

1 individual: An R package for individual based
2 epidemiological models

3 Giovanni D. Charles¹ and Sean L. Wu²

4 **1** MRC Centre for Global Infectious Disease Analysis, Abdul Latif Jameel Institute for Disease and
5 Emergency Analytics (J-IDEA), Imperial College London, London, UK. **2** Division of Epidemiology
6 and Biostatistics, School of Public Health, University of California, Berkeley, CA 94720, USA

DOI: [10.21105/joss.0XXXX](https://doi.org/10.21105/joss.0XXXX)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

7 **Summary**

8 individual is an R package which provides users a set of useful primitive elements for spec-
9 ifying individual based models (IBMs), with special attention to models for infectious disease
10 epidemiology. Users build models by specifying variables for each characteristic describing
11 individuals in the simulated population using data structures from the package. individual
12 provides efficient methods for finding subsets of individuals based on these variables, or co-
13 horts. Cohorts can then be targeted for variable updates or scheduled for events. Variable
14 updates queued during a time step are executed at the end of a discrete time step, and the
15 code places no restrictions on how individuals are allowed to interact. These data structures
16 are designed to provide an intuitive way for users to turn their conceptual model of a system
17 into executable code, which is fast and memory efficient.

Editor: [Editor Name](#) ↗

Submitted: 01 January XXXX

Published: 01 January XXXX

License

Authors of papers retain
copyright and release the work
under a Creative Commons
Attribution 4.0 International
License ([CC BY 4.0](#)).

18 **Statement of need**

19 Complex stochastic models are crucial for many tasks in infectious disease epidemiology
20 (Ganyani et al., 2021). Such models can formalize theory, generate synthetic data, evalu-
21 ate counterfactual scenarios, forecast trends, and be used for statistical inference. IBMs are a
22 way to design disaggregated simulation models, usually contrasted with mathematical mod-
23 els, which may model a density or concentration of individuals, or otherwise lump individuals
24 with similar attributes together in some way (Shalizi, 2006). For modeling finite numbers
25 of individuals with significant between-individual heterogeneity and complex dynamics, IBMs
26 are a natural modeling choice when a representation using mathematical models would be
27 cumbersome or impossible (Willem et al., 2017). Even if an aggregated representation were
28 feasible, there are many reasons why an individual-based representation is to be preferred.
29 Synthetic data may need to produce individual level outcomes, which aggregated models by
30 their very nature are unable to provide (Tracy et al., 2018). Other complexities, such as when
31 events occur after a random delay whose distribution differs from a Markovian one, mean even
32 aggregated models will need to store individual completion times, necessitating more complex
33 simulation algorithms and data structures; in such cases it is often more straightforward to
34 adopt an individual-based representation from the start.

35 For practical use, individual-based models need to balance comprehensibility and speed. A
36 fast model whose code is only understood by the author can be difficult to use as a basis for
37 scientific exploration, which necessarily requires the development of multiple models to test
38 hypotheses or explore sensitivity to certain assumptions. On the other hand a clear yet slow
39 model can be practically unusable for tasks such as uncertainty quantification or statistical
40 inference. individual provides a toolkit for users to write models that is general enough
41 to cover nearly all models of practical interest using simple, standardized code which is fast
42 enough to be useful for computationally heavy applications.

43 State of the field

44 There are many software libraries for epidemiological simulation, both in R and other pro-
45 gramming languages. However, based on our review of existing software, no other library
46 exists in the R language which provides users with a set of primitive elements for defining
47 epidemiological models without imposing strong restrictions upon the type of allowed models
48 (e.g.; compartmental, network, etc.), or limiting users to particular mathematical forms for
49 model dynamics.

50 General R Packages

51 Generic individual based simulation packages in R include IBMPopSim (Giorgi et al., 2020),
52 `ibm` (Oliveros-Ramos, 2016) and `ibmcrafter` (Tun, 2016). IBMPopSim provides sophisticated
53 simulation algorithms, but requires users to input C++ code as a string which is then compiled,
54 making it difficult to interface with the existing R ecosystem.

55 Epidemiological R Packages

56 `EpiModel` (Jenness et al., 2018) allows the simulation of highly detailed discrete time models on
57 networks, relying on the `statnet` (Krivitsky et al., 2003-2020) project for classes and algorithms.
58 However due to its focus on directly transmitted diseases, `individual` may be more generic
59 for other epidemiological situations (such as vector borne diseases). In addition it does not
60 offer an interface for compiled code.

61 `hybridModels` (Marques et al., 2020), similarly provides tools for generic IBM modelling in R.
62 However, it is fully implemented in R, limiting the scope for scale and optimisation.

63 Other epidemiology packages in R are more specialised and restrict user models to common
64 forms. These include `SimInf` (Bauer et al., 2016), `nosoi` (Lequime et al., 2020), `SPARSEMODr`
65 (Mihaljevic et al., 2021), `EpilLMCT` (Almutiry & Deardon, 2020) and `EpilLM` (Warriyar et
66 al., 2020).

67 Design Principles

68 Because in many epidemiological models the most important representation of state is a finite
69 set of mutually exclusive values, such as the Susceptible, Infectious, Recovered classes from
70 the well-known SIR model (Allen, 2017), `individual` uses a bitset to store these data. At the
71 R level users can call set operations (union, intersection, complement, symmetric difference,
72 set difference) which are implemented as bitwise operations in the C++ source. This lets
73 users write clear, highly efficient code for updating their model, fully in R.

74 In contrast to other individual based modeling software, where users focus on defining a type for
75 simulated individuals, in `individual` users instead define variables, one for each characteristic.
76 Individual agents are defined by their their position in each bitset giving membership in a
77 variable, or position in a vector of integers or floats. This design is similar to a component
78 system, a design pattern to help decouple complicated types (Nystrom, 2014). Because
79 of this disaggregated representation of state, performing operations to find and schedule
80 cohorts of individuals benefits from fast bitwise operators. This state representation is (to our
81 knowledge), novel for epidemiological simulation. While Rizzi et al. (2018) proposed using a
82 bitset to represent the state of each simulated individual, the population was still stored as
83 types in an array.

84 `individual` uses `Rcpp` (Eddelbuettel & François, 2011) to link to C++ source code, which
85 underlies the data structures exposed to the user. The API for `individual` uses R6 (Chang,

86 2020) classes at the R level which users call to create, update, and query variables. `individual`
87 `individual` also provides a C++ header-only interface which advanced users can link to from
88 their R package. Users can then write their own C++ code or benefit from other packages
89 with a compiled interface, significantly enhancing the extensibility of `individual`'s API, and
90 documentation on linking is available in [vignettes](#).

91 After a user has specified all the variables in their model, dynamics are specified by processes
92 which run each time step, and events which can be scheduled to update specific cohorts in
93 the future. The simulation loop then executes processes, fires events and updates state on
94 each discrete time step.

Individual simulation loop

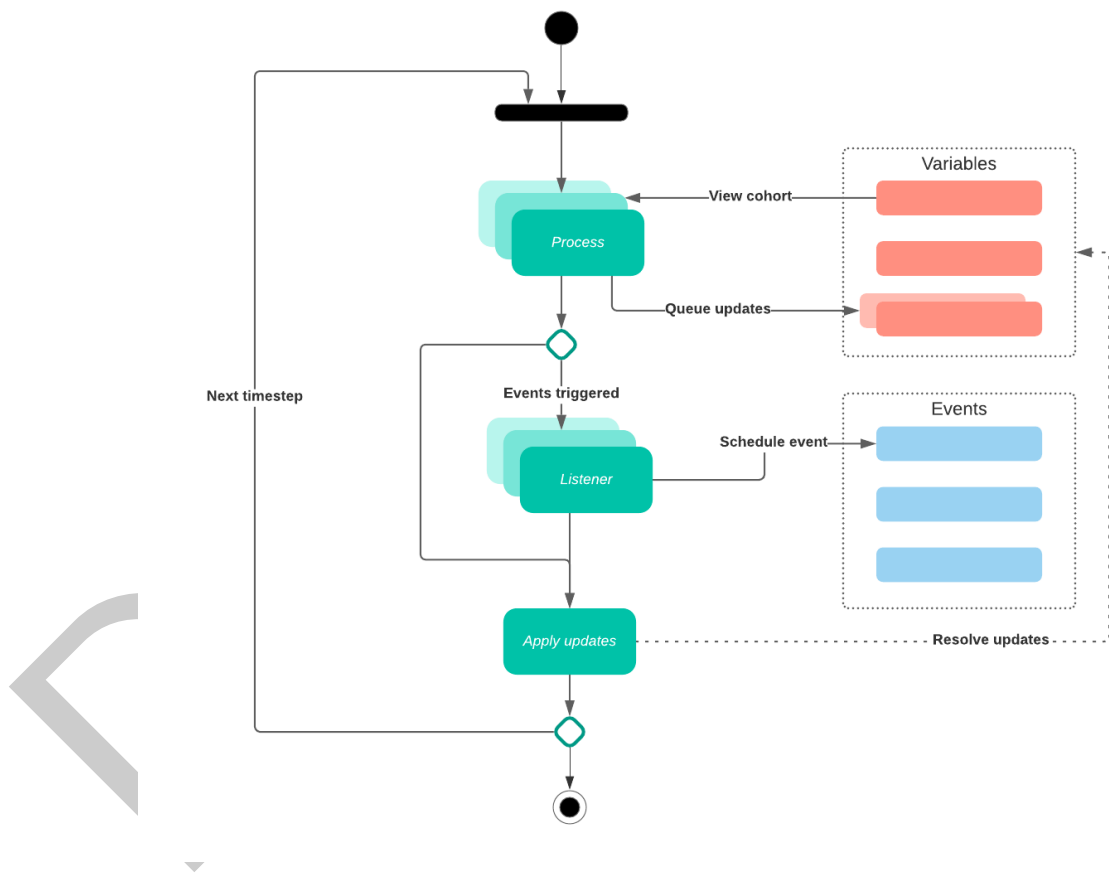


Figure 1: A flow diagram for the simulation loop

95 Licensing and Availability

96 `individual` is licensed under the MIT License, with all source code stored at [GitHub](#). Re-
97 quests, suggestions, and bug reports are encouraged via filing an [issue](#). A general guide on
98 how to contribute to `individual` is available at the [package's website](#).

99 **Acknowledgements**

100 Pete Winskill, Richard Fitzjohn, Oliver Watson

101 **References**

- 102 Allen, L. J. (2017). A primer on stochastic epidemic models: Formulation, numerical sim-
103 ulation, and analysis. *Infectious Disease Modelling*, 2(2), 128–142. [https://doi.org/10.](https://doi.org/10.1016/j.idm.2017.03.001)
104 [1016/j.idm.2017.03.001](https://doi.org/10.1016/j.idm.2017.03.001)
- 105 Almutiry, W., & Deardon, R. (2020). *EpiLMCT: Continuous time distance-based and*
106 *network-based individual level models for epidemics*. [https://CRAN.R-project.org/](https://CRAN.R-project.org/package=EpiLMCT)
107 [package=EpiLMCT](https://CRAN.R-project.org/package=EpiLMCT)
- 108 Bauer, P., Engblom, S., & Widgren, S. (2016). Fast event-based epidemiological simulations
109 on national scales. *The International Journal of High Performance Computing Applica-*
110 *tions*, 30(4), 438–453. <https://doi.org/10.1177/1094342016635723>
- 111 Chang, W. (2020). *R6: Encapsulated classes with reference semantics*. [https://CRAN.](https://CRAN.R-project.org/package=R6)
112 [R-project.org/package=R6](https://CRAN.R-project.org/package=R6)
- 113 Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of*
114 *Statistical Software*, 40(8), 1–18. <https://doi.org/10.18637/jss.v040.i08>
- 115 Ganyani, T., Faes, C., & Hens, N. (2021). Simulation and analysis methods for stochastic
116 compartmental epidemic models. *Annual Review of Statistics and Its Application*, 8(1),
117 69–88. <https://doi.org/10.1146/annurev-statistics-061120-034438>
- 118 Giorgi, D., Kaakai, S., & Lemaire, V. (2020). *IBMPopSim: Individual based model population*
119 *simulation*. <https://CRAN.R-project.org/package=IBMPopSim>
- 120 Jenness, S. M., Goodreau, S. M., & Morris, M. (2018). EpiModel: An r package for mathe-
121 matical modeling of infectious disease over networks. *Journal of Statistical Software*, 84.
122 <https://doi.org/10.18637/jss.v084.i08>
- 123 Krivitsky, P. N., Handcock, M. S., Hunter, D. R., Butts, C. T., Klumb, C., Goodreau, S. M.,
124 & Morris, M. (2003-2020). *Statnet: Software tools for the statistical modeling of network*
125 *data*. Statnet Development Team. <http://statnet.org>
- 126 Lequime, S., Bastide, P., Dellicour, S., Lemey, P., & Baele, G. (2020). Nosoi: A stochas-
127 tic agent-based transmission chain simulation framework in r. *Methods in Ecology and*
128 *Evolution*, 11(8), 1002–1007. <https://doi.org/10.1111/2041-210X.13422>
- 129 Marques, F. S., Grisi-Filho, J. H. H., Silva, J. C. R., Almeida, E. C., & Júnior, J. L. S.
130 (2020). hybridModels: An R package for the stochastic simulation of disease spreading
131 in dynamic networks. *Journal of Statistical Software*, 94(6), 1–32. [https://doi.org/10.](https://doi.org/10.18637/jss.v094.i06)
132 [18637/jss.v094.i06](https://doi.org/10.18637/jss.v094.i06)
- 133 Mihaljevic, J. R., Borkovec, S., Ratnavale, S., Hocking, T. D., Banister, K. E., Eppinger, J.
134 E., Hepp, C. M., & Doerry, E. (2021). SPARSEMODr: Rapid simulations of spatially
135 explicit and stochastic models of infectious disease. *medRxiv*. [https://doi.org/10.1101/](https://doi.org/10.1101/2021.05.13.21256216)
136 [2021.05.13.21256216](https://doi.org/10.1101/2021.05.13.21256216)
- 137 Nystrom, R. (2014). *Game programming patterns*. Genever Benning.
- 138 Oliveros-Ramos, R. (2016). *Ibm: Individual based models in r*. [https://CRAN.R-project.org/](https://CRAN.R-project.org/package=ibm)
139 [package=ibm](https://CRAN.R-project.org/package=ibm)
- 140 Rizzi, R. L., Kaizer, W. L., Rizzi, C. B., Galante, G., & Coelho, F. C. (2018). Modeling
141 direct transmission diseases using parallel bitstring agent-based models. *IEEE Transactions*

- 142 on *Computational Social Systems*, 5(4), 1109–1120. [https://doi.org/10.1109/tcss.2018.](https://doi.org/10.1109/tcss.2018.2871625)
143 [2871625](https://doi.org/10.1109/tcss.2018.2871625)
- 144 Shalizi, C. R. (2006). Methods and techniques of complex systems science: An
145 overview. *Complex Systems Science in Biomedicine*, 33–114. [https://doi.org/10.](https://doi.org/10.1007/978-0-387-33532-2_2)
146 [1007/978-0-387-33532-2_2](https://doi.org/10.1007/978-0-387-33532-2_2)
- 147 Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health:
148 Current applications and future directions. *Annual Review of Public Health*, 39(1), 77–
149 94. <https://doi.org/10.1146/annurev-publhealth-040617-014317>
- 150 Tun, S. T. T. (2016). *Ibmcraft*: Toolkits to develop individual-based models in infectious
151 disease. <https://CRAN.R-project.org/package=ibmcraft>
- 152 Warriyar, V. K. V., Almutiry, W., & Deardon, R. (2020). *EpilLM*: Spatial and network based
153 individual level models for epidemics. <https://CRAN.R-project.org/package=EpilLM>
- 154 Willem, L., Verelst, F., Bilcke, J., Hens, N., & Beutels, P. (2017). Lessons from
155 a decade of individual-based models for infectious disease transmission: A sys-
156 tematic review (2006-2015). *BMC Infectious Diseases*, 17(1), 1–16. [https:](https://doi.org/10.1186/s12879-017-2699-8)
157 [//doi.org/10.1186/s12879-017-2699-8](https://doi.org/10.1186/s12879-017-2699-8)

DRAFT