

Virus Transmission with Super-spreaders Model

J M Applegate

September 10, 2021

Model Purpose

A curious aspect of the Covid-19 pandemic is the clustering of outbreaks. Evidence suggests that 80% of people who contract the virus are infected by only 19% of infected individuals, and that the majority of infected individuals fail to infect another person.¹ Thus, the dispersion of a contagion, k , may be of more use in understanding the spread of Covid-19 than the reproduction number, R_0 .

The Virus Transmission with Super-spreaders model, written in NetLogo, is an adaptation of the canonical Virus Transmission on a Network model² and allows the exploration of various mitigation protocols such as testing and quarantines with both homogenous transmission and heterogenous transmission.

The model consists of a population of individuals arranged in a network, where both population and network degree are tunable. At the start of the simulation, a subset of the population is initially infected. As the model runs, infected individuals will infect neighboring susceptible individuals according to either homogenous or heterogenous transmission, where heterogenous transmission models super-spreaders. In this case, k is described as the percentage of super-spreaders in the population and the differing transmission rates for super-spreaders and non super-spreaders. Infected individuals either recover, at which point they become resistant to infection, or die. Testing regimes cause discovered infected individuals to quarantine for a period of time.

Entities, State Variables and Scale

The model entities are individuals arranged in a network. Each individual, or node, has the following state variables:

infected?: if true, node is infected

resistant?: if true, node is recovered

quarantine?: if true, node is in quarantine

dead?: if true, node has died of an infection

spreader?: if true, for heterogenous transmission node is a super-spreader

infected-timer: time elapsed since node became infected

¹Adam, D.C., Wu, P., Wong, J.Y. et al. Clustering and super spreading potential of SARS-CoV-2 infections in Hong Kong. Nat Med 26, 1714–1719 (2020).

²Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

test-timer: time elapsed since node was tested

quarantine-timer: time elapsed since the start of quarantine

The model's global variables, set on the interface, describe the population setup, transmission variables, virus characteristics and interventions.

population: number of network nodes, or individuals, in the model

average-degree: average number of neighbors for each node

initial-outbreak-size: number of nodes initially infected

heterogenous: if true, super-spreaders nodes transmit at a higher rate than non super-spreader nodes

transmission: for homogenous transmission, the transmission probability for all nodes

spreader-frequency: for heterogenous transmission, the percentage of super-spreaders in the population

spreader-transmission: for heterogenous transmission, the transmission probability for super-spreaders

non-spreader-transmission: for heterogenous transmission, the transmission probability for nodes that are not super-spreaders

max-recovery-time: the maximum time a node will be infected

morbidity: the probability an infected node will die

test-frequency: how often nodes are tested

quarantine-time: how long nodes have been in quarantine.

Process Overview and Scheduling

The model is set up with susceptible nodes linked into a network, a subset of nodes randomly infected in an initial outbreak, and in the case of heterogenous transmission, super-spreaders randomly selected.

Each time step, all node timers are incremented. Each infected node not in quarantine infects its susceptible neighbors not in quarantine with a probability given by the appropriate transmission value. Resistant or quarantined nodes cannot be infected.

Infected nodes are not immediately aware that they are infected. Every so often the nodes are tested for infection, and if infected a node will move into quarantine. Nodes that are infected will either recover or die within a specified amount of time. Nodes in quarantine leave quarantine after a specified number of time steps, and the cycle repeats until no nodes are infected. This sequence is illustrated in Figure 1.

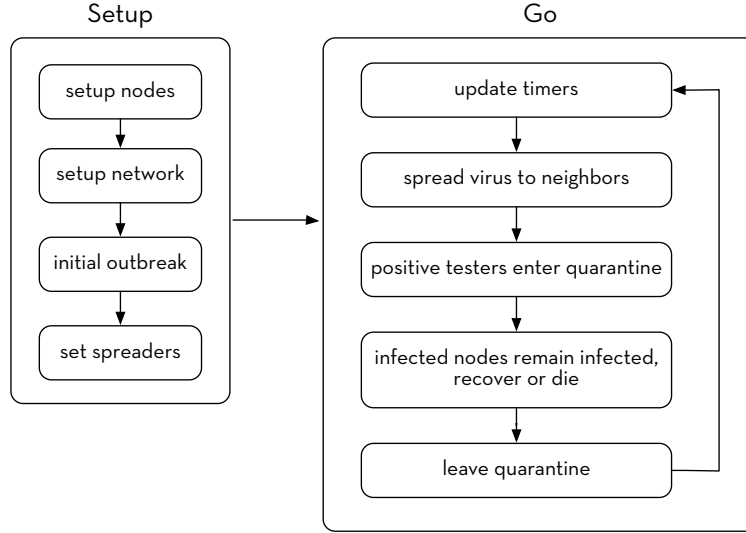


Figure 1: Virus Transmission with Super-spreaders model overview.

Inputs and Outputs

The model does not require any inputs other than global settings such as transmission variables or virus characteristics on the interface. The model outputs both a world graphic showing the evolution of the network node states, as well as a plot showing the numbers of susceptible, infected and recovered (or resistant) over time. In the world graphic, susceptible nodes are pale blue, infected nodes are red, recovered nodes are white and dead nodes are black. In the case of heterogenous transmission, spreader nodes are larger than non-spreader nodes. Links where transmission is no longer possible are gray. The SIR output parameters are summarised in Table 1.

Parameter	Description
susceptible	nodes where not infected? and not recovered? and not dead?
infected	nodes where infected?
resistant	nodes with recovered?
dead	nodes with dead?

Table 1: Output parameters in the interface plot panel based on true or false values for specified variables.

Submodels

For the initial model's `setup` function:

Setup Nodes

for each node:

```

place node in the world
set infected? and resistant? false
set test-timer = random ( test-frequency )

```

```

    set infected-timer and quarantine-timer = 0
    set quarantine?, dead? and spreader? false
Setup Network
num-links = average-degree * population / 2
while links < num-links:
    link a random node with its closest non-neighbor node
layout network in spring format
Initial Outbreak
randomly select initial-outbreak-size nodes
for selected nodes:
    set infected? true
    set infected-timer = 1 + random ( max-recovery-time )
Select Spreaders
randomly select spreader nodes
for selected nodes:
    set spreader? true

```

For each time step, gathered under the **go** function,

```

Update Timers
for each node:
    increment test-timer
    if test-timer ≥ test-frequency:
        set test-timer = 0
    if infected?:
        increment infected-timer
        if infected-timer ≥ max-recovery-time:
            set infected-timer = 0
    if quarantine?:
        increment quarantine-timer
        if quarantine-timer ≥ quarantine-time:
            set quarantine-timer = 0

```

Spread Virus

```

if heterogenous?:
    for nodes with infected? and spreader? and not quarantine?:
        for neighbors with not resistant?, quarantine?, infected? or dead?:
            if random(100) < spreader-transmission:
                set infected? true
                set infected-timer 1 + random ( max-recovery-time )
    for nodes with infected? and not quarantine? and spreader?:

```

```

    for neighbors with not resistant?, quarantine?, infected? or dead?:
        if random(100) < non-spreader-transmission:
            set infected? true
            set infected-timer 1 + random ( max-recovery-time )
else:
    for nodes with infected? and not quarantine?:
        for neighbors with not resistant?, quarantine?, infected? or dead?:
            if random(100) < transmission:
                set infected? true
                set infected-timer 1 + random ( max-recovery-time )

Enter Quarantine
for nodes with infected? and test-timer = 0:
    set quarantine? true
    set quarantine timer = 1

Recover
for nodes with infected? and infected-timer = 0:
    if random(100) < morbidity:
        set dead? true
        set quarantine?, infected? and resistant? false
    else:
        set infected? false and resistant? true

Leave Quarantine
for nodes with quarantine? and quarantine-timer = 0:
    set quarantine? false

```