simulation engine was well adapted to facilitate the modular and incremental development of the software, and its portability on various operating systems and environments. Genericity was also achieved through introspection capabilities of the language, but which came with an additional computational cost. This constitutes a limit of the software when tackling multi-scale models with numerous components (e.g. large number of interacting populations), models which analysis is based mainly on massive simulations. Such models written using EMULSION's DSL then would benefit from being automatically translated into simulation codes generated from templates, allowing for higher computational performances.

The generic modelling approach we promote in EMULSION both has advantages and drawbacks. The gathering of various model features within a homogeneous software tool, which can be picked up and combined according to the modeller's needs with little or even no additional code, reduces development time, ensures that model assumptions and processes are well implemented, and allows optimized code generation as a mid-term perspective. As a common software limitation, potential bugs in the generic engine could impact all models developed with EMULSION: but its diffusion as an open-source software among a broad community of users is the best way to get feedback on potential malfunctions, identify their origin and correct them rapidly.

To our knowledge, EMULSION is currently the only software which addresses all major computing issues in predictive epidemiology. It helps modellers focus on model assumptions and structure rather than coding, ensures more reliable models, provides classical modelling paradigms at multiple scales, and makes models explicit and readable, so that they can be assessed by other scientists and shared in reproducible publications. As a result, the whole model design process is accelerated, which substantially enhances the reactiveness of epidemiologists against emerging outbreaks.

Supporting information

S1 Text. EMULSION model file for a simple SIR model with hybrid modelling paradigm. This file ("hybrid_SIR.yaml") contains all model sections required to describe a classical stochastic SIR model. The hybrid modelling paradigm mixes an individual-based approach (individuals are created in the model) and the compartment-based approach (individuals are grouped automatically depending on their similarities), which renders the hybrid model performance almost as high as for a compartment-based model while enabling individual features.

S1 Appendix. Comparison of simple SIR models (with constant population and with birth/death processes) in EMULSION with compartment-based, hybrid and individual-based modelling paradigms. For each model, simulation outcomes for 500 stochastic repetitions are shown and compared with the deterministic version. We also provide an indication on calculation time for each model and paradigm. A state machine diagram describes the states and transitions involved in the model. Finally, differences between respective EMULSION files are presented side-by-side and highlighted.

S2 Text. Changes to introduce age groups and demography in the hybrid SIR model. Differences between original and modified models (resp. "hybrid_SIR.yaml" and "hybrid_SIR_ages.yaml") are presented side-by-side and highlighted (in bold; red: parts removed, green: parts added; yellow: parts modified; dashes: similar parts not shown).

S3 Text. Changes to upscale the previous SIR model with demography to the metapopulation level, with data-driven migrations. Differences between original and modified models (resp. "hybrid_SIR_ages.yaml" and "hybrid_SIR_ages _metapop.yaml") are presented side-by-side and highlighted (in bold; red: parts removed, green: parts added; yellow: parts modified; dashes: similar parts not shown).

S4 Text. Python code for add-ons associated with the metapopulation. This file ("metapop_movements.py") is required to connect the model described in the YAML file to data used on trade movements. It is composed of two functions: one for reading the CSV file, the other for moving individuals from one population to another according to data.

S5 Text. Changes to previous model (metapopulation) to introduce vaccination for a targeted group and a specific period of the year. Differences between original and modified models (resp. "hybrid_SIR_ages _metapop.yaml" and "hybrid_VSIR_ages_metapop.yaml") are presented side-by-side and highlighted (in bold; red: parts removed, green: parts added; yellow: parts modified; dashes: similar parts not shown).

S6 Text. Installation procedure, tests and help page for EMULSION. Please refer to EMULSION documentation for the latest version: `https://sourcesup.renater.fr/emulsion-public`

S1 File. Zip file containing all example files mentioned in the software paper. File "README.md" provides description and usage for each of them.

S2 File. Zip file containing EMULSION source files. This zip file is a clone of EMULSION public Git repository (hosted at `https://git.renater.fr/emulsion-public.git`). To install from this file, follow Git-based installation instructions from EMULSION documentation page: `https://sourcesup.renater.fr/emulsion-public/pages/Install.html`

Acknowledgments

This work was carried out with the financial support of the French Research Agency (ANR), Program Investments for the Future, project ANR-10-BINF-07 (MIHMES), and the European Union through the European fund for the regional development (FEDER) of Pays-de-la-Loire. It was also funded by the French Research Agency (ANR) through project CADENCE (ANR-16-CE32-0007-01), and the Animal Health Division of the French National Institute for Agricultural Resarch (INRA).

We are also very grateful to Thibaut Morel-Journel for his careful reading and relevant comments.

References

1. Ferguson NM, Cummings DAT, Cauchemez S, Fraser C, Riley S, Meeyai A, et al. Strategies for containing an emerging influenza pandemic in Southeast Asia. Nature. 2005;437: 209–214. doi:10.1038/nature04017

2. Halloran ME, Ferguson NM, Eubank S, Longini IM, Cummings DAT, Lewis B, et al. Modeling targeted layered containment of an influenza pandemic in the United States. Proc Natl Acad Sci. 2008;105: 4639–4644. doi:10.1073/pnas.0706849105
3. Keeling M. The implications of network structure for epidemic dynamics. Theor Popul Biol. 2005;67: 1–8. doi:10.1016/j.tpb.2004.08.002
4. Arino J, van den Driessche P. Disease spread in metapopulations. In: Brunner H, Zhao X-Q, Zou X, editors. Nonlinear Dynamics and Evolution Equations. American Mathematical Society; 2006. pp. 1–12.
5. Marshall BDL, Galea S. Formalizing the Role of Agent-Based Modeling in Causal Inference and Epidemiology. Am J Epidemiol. 2015;181: 92–99. doi:10.1093/aje/kwu274
6. Peng RD. Reproducible Research in Computational Science. Science. 2011;334: 1226–1227. doi:10.1126/science.1213847
7. Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, Giske J, et al. A standard protocol for describing individual-based and agent-based models. Ecol Model. 2006;198: 115–126. doi:10.1016/j.ecolmodel.2006.04.023
8. Amouroux E, Gaudou B, Desvaux S, Drogoul A. O.D.D.: A Promising but Incomplete Formalism for Individual-Based Model Specification. 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF). Hanoi, Vietnam: IEEE; 2010. pp. 1–4. doi:10.1109/RIVF.2010.5633421
9. Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten Simple Rules for Reproducible Computational Research. Bourne PE, editor. PLoS Comput Biol. 2013;9: e1003285. doi:10.1371/journal.pcbi.1003285
10. Leek JT, Peng RD. Opinion: Reproducible research can still be wrong: Adopting a prevention approach: Fig. 1. Proc Natl Acad Sci. 2015;112: 1645–1646. doi:10.1073/pnas.1421412111
11. Amouroux E, Desvaux S, Drogoul A. Towards Virtual Epidemiology: An Agent-Based Approach to the Modeling of H5N1 Propagation and Persistence in North-Vietnam. In: Bui TD, Ho TV, Ha QT, editors. 11th Pacific Rim Int Conf on Multi-Agents (PRIMA). Springer; 2008. pp. 26–33. doi:10.1007/978-3-540-89674-6_6
12. Robins J, Bogen S, Francis A, Westhoek A, Kanarek A, Lenhart S, et al. Agent-based model for Johne's disease dynamics in a dairy herd. Vet Res. 2015;46. doi:10.1186/s13567-015-0195-y
13. Collier N, Ozik J, Macal CM. Large-Scale Agent-Based Modeling with Repast HPC: A Case Study in Parallelizing an Agent-Based Model. Parallel Processing Workshops (Euro-Par). Springer Nature; 2015. pp. 454–465. doi:10.1007/978-3-319-27308-2_37
14. Widgren S, Bauer P, Eriksson R, Engblom S. SimInf: An R package for Data-driven Stochastic Disease Spread Simulations. ArXiv160501421 Q-Bio Stat. 2016; Available: http://arxiv.org/abs/1605.01421
15. Cakici B, Boman M. A workflow for software development within computational epidemiology. J Comput Sci. 2011;2: 216–222. doi:10.1016/j.jocs.2011.05.004
16. Broeck WV den, Gioannini C, Gonçalves B, Quaggiotto M, Colizza V, Vespignani A. The GLEaMviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale. BMC Infect Dis. 2011;11. doi:10.1186/1471-2334-11-37
17. O'Hare A, Lycett SJ, Doherty T, M. Salvador LC, Kao RR. Broadwick: a framework for computational epidemiology. BMC Bioinformatics. 2016;17. doi:10.1186/s12859-016-0903-2
18. Fowler M, Parsons R. Domain-specific languages. Upper Saddle River, NJ: Addison-Wesley; 2011.

19. Bui T-M-A, Stinckwich S, Ziane M, Roche B, Ho TV. KENDRICK: A Domain Specific Language and platform for mathematical epidemiological modelling. The 2015 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future. IEEE; 2015. pp. 132–137. doi:10.1109/RIVF.2015.7049888
20. Picault S, Huang Y-L, Sicard V, Ezanno P. Enhancing Sustainability of Complex Epidemiological Models through a Generic Multilevel Agent-based Approach. In: Sierra C, editor. Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'2017). Melbourne, Australia: AAAI; 2017. doi:10.24963/ijcai.2017/53
21. Mathieu P, Morvan G, Picault S. Multi-level agent-based simulations: Four design patterns. Simul Model Pract Theory. 2018;83: 51–64. doi:10.1016/j.simpat.2017.12.015
22. Keeling MJ, Rohani P. Modeling Infectious Diseases in Humans and Animals. Princeton University Press; 2008.
23. Diekmann O, Heesterbeek H, Britton T. Mathematical Tools for Understanding Infectious Disease Dynamics. 2012; doi:10.1515/9781400845620
24. Booth TL. Sequential Machines and Automata Theory. 1st ed. New York: John Wiley and Sons; 1967.
25. Picault S, Mathieu P. An Interaction-Oriented Model for Multi-Scale Simulation. In: Walsh T, editor. Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'2011). AAAI; 2011. pp. 332–337. Available: https://hal.archives-ouvertes.fr/hal-00826401
26. Morvan G, Veremme A, Dupont D. IRM4MLS: The Influence Reaction Model for Multi-Level Simulation. Multi-Agent-Based Simulation XI. Springer; 2011. pp. 16–27. doi:10.1007/978-3-642-18345-4_2
27. Camus B, Bourjot C, Chevrier V. Multi-level modeling as a society of interacting models. In: Yilmaz L, Ören TI, Madey G, Sierhuis M, Zhang Y, editors. Agent-Directed Simulation Symposium (in SpringSim). SCS/ACM; 2013. Available: http://dl.acm.org/citation.cfm?id=2499595
28. Huraux T, Sabouret N, Haradji Y. A Multi-level Model for Multi-agent based Simulation: Proceedings of the 6th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and and Technology Publications; 2014. pp. 139–146. doi:10.5220/0004814501390146
29. Hethcote HW. The Mathematics of Infectious Diseases. SIAM Rev. 2000;42: 599–653. doi:10.1137/S0036144500371907
30. Railsback SF, Grimm V. Agent-Based and Individual-Based Modelling: A Practical Introduction. Princeton University Press; 2011.
31. Gilpin M, Hanski I, editors. Metapopulation Dynamics: Empirical and Theoretical Investigations [Internet]. Elsevier BV; 1991. doi:10.1016/b978-0-12-284120-0.50003-6
32. Grenfell B, Harwood J. (Meta)population dynamics of infectious diseases. Trends Ecol Evol. 1997;12: 395–399. doi:http://dx.doi.org/10.1016/S0169-5347(97)01174-9
33. Ben-Kiki O, Evans C, döt Net I. YAML Ain't Markup Language (YAML™) version 1.2 [Internet]. 2009 Sep. Available: https://yaml.org/spec/
34. Courcoul A, Monod H, Nielen M, Klinkenberg D, Hogerwerf L, Beaudeau F, et al. Modelling the effect of heterogeneity of shedding on the within herd *Coxiella burnetii* spread and identification of key parameters by sensitivity analysis. J Theor Biol. 2011;284: 130–141. doi:10.1016/j.jtbi.2011.06.017
35. Pandit P, Hoch T, Ezanno P, Beaudeau F, Vergu E. Spread of *Coxiella burnetii* between dairy cattle herds in an enzootic region: modelling contributions of airborne transmission and trade. Vet Res. 2016;47. doi:10.1186/s13567-016-0330-4

36. Picault S, Huang Y-L, Sicard V, Beaudeau F, Ezanno P. A Multi-Level Multi-Agent Simulation Framework in Animal Epidemiology. In: Demazeau Y, Davidsson P, Vale Z, Bajo J, editors. Proceedings of the 15th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'2017). Porto: Springer; 2017. pp. 209–221. doi:10.1007/978-3-319-59930-4_17

37. Picault S, Ezanno P, Assié S. Combining early hyperthermia detection with metaphylaxis for reducing antibiotics usage in newly received beef bulls at fattening operations: a simulation-based approach. In: van Schaik G, editor. Proceedings of the Conference of the Society for Veterinary Epidemiology and Preventive Medicine (SVEPM). 2019.