Getting Started with Episimmer

A tutorial for beginner or non coders.

This is a simulation platform that will allow you to customize and simulate epidemic spread. You can easily set up basic simulations of SIR, SEYAR models and also complex simulations that model entire campuses with different buildings and heterogenous agents.

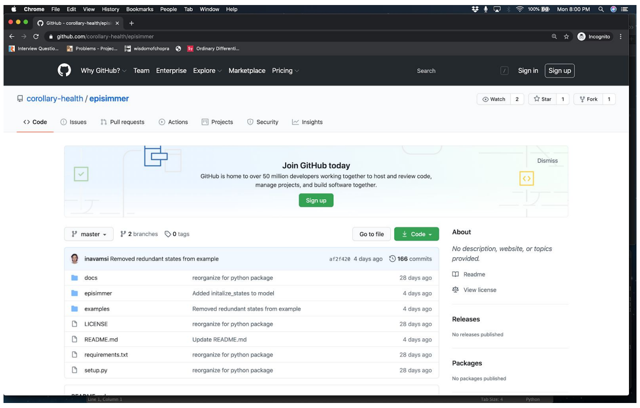
To use this tool, you do not need to know python. Although it will help to take advantage of the full capacity of the package. But for basic simulations, altering the examples is as simple as changing a few parameters. To run this simulation platform you will require python version 3.6 and above to be installed. To check your version try the following command on the terminal/command line without the angular brackets <python --version>. Alternatively if you don't have it, download python here.

Required Background for tutorial

- What is a compartmental model : Link
- Navigating directories (cd) on Terminal/Command Prompt/Command Line : Link

How to Download Simulation Package

1. Go to https://github.com/corollary-health/episimmer
   A page like this will open up.



Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

inavamsi Removed redundant states from example af2f420 4 days ago 166 commits

docs	reorganize for python package	28 days ago
episimmer	Added initalize_states to model	4 days ago
examples	Removed redundant states from example	4 days ago
LICENSE	reorganize for python package	28 days ago
README.md	Update README.md	4 days ago
requirements.txt	reorganize for python package	28 days ago
setup.py	reorganize for python package	28 days ago

About

No description, website, or topics provided.

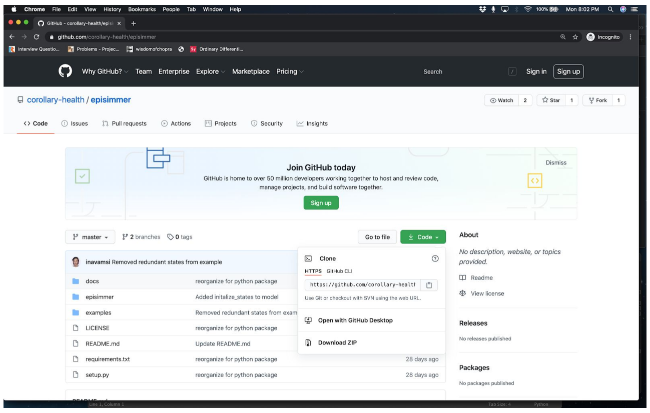
Releases

No releases published

Packages

No packages published

2. Click the download code button in green.


github.com/corollary-health/episimmer
Why GitHub? Team Enterprise Explore Marketplace Pricing
Search
Sign in
Sign up
corollary-health / episimmer
Watch 2
Star 1
Fork 1
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Join GitHub today
GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.
Sign up
Dismiss
master
2 branches
0 tags
Go to file
Code
inavamsi Removed redundant states from example
docs
reorganize for python package
episimmer
Added initalize_states to model
examples
Removed redundant states from exam
LICENSE
reorganize for python package
README.md
Update README.md
requirements.txt
reorganize for python package
28 days ago
setup.py
reorganize for python package
28 days ago
Clone
HTTPS
GitHub CLI
https://github.com/corollary-healt
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP
About
No description, website, or topics provided.
Readme
View license
Releases
No releases published
Packages
No packages published

3. Now click download ZIP
4. You will find ‘episimmer-master.zip’ in your Downloads
5. Unzip this file and move it to wherever you want.

Platform Brief

The platform has been built with the notion of flexibility and wide variety of uses. Thus some of the simpler functions may seem a bit superfluous but will make sense once the higher functionalities are explored. The aim of the platform was to be completely modular. This means that one can plug and play with different models, underlying structure of interactions, restrictions and parameters.

As of right now there are two basic types of models that can be plugged in.

- Stochastic Model : This model allows us to implement compartmental models with probabilistic changes in state.
- Scheduled Model : The difference is that changes are scheduled and not probabilistic. For example an Infected can get recovered with a distribution in the number of days. This helps implement such models.

Building blocks of the platform

- Agents : In standard SIR models agents are assumed to be homogenous. This platform supports agent based modelling where each agent can have a different set of parameters that include stuff like Genetic markers, demography, residence, mobility, policy state(quarantined, vaccinated,...)...
- Locations : The user can choose to create multiple locations in which agents can interact.

Two major types of interaction

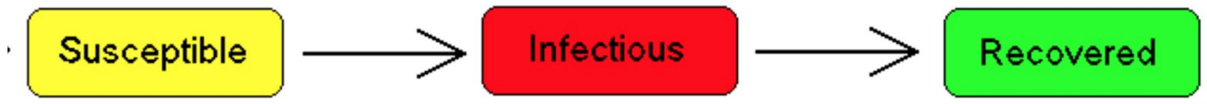
- Events : Events happen at locations. FOr example if Classroom A is a location then Elementary Physics and Elementary Chemistry are events that happen at the location of classroom A
- Individual Interactions : There are individual interactions between two agents that are location agnostic. USing this we can alternatively build up an interaction graph.

The platform supports policy and restrictions, which will be discussed in further tutorials.

Getting started with basic SIR Model

SIR Model

We start with the basic stochastic SIR model occurring in a region for 1000 agents. Note that the simulation platform is not continuous(as in, it does not solve the differential equations but simulates it out) like the equations below but discrete in both time and number of agents. As the time step becomes smaller and the number of agents increase the plot will tend towards a continuous one as produced by the equations.



$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$
$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

There are two ways to run a basic SIR model. One is by letting all agents interact in an event at a location repeatedly, this is analogous to agents in a room. The second approach is by running the model on a complete graph with an edge between every two agents. We shall start with the first approach of agents in an event.

- SIR with agents at one event : ‘**Stochastic_SIR’**’ **-** this runs very quickly as each agent contributes to and receives from a scalar ambient infection.
- SIR with agents on a graph : ‘**Stochastic_SIR_complete_graph’ -** this assumes a complete graph instead and takes longer to run.

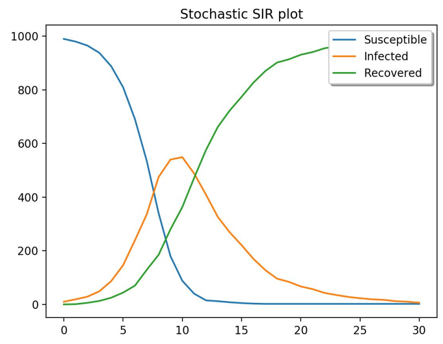
Running example

To run ‘**Stochastic_SIR**’’ follow the following steps.

1. Change directory to where epissimer-master is stored. This is the package that you have downloaded from the github link. If you haven’t moved the file after unzipping it, it will still be in your downloads folder.
2. Run following commands in terminal/command line :

```
cd episimmer-master/examples

python ../episimmer/Main.py Stochastic_SIR
```

If you have done the above steps correctly without errors you will see an image like this pop up.

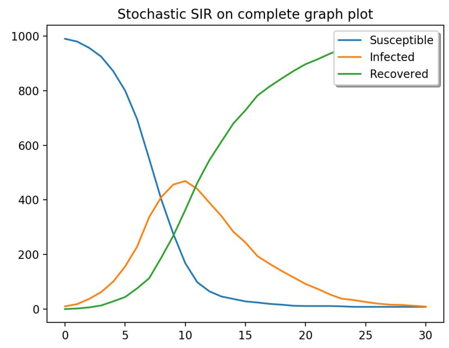


Alternatively to run ‘**Stochastic_SIR_complete_graph’** follow the steps below :

1. Change directory to where epissimer-master is stored. This is the package that you have downloaded from the github link. If you haven’t moved the file after unzipping it, it will still be in your downloads folder.
2. Run following commands in terminal/command line :

```
cd episimmer-master/examples

python ../episimmer/Main.py Stochastic_SIR_complete_graph
```

If you have done the above steps correctly without errors you will see an image like this pop up. Note that this will take significantly longer to run as you will have to go through 999000 interactions(directed graph) at every time step.



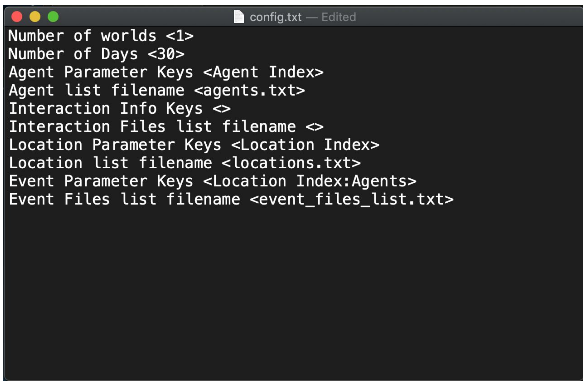
One may wonder why we might use a graph if it takes more time. This is because in the real world every agent does not interact with every other agent. So there is an underlying interaction graph with each agent only interacting with a few other agents. We shall show some examples with some random graphs

Making changes

For simplicity we shall show how to make changes on **‘Stochastic_SIR’** example but it will be the same for other examples also.

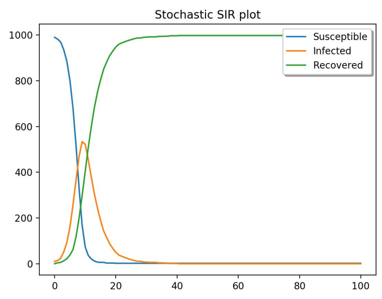
If you want to make changes to the plot and experiment with different rates, it is as follows. To make changes to an example, open the respective example file, in this case the Stochastic_SIR file inside the examples folder.
You can make multiple changes in config.txt(open using text editor) and UserModel.py(open using python editor). Don't worry about the other files for now.

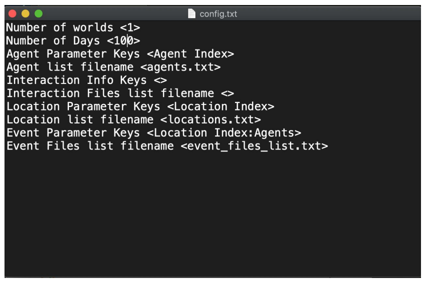
1. Plotting details : to make changes to details like total number of days of plotting and averaging over multiple simulations changes can be made in the config.txt text file. The config.txt file should look like this.



```
Number of worlds <1>
Number of Days <30>
Agent Parameter Keys <Agent Index>
Agent list filename <agents.txt>
Interaction Info Keys <>
Interaction Files list filename <>
Location Parameter Keys <Location Index>
Location list filename <locations.txt>
Event Parameter Keys <Location Index:Agents>
Event Files list filename <event_files_list.txt>
```

   a. **Total number of days** : By changing total number of days in the config.txt file the x axis of the plot will change accordingly. For example, increasing the total number of days to 100 will make the plot look like this.

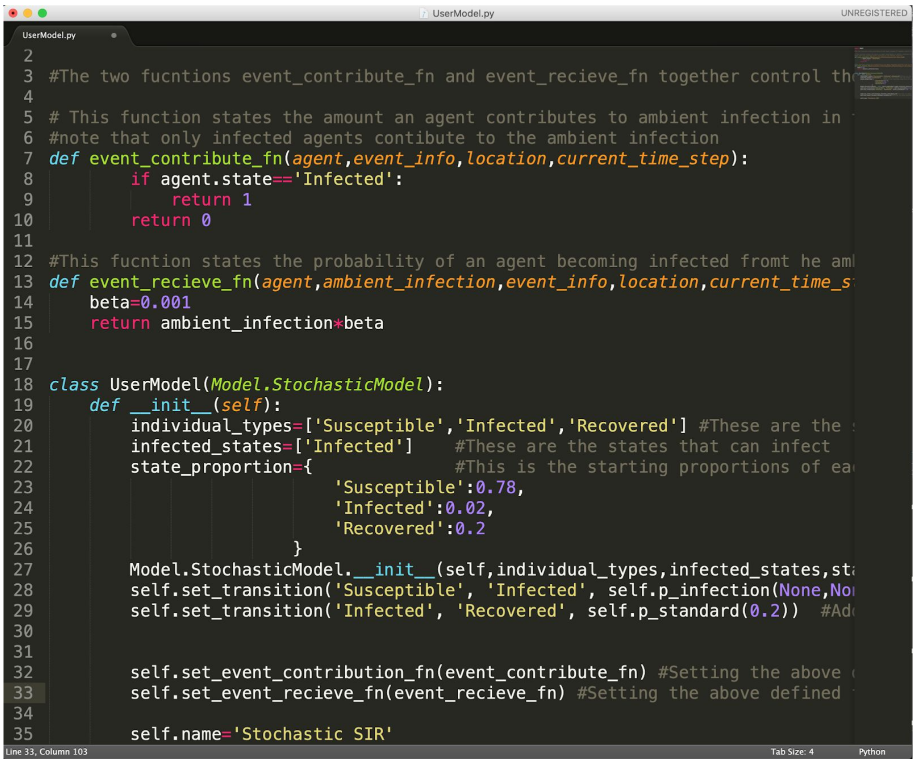




```
Number of worlds <1>
Number of Days <100>
Agent Parameter Keys <Agent Index>
Agent list filename <agents.txt>
Interaction Info Keys <>
Interaction Files list filename <>
Location Parameter Keys <Location Index>
Location list filename <locations.txt>
Event Parameter Keys <Location Index:Agents>
Event Files list filename <event_files_list.txt>
```

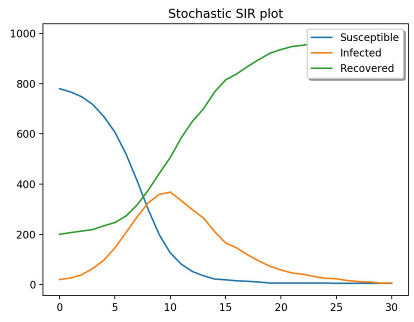
   b. **Averaging over multiple plots** : By changing Number of worlds, the platform will run your simulation that many times and print the averaged plot. This will reduce noise and variance of the simulation. The plot won't vary much for this simple SIR model but this will be more evident in more complex examples.

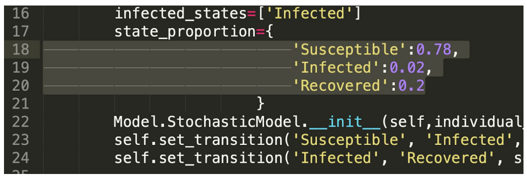
2. Changing the model : to make changes in the model you will have to open UserModel.py. It will look like this.



```
UserModel.py                                                          UNREGISTERED
UserModel.py
 2
 3 #The two fucntions event_contribute_fn and event_recieve_fn together control the
 4
 5 # This function states the amount an agent contributes to ambient infection in
 6 #note that only infected agents contibute to the ambient infection
 7 def event_contribute_fn(agent,event_info,location,current_time_step):
 8         if agent.state=='Infected':
 9             return 1
10         return 0
11
12 #This fucntion states the probability of an agent becoming infected fromt he aml
13 def event_recieve_fn(agent,ambient_infection,event_info,location,current_time_s
14     beta=0.001
15     return ambient_infection*beta
16
17
18 class UserModel(Model.StochasticModel):
19     def __init__(self):
20         individual_types=['Susceptible','Infected','Recovered'] #These are the
21         infected_states=['Infected']    #These are the states that can infect
22         state_proportion={              #This is the starting proportions of ea
23                             'Susceptible':0.78,
24                             'Infected':0.02,
25                             'Recovered':0.2
26                         }
27         Model.StochasticModel.__init__(self,individual_types,infected_states,sta
28         self.set_transition('Susceptible', 'Infected', self.p_infection(None,No
29         self.set_transition('Infected', 'Recovered', self.p_standard(0.2))  #Ad
30
31
32         self.set_event_contribution_fn(event_contribute_fn) #Setting the above
33         self.set_event_recieve_fn(event_recieve_fn) #Setting the above defined
34
35         self.name='Stochastic SIR'
Line 33, Column 103                                        Tab Size: 4    Python
```

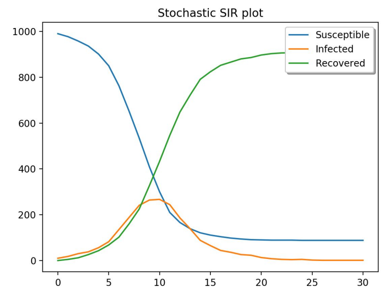
a. **Initial proportions** : You can make changes to the initial proportions by changing the values in line 18-20. Just ensure that the proportions add up to 1. For example, in a population if initially 20% are immune or recovered before the epidemic breakout.

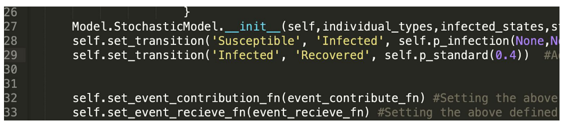




```
16         infected_states=['Infected']
17         state_proportion={
18                                 'Susceptible':0.78,
19                                 'Infected':0.02,
20                                 'Recovered':0.2
21                         }
22         Model.StochasticModel.__init__(self,individual
23         self.set_transition('Susceptible', 'Infected',
24         self.set_transition('Infected', 'Recovered', s
```

b. **Changing rate of recovery** : To change rate of recovery one will have to change the value at line 29 when defining the I->R transition. We use the inbuilt function p_standard to denote that this is a fixed rate. We shall see rates that change with time in later examples. Note that we have increased the rate of recovery to 0.4 and thus reduced the peak significantly from before as agents recover faster.



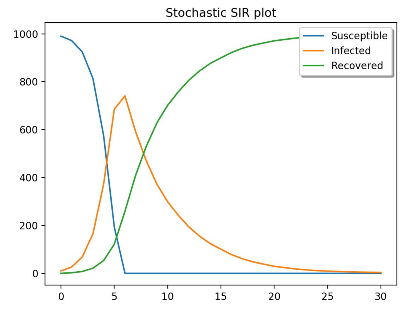


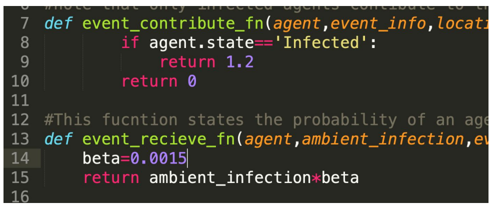
```
                    }
        Model.StochasticModel.__init__(self,individual_types,infected_states,st
        self.set_transition('Susceptible', 'Infected', self.p_infection(None,No
        self.set_transition('Infected', 'Recovered', self.p_standard(0.4))  #Ad


        self.set_event_contribution_fn(event_contribute_fn) #Setting the above
        self.set_event_recieve_fn(event_recieve_fn) #Setting the above defined
```

c. **Changing rate of infection :** There are two ways to change the rate of infection

- Increasing each agents contribution to ambient infection
- Increasing each agents receiving probability from ambient infection

This can be done by changing the values in event_contribute_fn and event_recieve_fn.





```
def event_contribute_fn(agent,event_info,locati
        if agent.state=='Infected':
            return 1.2
        return 0

#This fucntion states the probability of an age
def event_recieve_fn(agent,ambient_infection,ev
    beta=0.0015
    return ambient_infection*beta

```

d. **Changing number of agents** : This is an agent based simulation platform and thus we have to change the number of agents. This can be done by changing agents.txt. It is recommended that one does not do this by hand and instead generates it using code. We have created a python script that does this for you. Run the following commands to create 800 agents.

```
cd episimmer-master/examples/Stochastic_SIR
python generate_files.py 800
```

The above commands will create the necessary agents.txt and one_event.txt file. Now you just have to run the code normally. Remember that when you change the number of agents you have to also change the rates of infection if you want the same shape. This is because ambient infection and depends on total agents.