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 models are essential to predictive epidemiology, to better explore biological assumptions and anticipate effects of control measures on pathogen spread. Their development is usually long and iterative, involving multi-disciplinary knowledge integration. However, knowledge often becomes lost in model code, hindering late changes in assumptions and often making models appearing as black boxes to end-users. We introduce here EMULSION, an artificial intelligence-based software intended to 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 metapopulations). 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 (zoonoses such as Q fever or the vector-borne Rift Valley fever; or bovine respiratory diseases, with a focus on detection and treatment protocols). EMULSION provides a command-line tool for checking models, producing model diagrams, running simulations, and plotting outputs. Implemented in Python 3, EMULSION runs on Linux, MacOS, and soon Windows. It is released under Apache-2.0 license. A comprehensive documentation with installation instructions, tutorial and examples is available from: `https://sourcesup.renater.fr/emulsion-public`.

Introduction

Understanding and controlling the emergence and spread of pathogens at several scales (from individuals to territories) indeed relies on realistic models. Yet, predictive epidemiology is currently facing key challenges. 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

end-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 [1], thus it is difficult to ensure that model assumptions are correctly implemented in the simulation code. Last, 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 that they are presented with underlying hypotheses and their implications, to help understand their benefits and limitations and assess their relevance. This implies that models be described in a readable form, rather than translated into simulation code which cannot be assessed without programming skills. The ODD protocol ("Overview, Design concepts, Details") [2] is a substantial effort to obtain a comprehensive description of model components within a textual template; however, it suffers from "the lack of real specification" [3]. 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 [4] cannot settle those issues alone: ad hoc codes can be documented, tested, reproducible, and yet wrong [5].

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 [6], NetLogo [7] or Repast [8], 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 [9], a R library for data-driven compartment-based models; MicroSim [10], an agent-based platform for several kinds of diseases; or GLEaMviz [11], a metapopulation-oriented

platform. To our knowledge, the most advanced approach in terms of diversity of modelling paradigms is Broadwick [12], 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, [13]) for compartment-based models [14]. 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.

This article introduces EMULSION, an artificial intelligence-based framework intended 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.

Design and implementation

To address the challenges presented above, EMULSION combines two AI methods [15]. 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 [16]. This allows several scales (individuals, groups, populations, territories) and modelling paradigms (compartment- or individual-based models, metapopulations) 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 of the scale at which they operate. To run an experiment, the simulation engine reads the DSL file containing the description of a model, 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).

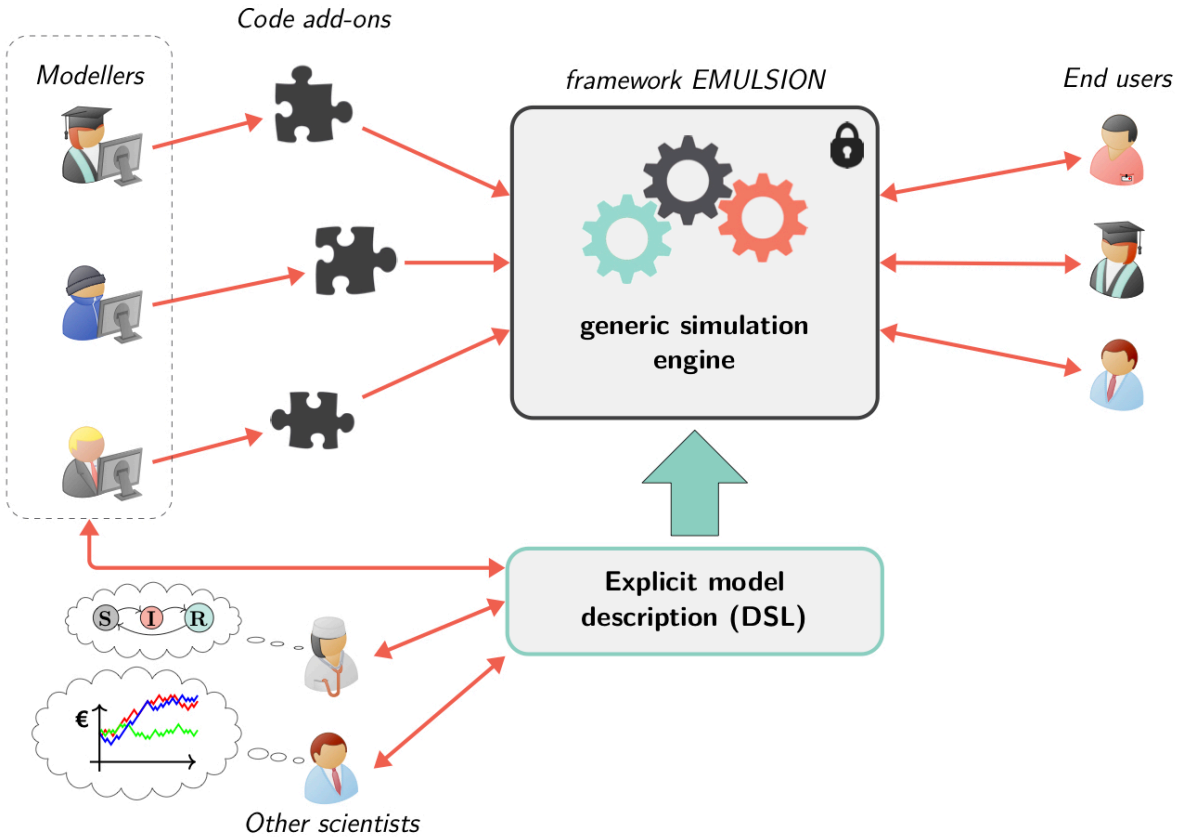

Fig 1. Approach enforced in the EMULSION framework. A generic simulation engine is coupled to a domain-specific modelling language (DSL), reinforcing interactions between modellers and scientists from other fields. Expert knowledge involved in epidemiological models is kept explicit, understandable and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

Representing processes: from flow diagrams to state machines

Processes occurring in the pathosystem (infection, demography, migrations…) are a core component of epidemiological models, which are often described by flow diagrams [17,18], with nodes denoting state variables (the 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 [19], widespread used in computer science. Major differences with flow diagrams are 1) that state machines describe the evolution of one individual instead of a population; and 2) that 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 1) a duration distribution specifying how long an individual is expected to stay in the current state, and 2) 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).

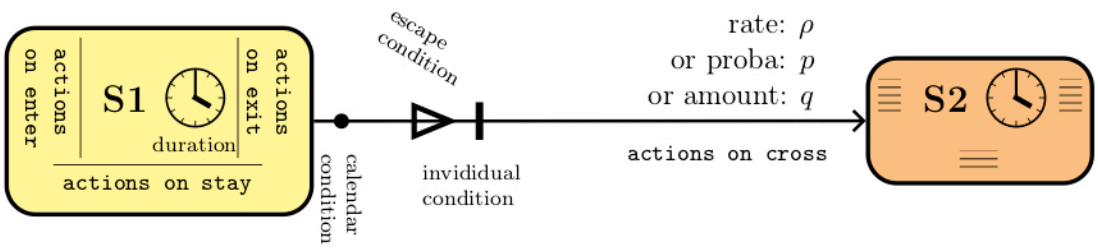

Fig 2. Structure of a transition between two states in state machines. Individuals can be given a duration and actions when entering, staying in or leaving the state. Transitions feature a rate, or probability, or amount, and can be associated with actions performed on crossing, time-dependent ("calendar") 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 behavior 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 [20–23].

Recent advances in this field [16] 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 three main paradigms used in epidemiological models: 1) compartment-based models [24], where state variables represent aggregate amounts of individuals which only differ by few key variables, such as health state or age group; 2) individual-based models [25] necessary for finer grained representations; 3) metapopulations [26,27] which can handle region-wide models at a moderate computational cost. 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). Hence, a metapopulation at regional scale can rely at the local scale upon compartment-based, individual-based or hybrid models, depending on the required details level .

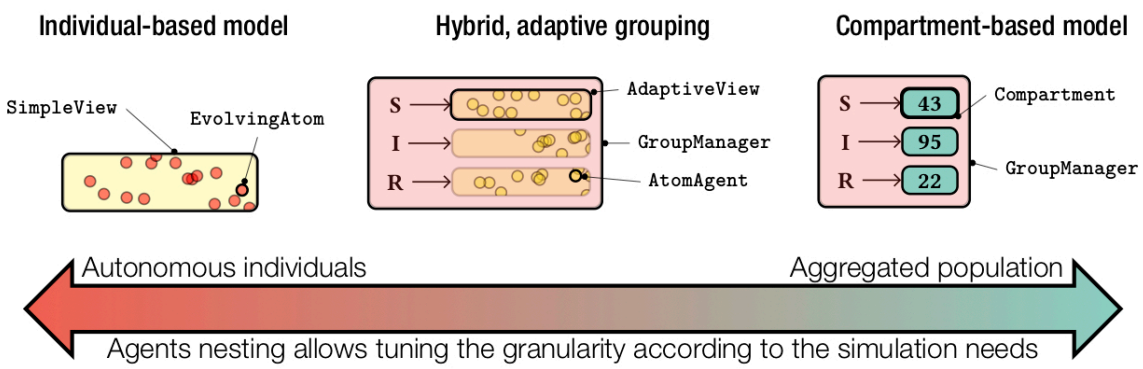

Fig 3. Diversity of modelling paradigms in EMULSION models. EMULSION allows to represent within a same formalism (nested agents) several modelling paradigms, from the finest grained (individuals) to aggregations (compartments), including intermediary representations as a trade-off between computation 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

An EMULSION model is a text file based on the YAML format [28], 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. The main part of a model consists in sections "state_machines", which defines the state machines underlying the processes occurring in the pathosystem, and "parameters", which provides an exhaustive, 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.

The main benefits of the DSL are a modular decomposition of the model, which reduces dependencies between model components and facilitates further extensions, and a high readability by non-modellers 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 YAML 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 necessary.

Results

Incremental design of complex epidemiological models

Here we provide an example to demonstrate how to build a non-trivial model incrementally, from within-population to between-population scales, and finally with late changes in core model assumptions.

First, we start with a classical SIR model, assuming that infection process follows three states: susceptible (S), infectious (I) and resistant (R). In this example, we use the hybrid modelling paradigm, where individuals are grouped according to similar health states. We assume that individuals move from

S to I according to a frequency-dependent force of infection (i.e. at a rate proportional to the proportion of infectious individuals in the whole population), and that I individuals become R at rate "recovery". These transitions are driven by a state machine ("health_state") which is associated with a way of grouping individuals (all individuals with the same health states are grouped together). The corresponding YAML file with full description is provided in S1 Text.

In a second step, 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". This new assumption actually involves an additional process in the model, hence a new state machine (called here "age_group"). Like "health_state", state machine "age_group" is associated to a grouping. The differences to introduce in the original model are highlighted in S2 Text.

Third, we move the model at the metapopulation scale. Assuming that pathogen spread from one population to another results based on a contact network, we want to replay historical recorded data in the model. This part involves two additional Python functions, one for reading a CSV file which gives the dates, age groups, quantity, source and destination for each move, and the other to perform the relevant migrations at each time step. The differences to introduce in the previous model are highlighted in S3 Text, and the Python code add-on required to handle data is provided in S4 Text.

Finally, we want to test a control measure, e.g. the vaccination of juvenile individuals during spring. This obviously involves a substantial change of the infection process, since the initial SIR model now has to incorporate vaccination. Fortunately, the modularity of EMULSION allows 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 "I" 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 to introduce in the previous model are highlighted in S5 Text.

Application to real diseases

EMULSION is currently used to understand the spread of pathogens within populations or metapopulations, and to compare the cost-efficiency of control strategies. Being developed at INRA (the French National Institute for Agricultural Research), such real diseases applications come from veterinary epidemiology for now.

A first study was conducted on Q fever, a zoonosis caused by 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 ad hoc models (respectively a within- [29] and a between-herd [30] model) were reimplemented 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 [31]. Then, it was extended to the regional between-herd level, where assumptions regarding airborne transmission were compared. Finally, the within-herd model was revised with alternative hypotheses in the infection process, to better reproduce the observed annual herd incidence at the between-herd level while making plausible assumptions about host infection processes. The relative roles of trade and airborne transmission in regional pathogen spread were reassessed under these new modelling assumptions [32].

EMULSION was also used to assess strategies for the control of Bovine Respiratory Diseases in young bulls at fattening. In an antibioreduction perspective, 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 [33].

EMULSION is currently used in ongoing research on a zoonotic vector-borne disease, the Rift Valley fever, which involves several host and vector species with mobility data, on the bovine viral diarrhea, to account for the diversity of herd management in dairy and beef herds at regional scale, and on the bovine brucellosis, to assess the capability of epidemiological surveillance systems to detect the re-emergence of the disease and evaluate associated economic costs.

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 theoretical SIR models declined in compartmental, individual-based or

hybrid approaches, to structured populations mixing infectious process, demography, seasonality and non-exponential distributions of durations in states. Examples also demonstrate how to move from a within-population to a between-population model, using for instance metapopulations of cattle with data-driven commercial movements. Updates will be released regularly based on the integration on new modelling features or code optimizations.

We have shown the flexibility of this framework and its capability to integrate a broad diversity of epidemiological stochastic mechanistic models at several scales. EMULSION is currently used to understand the spread of several real diseases and asses control measures, through models which involve different propagation pathways (direct contacts, environment contamination, airborne or vector-borne transmission, vertical transmission), different demographic or herd management processes, and different detection methods (observation of clinical signs, sensors, tests) or control measures (vaccination, antibiotics, tests at purchase, selective culling). They also rely on observed commercial, meteorological, or epidemiological data, either to define parameter values or to confront model outcomes with observations. Recurrent features identified in such epidemiological models will be incorporated to the DSL and to the simulation engine as generic modules to help encompass a broader range of mechanisms within EMULSION.

We already identified three priorities for new features in EMULSION in the short term. First, we aim at a better automation of data integration: the DSL should allow to specify how to inform some parameter values from data, and to define data-driven processes. Second, we have to incorporate methods for facilitating the definition of spatially explicit models. Third, we plan to allow a model to combine several modelling paradigms, e.g. in a vector-borne disease, using compartments for vectors and individuals for hosts.

We also recently initiated a benchmark with three other research teams involved in the design of epidemiological modelling software (Broadwick [12], SimInf [34] and KENDRICK [14]), to identify the specific audience and added-value of each approach, benefit from our mutual experience, and help modellers choose the most appropriate approach depending on the purpose and conditions of their study. This could lead in the long-term to a shared, comprehensive modelling language for epidemiology that would facilitate model comparison and cross-platform interoperability.

Supporting information

S1 Text. EMULSION model 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).

S2 Text. Changes to introduce age groups and demography in the previous SIR model. Differences between original and modified models (resp. "hybrid_sir.yaml" and "hybrid_sir_demo.yaml") are presented side-by-side and highlighted (in bold; red: parts removed, green: parts added).

S3 Text. Changes to change the scale previous SIR model with demography to the metapopulation level, with data-driven migrations. Differences between original and modified models (resp. "hybrid_sir_demo.yaml" and "hybrid_sir_demo_metapop.yaml") are presented side-by-side and highlighted (in bold; red: parts removed, green: parts added).

S4 Text. Python code for the add-on associated with the metapopulation. This file is required to connect the model described in the YAML file to the data used for replaying 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.yaml" and "hybrid_sir_demo.yaml") are presented side-by-side and highlighted (in bold; red: parts removed, green: parts added).

Acknowledgments

The development of EMULSION was funded by the French Research Agency (ANR) through projects MIHMES (ANR-10-BINF-07) and CADENCE (ANR-16-CE32-0007-01), the European fund for the Regional Development (FEDER) of Pays-de-la-Loire, and the Animal Health Division of the French National Institute for Agricultural Resarch (INRA). This work was also supported by the SANT'Innov project of the For and On Regional Development (PSDR) French research programme, which is funded by INRA, the National Research Institute of Science and Technology for Environment

and Agriculture (IRSTEA), and the French regions of Bretagne, Pays de la Loire, Normandie and Nouvelle Aquitaine.

References

1. Peng RD. Reproducible Research in Computational Science. Science. 2011;334: 1226–1227. doi:10.1126/science.1213847
2. 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
3. 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
4. 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
5. 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
6. 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
7. 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
8. 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
9. 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
10. 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
11. 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
12. 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
13. Fowler M, Parsons R. Domain-specific languages. Upper Saddle River, NJ: Addison-Wesley; 2011.
14. 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
15. 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

16. 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
17. Keeling MJ, Rohani P. Modeling Infectious Diseases in Humans and Animals. Princeton University Press; 2008.
18. Diekmann O, Heesterbeek H, Britton T. Mathematical Tools for Understanding Infectious Disease Dynamics. 2012; doi:10.1515/9781400845620
19. Booth TL. Sequential Machines and Automata Theory. 1st ed. New York: John Wiley and Sons; 1967.
20. 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
21. 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
22. 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
23. 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
24. Hethcote HW. The Mathematics of Infectious Diseases. SIAM Rev. 2000;42: 599–653. doi:10.1137/S0036144500371907
25. Railsback SF, Grimm V. Agent-Based and Individual-Based Modelling: A Practical Introduction. Princeton University Press; 2011.
26. 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
27. 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
28. 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/
29. 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
30. 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
31. 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
32. Picault S, Huang Y-L, Sicard V, Hoch T, Vergu E, Beaudeau F, et al. A generic multi-level stochastic modelling framework in computational epidemiology. bioRxiv. 2018; 491605. doi:10.1101/491605
33. 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.

34. Bauer P, Engblom S, Widgren S. Fast event-based epidemiological simulations on national scales. Int J High Perform Comput Appl. 2016;30: 438–453. doi:10.1177/1094342016635723