epidemik

Release 0.1.3.3

Bruno Gonçalves

Mar 30, 2025

CONTENTS:

1 src **1**
- 1.1 epidemik package 1

2 Indices and tables **11**

Python Module Index **13**

Index **15**

CHAPTER ONE

SRC

1.1 epidemik package

1.1.1 Submodules

1.1.2 epidemik.EpiModel module

`class epidemik.EpiModel.`**`EpiModel`**(*compartments=None*, *seed=None*, *rng=None*)

Bases: `object`

Simple Epidemic Model Implementation

Provides a way to implement and numerically integrate

`R0``()`

Calculate the basic reproductive ratio (R0) for the model.

Uses the Next-Generation matrix approach defined in J. R. Soc Interface 7, 873 (2010).

Returns

The value of the largest eigenvalue of the next generation matrix, or None if calculation fails

Return type

float or None

`add_age_structure`(*matrix*, *population*)

Add age structure to the model using a contact matrix.

Parameters

- **`matrix`** (*`list[list[float]]`*) – Contact matrix between age groups
- **`population`** (*`list[int]`*) – Population size for each age group

Returns

The modified model structure

Return type

list[list]

`add_birth_rate`(*comps=None*, *rate=None*, *fixed=False*, *global_rate=True*, ***rates*)

Add a birth rate to one or more compartments.

Parameters

- **`rate`** (*`float`*) – Birth rate

- **comps** (*list[str], optional*) – List of compartments to which to assign this birth rate. If None, apply to all compartments

Return type
`None`

Returns
None

add_death_rate(*comps=None*, *rate=None*, *fixed=False*, *global_rate=False*, ***rates*)

Add a death rate to one or more compartments.

Parameters

- **rate** (*float*) – Death rate
- **comps** (*list[str], optional*) – List of compartments to which to assign this death rate. If None, apply to all compartments

Return type
`None`

Returns
None

add_interaction(*source*, *target*, *agent*, *rate=None*, *norm=True*, ***rates*)

Add an interaction between two compartments.

This method adds a directed edge from the source compartment to the target compartment in the transition graph, with the specified agent and rate. The rates are passed as keyword arguments and will be added to the model's parameters.

Parameters

- **source** (*str*) – Name of the source compartment
- **target** (*str*) – Name of the target compartment
- **agent** (*str*) – Name of the agent
- **rates** (*dict*) – Named parameters representing the interaction rates

Return type
`None`

Returns
None

add_spontaneous(*source*, *target*, *rate=None*, ***rates*)

Add a spontaneous transition between two compartments.

Parameters

- **source** (*str*) – Name of the source compartment
- **target** (*str*) – Name of the target compartment
- **rates** (*dict*) – Named parameters representing the transition rates

Return type
`None`

Returns
None

`add_vaccination`(*source*, *target*, *start*, *rate*, ***rates*)

Add a vaccination transition between two compartments.

Parameters

- **`source`** (*`str`*) – Name of the source compartment
- **`target`** (*`str`*) – Name of the target compartment
- **`rate`** (*`float`*) – Rate of the vaccination
- **`start`** (*`int`*) – Start time of the vaccination

Return type

`None`

Returns

None

`add_viral_generation`(*source*, *target*, *source_rate=None*, *target_rate=None*, ***rates*)

Add a viral generation transition

Return type

`None`

Parameters: - source: string

Name of the source compartment

- **target: string**
  Name of the target compartment
- **source_rate: float**
  Rate of destruction of infected cells
- **target_rate: float**
  Rate of creation of viral particles

`download_model`(*repo=None*, *load_model=True*)

Download model from offical repository

Return type

`Optional[Self]`

`draw_model`(*ax=None*, *show=True*)

Plot the model structure.

Parameters

- **`ax`** (*`matplotlib.axes._subplots.AxesSubplot, optional`*) – The Axes object to plot to. If None, a new figure is created
- **`show`** (*`bool, default=True`*) – Whether to call plt.show() or not

Return type

`None`

Returns

None

`integrate`(*timesteps*, *t_min=1*, *seasonality=None*, ***kwargs*)

Numerically integrate the epidemic model.

Parameters

- **`timesteps`** (*`int`*) – Number of time steps to integrate
- **`t_min`** (*`int, default=1`*) – Starting time
- **`seasonality`** (*`numpy.ndarray, optional`*) – Array of seasonality values
- **`kwargs`** (*`dict`*) – Initial population of each compartment

Return type
`None`

Returns
None

`list_models()`

List the models available in the official repository

Return type
`List[str]`

`load_model()`

Load a model from a file.

Parameters
`filename` (*`str`*) – Name of the file to load the model from

Returns
The loaded model

Return type
EpiModel

`plot`(*title=None*, *normed=True*, *show=True*, *ax=None*, ***kwargs*)

Convenience function for plotting model results.

Parameters

- **`title`** (*`str, optional`*) – Title of the plot
- **`normed`** (*`bool, default=True`*) – Whether to normalize the values or not
- **`show`** (*`bool, default=True`*) – Whether to call plt.show() or not
- **`ax`** (*`matplotlib.axes._subplots.AxesSubplot, optional`*) – The Axes object to plot to. If None, a new figure is created
- **`kwargs`** (*`dict`*) – Additional arguments to pass to the plot function

Returns
The plot object

Return type
matplotlib.axes._subplots.AxesSubplot

`save_model`(*filename*)

Save the model to a file.

Parameters
`filename` (*`str`*) – Name of the file to save the model to

Return type
`None`

Returns
None

`simulate`(*timesteps*, *t_min=1*, *seasonality=None*, ***kwargs*)

Stochastically simulate the epidemic model.

Parameters

- **`timesteps`** (*`int`*) – Number of time steps to simulate
- **`t_min`** (*`int, default=1`*) – Starting time
- **`seasonality`** (*`numpy.ndarray, optional`*) – Array of seasonal factors
- **`kwargs`** (*`dict`*) – Initial population of each compartment

Return type

`None`

Returns

None

`single_step`(*seasonality=None*, ***kwargs*)

Perform a single simulation step.

Parameters

- **`seasonality`** (*`numpy.ndarray, optional`*) – Array of seasonality values
- **`kwargs`** (*`dict`*) – Initial population of each compartment if simulation hasn't started

Returns

None

1.1.3 epidemik.MetaEpiModel module

`class epidemik.MetaEpiModel.`**`MetaEpiModel`**(*travel_graph*, *populations*, *population='Population'*, *seed=None*)

Bases: `object`

Simple Epidemic Model Implementation

Provides a way to implement and numerically integrate

`R0`()

`add_interaction`(*source*, *target*, *agent*, ***rates*)

Add an interaction between two **compartments_**

Return type

`None`

Parameters: - source: string

Name of the source compartment

- **target: string**
  Name of the target compartment
- **agent: string**
  Name of the agent
- **rate: float**
  Rate of the interaction

Returns: None

`add_spontaneous`(*source*, *target*, ***rates*)

Add a spontaneous transition between two **compartments_**

Parameters: - source: string

Name of the source compartment

- **target: string**
  Name of the target compartment
- **rate: float**
  Rate of the transition

Returns: None

`add_vaccination`(*source*, *target*, *start*, ***rates*)

Add a vaccination transition between two **compartments_**

Return type
`None`

Parameters: - source: string

Name of the source compartment

- **target: string**
  Name of the target compartment
- **rate: float**
  Rate of the vaccination
- **start: int**
  Start time of the vaccination

Returns: None

`draw_model`()

Return type
`None`

`get_state`(*state*)

Return a reference to a state EpiModel object

Return type
`EpiModel`

Parameters: - state: string

Name of the state to return

`integrate`(***kwargs*)

`plot`(*title=None*, *normed=True*, *layout=None*, ***kwargs*)

Return type
`None`

plot_peaks()

Return type
`None`

simulate(*timestamp*, *t_min=1*, *seasonality=None*, *seed_state=None*, *susceptible='S'*, ***kwargs*)

Return type
`None`

1.1.4 epidemik.NetworkEpiModel module

class epidemik.NetworkEpiModel.**NetworkEpiModel**(*network*, *compartments=None*)

Bases: *EpiModel*

R0()

Calculate the basic reproductive ratio (R0) for the model.

Uses the Next-Generation matrix approach defined in J. R. Soc Interface 7, 873 (2010).

Returns
The value of the largest eigenvalue of the next generation matrix, or None if calculation fails

Return type
float or None

add_interaction(*source*, *target*, *agent*, *rate=None*, *rescale=False*, ***rates*)

Add an interaction between two compartments.

This method adds a directed edge from the source compartment to the target compartment in the transition graph, with the specified agent and rate. The rates are passed as keyword arguments and will be added to the model's parameters.

Parameters

- **source** (*str*) – Name of the source compartment
- **target** (*str*) – Name of the target compartment
- **agent** (*str*) – Name of the agent
- **rates** (*dict*) – Named parameters representing the interaction rates

Return type
`None`

Returns
None

add_spontaneous(*source*, *target*, *rate*, ***rates*)

Add a spontaneous transition between two compartments.

Parameters

- **source** (*str*) – Name of the source compartment
- **target** (*str*) – Name of the target compartment
- **rates** (*dict*) – Named parameters representing the transition rates

Returns
None

integrate(*timesteps*, ***kwargs*)

Numerically integrate the epidemic model.

Parameters

- **timesteps** (*int*) – Number of time steps to integrate
- **t_min** (*int, default=1*) – Starting time
- **seasonality** (*numpy.ndarray, optional*) – Array of seasonality values
- **kwargs** (*dict*) – Initial population of each compartment

Returns

None

simulate(*timesteps*, *seeds*, ***kwargs*)

Stochastically simulate the epidemic model

Return type

`None`

1.1.5 epidemik.utils module

exception `epidemik.utils.`**NotImplementedError**

Bases: `Exception`

exception `epidemik.utils.`**NotInitialized**

Bases: `Exception`

class `epidemik.utils.`**Parameters**

Bases: `dict`

compute_parameter(*param*)

Compute the rate from a string

Return type

`float`

Parameters: - rate: string

Rate of the transition

Returns: float

The computed rate

define_parameters(***kwargs*)

Define one or more parameter for the model

Return type

`None`

Parameters: - kwargs: keyword arguments

Named parameters for the model

Returns: None

`epidemik.utils.`**get_cache_directory**()

Return the location of the cache directory for the current platform.

1.1.6 Module contents

epidemik

epidemik is a Python package for the simulation of Compartmental Epidemic Models.

See https://www.github.com/DataForScience/epidemik for complete documentation.

CHAPTER
TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

e

`epidemik`, 9
`epidemik.EpiModel`, 1
`epidemik.MetaEpiModel`, 5
`epidemik.NetworkEpiModel`, 7
`epidemik.utils`, 8

INDEX

A

`add_age_structure()` (*epidemik.EpiModel.EpiModel method*), 1
`add_birth_rate()` (*epidemik.EpiModel.EpiModel method*), 1
`add_death_rate()` (*epidemik.EpiModel.EpiModel method*), 2
`add_interaction()` (*epidemik.EpiModel.EpiModel method*), 2
`add_interaction()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 5
`add_interaction()` (*epidemik.NetworkEpiModel.NetworkEpiModel method*), 7
`add_spontaneous()` (*epidemik.EpiModel.EpiModel method*), 2
`add_spontaneous()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6
`add_spontaneous()` (*epidemik.NetworkEpiModel.NetworkEpiModel method*), 7
`add_vaccination()` (*epidemik.EpiModel.EpiModel method*), 2
`add_vaccination()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6
`add_viral_generation()` (*epidemik.EpiModel.EpiModel method*), 3

C

`compute_parameter()` (*epidemik.utils.Parameters method*), 8

D

`define_parameters()` (*epidemik.utils.Parameters method*), 8
`download_model()` (*epidemik.EpiModel.EpiModel method*), 3
`draw_model()` (*epidemik.EpiModel.EpiModel method*), 3
`draw_model()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6

E

`epidemik`
    `module`, 9
`epidemik.EpiModel`
    `module`, 1
`epidemik.MetaEpiModel`
    `module`, 5
`epidemik.NetworkEpiModel`
    `module`, 7
`epidemik.utils`
    `module`, 8
`EpiModel` (*class in epidemik.EpiModel*), 1

G

`get_cache_directory()` (*in module epidemik.utils*), 8
`get_state()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6

I

`integrate()` (*epidemik.EpiModel.EpiModel method*), 3
`integrate()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6
`integrate()` (*epidemik.NetworkEpiModel.NetworkEpiModel method*), 7

L

`list_models()` (*epidemik.EpiModel.EpiModel method*), 4
`load_model()` (*epidemik.EpiModel.EpiModel method*), 4

M

`MetaEpiModel` (*class in epidemik.MetaEpiModel*), 5
`module`
    `epidemik`, 9
    `epidemik.EpiModel`, 1
    `epidemik.MetaEpiModel`, 5
    `epidemik.NetworkEpiModel`, 7
    `epidemik.utils`, 8

N

`NetworkEpiModel` (*class in epidemik.NetworkEpiModel*), 7

`NotImplementedError`, 8

`NotInitialized`, 8

P

`Parameters` (*class in epidemik.utils*), 8

`plot()` (*epidemik.EpiModel.EpiModel method*), 4

`plot()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6

`plot_peaks()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 6

R

`R0()` (*epidemik.EpiModel.EpiModel method*), 1

`R0()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 5

`R0()` (*epidemik.NetworkEpiModel.NetworkEpiModel method*), 7

S

`save_model()` (*epidemik.EpiModel.EpiModel method*), 4

`simulate()` (*epidemik.EpiModel.EpiModel method*), 4

`simulate()` (*epidemik.MetaEpiModel.MetaEpiModel method*), 7

`simulate()` (*epidemik.NetworkEpiModel.NetworkEpiModel method*), 8

`single_step()` (*epidemik.EpiModel.EpiModel method*), 5