The Journal of Open Source Software

# EoN (Epidemics on Networks): a fast, flexible Python package for simulation, analytic approximation, and analysis of epidemics on networks

**Joel C. Miller**[1, 2] **and Tony Ting**[2]

**1** La Trobe University, Melbourne, Australia **2** Institute for Disease Modeling, Seattle, Washington, USA

**DOI:** 10.21105/joss.01731

**Software**

- Review
- Repository
- Archive

**Editor:** Lorena Pantano

**Reviewers:**

- @acolum
- @pholme
- @hagberg

**Submitted:** 20 August 2019
**Published:** 20 December 2019



## Summary

EoN (EpidemicsOnNetworks) is a pure-Python package designed to study infectious processes spreading in networks. It rose out of the book *Mathematics of Epidemics on Networks* (Kiss, Miller, & Simon, 2017), and now consists of over 100 user-functions. EoN relies on the networkx package (Hagberg, Schult, & Swart, 2008).

EoN provides methods for

- Susceptible-Infected-Susceptible (SIS) and Susceptible-Infected-Recovered (SIR) disease
  - Stochastic simulation of disease spread in networkx graphs
    - continuous time Markovian
    - continuous time nonMarkovian
    - discrete time
  - Numerical solution of over 20 differential equation models, including
    - individual and pair-based models
    - pairwise models
    - edge-based compartmental models
- Stochastic simulation of a wide range of Simple and Complex contagions
- Visualization and analysis of stochastic simulations

The documentation is maintained at https://epidemicsonnetworks.readthedocs.io/en/latest/ including numerous examples at https://epidemicsonnetworks.readthedocs.io/en/latest/Examples.html. This paper briefly describes a few of EoN's main tools. The online documentation gives full detail, including examples.

## SIR and SIS disease

### Stochastic simulation

The main stochastic simulation tools allow the user to investigate standard SIS and SIR dynamics:

- Markovian SIS and SIR simulations (`fast_SIS`, `Gillespie_SIS`, `fast_SIR`, and `Gillespie_SIR`).

- non-Markovian SIS and SIR simulations (`fast_nonMarkovian_SIS` and `fast_nonMarkovian_SIR`).
- discrete time SIS and SIR simulations where infections last a single time step (`basic_discrete_SIS`, `basic_discrete_SIR`, and `discrete_SIR`).

For both Markovian and non-Markovian methods it is possible for the transition rates to depend on individual or partnership properties.

The continuous-time stochastic simulations have two implementations: a Gillespie implementation (Doob, 1945; Gillespie, 1977) and an event-driven implementation. They have similar speed if the dynamics are Markovian (depending on network and disease parameters either may be faster than the other), but the event-driven implementations also handle non-Markovian dynamics. In earlier versions, the event-driven simulations were consistently faster, leading to the names `fast_SIR` and `fast_SIS`. The Gillespie simulations have been optimized using ideas from (Holme, 2014) and (Cota & Ferreira, 2017).

The algorithms typically handle an SIR epidemic spreading on hundreds of thousands of individuals in well under a minute on a laptop. SIS versions are slower because the number of events that happen is typically much larger.

### Differential Equations Models

EoN also provides tools to numerically solve about 20 differential equations models for SIS or SIR disease spread in networks using SciPy integration tools. The models use different information about the network to predict the number of infected individuals as a function of time. Model derivations and explanations of their simplifying assumptions are in (Kiss et al., 2017).

## Simple and Complex Contagions

Other contagious processes in networks have received attention. Many of these can be classifed as either "simple contagions" or "complex contagions".

In a "simple contagion," an individual `u` may be induced to change status by an interaction with its partner `v`. This status change occurs with the same rate regardless of the statuses of other partners of `u` (there may be a race between partners to determine which transmits first). SIS, SIR, SEIR, and SIRS diseases are special cases of simple contagions.

In "complex contagions," however, the rate at which `u` changes from one status to another may depend on the statuses of others in more complex ways. Two infected individuals may work synergistically to cause a susceptible individual to become infected. This is frequently thought to model social contagions where an individual may only believe something if multiple partners believe it (Centola, Eguíluz, & Macy, 2007).

Simple and complex contagions are currently implemented only in a Gillespie framework, so they require Markovian assumptions. It would typically be feasible to make a bespoke algorithm that runs significantly faster.

### Simple contagions

EoN provides a function `Gillespie_simple_contagion` that allows a user to specify the rules governing an arbitrary simple contagion. The implementation requires the user to separate out two distinct ways in which transitions occur: those that occur spontaneously from an individual's current state and those that are induced by a partner. To help demonstrate, consider an "SEIR" epidemic, where individuals begin susceptible, but when they interact

with infectious partners they may enter an exposed state. They remain in that exposed state for some period of time before transitioning into the infectious state independently of the status of any partner. They remain infectious and eventually transition into the recovered state, again independently of the status of any partner. Here the "E" to "I" and "I" to "R" transitions occur spontaneously given the individual's state, while the "S" to "E" transition is induced by a partner.

### Complex contagions

Complex contagions are implemented through `Gillespie_complex_contagion` for which the user specifies the rules governing a relatively arbitrary complex contagion. There is a constraint that there is no memory: an individual will change from one status to another based on the current statuses of its neighbors, and not based on previous interactions with some neighbors who may have since changed status.

The Gillespie implementation requires a user-defined function that calculates the rate at which `u` will change status given the current system state and another function which chooses its new status. It also needs a function that determines which nodes have their rate change due to `u`'s transition. With these functions defined, the Gillespie algorithm can simulate the complex contagion.

### Visualization & Analysis

By default, simulations return NumPy arrays providing counts of each state.
However, if we set a flag `return_full_data=True`, the simulations return a `Simulation_Investigation` object. This provides access to complete information about the simulation, including the transmission chains.

The `Simulation_Investigation` object can create a snapshot of the network at a given time. By default the visualization includes the time series (e.g., S, I, and R) plotted beside the network snapshot, with flexibility about what (or if) other time series appear. With appropriate additional packages for Matplotlib's animation tools, it can produce animations as well.

## Discussion

EoN provides tools for contagious processes spreading in contact networks, including SIR and SIS diseases and more generally simple and complex contagions. It also provides tools for visualizing stochastic simulation output. Full documentation is available at https://epidemicsonnetworks.readthedocs.io/en/latest/

# Dependencies:

Scipy NumPy networkx Matplotlib

# Related Packages

Several alternative software packages allow for simulation of epidemics on networks. Here we briefly review some of these.

### epydemic

The Python package Epydemic simulates SIS and SIR epidemics in networks. It is built on networkx. It handles both discrete-time or continuous-time Markovian simulations for which it uses a Gillespie-style algorithm. It can handle any process which can be simulated using `EoN.simple_contagion`.

The documentation is available at https://pyepydemic.readthedocs.io/en/latest/

### Graph-tool

The Python package Graph-tool (Peixoto, 2014) serves as a networkx alternative. Its underlying processes are written in C++, so it is often much faster.

Graph-tool has a number of built-in dynamic models, including the SIS, SIR, and SIRS models. The disease models are currently available only in discrete-time versions.

The disease model documentation is available at https://graph-tool.skewed.de/static/doc/dynamics.html.

### EpiModel

The R package EpiModel (Jenness, Goodreau, & Morris, 2018) can handle SI, SIS, and SIR disease spread. It is possible to extend EpiModel to other processes. EpiModel is built around the StatNet package. More details are available at https://www.epimodel.org/

## Funding and Support

The development of EoN has been supported by Global Good and by La Trobe University.

## References

Centola, D., Eguíluz, V. M., & Macy, M. W. (2007). Cascade dynamics of complex propagation. *Physica A: Statistical Mechanics and its Applications*, *374*(1), 449–456. doi:10.1016/j.physa.2006.06.018

Cota, W., & Ferreira, S. C. (2017). Optimized gillespie algorithms for the simulation of markovian epidemic processes on large and heterogeneous networks. *Computer Physics Communications*, *219*, 303–312. doi:10.1016/j.cpc.2017.06.007

Doob, J. L. (1945). Markoff chains–denumerable case. *Transactions of the American Mathematical Society*, *58*(3), 455–473. doi:10.1090/S0002-9947-1945-0013857-4

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, *81*(25), 2340–2361. doi:10.1021/j100540a008

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15). doi:10.25080/issn.2575-9752

Holme, P. (2014). Model versions and fast algorithms for network epidemiology. *arXiv preprint arXiv:1403.1011*.

Jenness, S. M., Goodreau, S. M., & Morris, M. (2018). EpiModel: An R package for mathematical modeling of infectious disease over networks. *Journal of Statistical Software*, *84*(8), 1–47. doi:10.18637/jss.v084.i08

Kiss, I. Z., Miller, J. C., & Simon, P. L. (2017). *Mathematics of epidemics on networks: From exact to approximate models*. IAM. Springer. doi:10.1007/978-3-319-50806-1

Peixoto, T. P. (2014). The graph-tool python library. *figshare*. doi:10.6084/m9.figshare.1164194