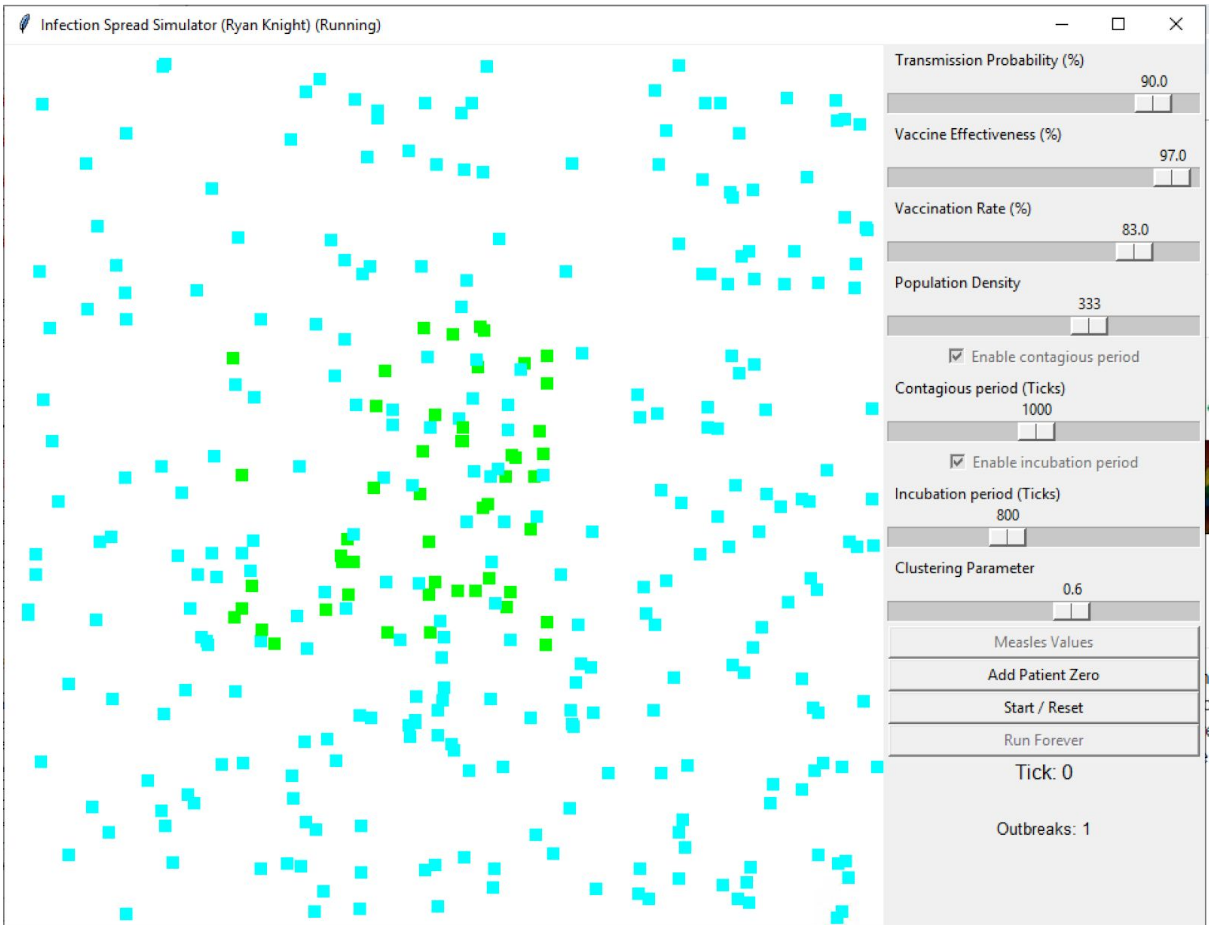
Infection Spread Simulator

Ryan Knight



Infection Spread Simulator (Ryan Knight) (Running)

Transmission Probability (%)
90.0

Vaccine Effectiveness (%)
97.0

Vaccination Rate (%)
83.0

Population Density
333

☑ Enable contagious period

Contagious period (Ticks)
1000

☑ Enable incubation period

Incubation period (Ticks)
800

Clustering Parameter
0.6

Tick: 0

Outbreaks: 1

Group project: Community Immunity

Does Clustering of Unvaccinated Individuals and a Low Level of Vaccination Increase Measles Infection in a Given Population: A Model

Hannah Clark, Bryce Drylie, Lara Hillyard, Ryan Knight, Tom McKeown, Liz Phillips

This User Guide explains the function and approach of the basic Infection Spread Simulator that I produced for our group project.

Contents

Basis of the Simulation

A variety of dots exist on the screen moving with their own x and y direction characteristic, bouncing off walls. This means that they effectively each have a predefined path, rather than random movements.

As real life individuals have some form of routine, this method is more suitable over strictly random movements.

Setting up the simulation

The following parameters can be varied:

- Population density (0-500)
- Transmission rate (0-100%, resolution 0.1)
- Vaccine effectiveness (0-100%, resolution 0.1)
- Vaccination rate (0-100%, resolution 0.1)
- Contagious period (0-2000 ticks, resolution 100)
- Incubation period (0-2000 ticks, resolution 100)
- Clustering parameter (0-1, resolution 0.05)

This allows a wide variety of simulation conditions to be experimented with, with the main limitation being population density, due to the increased processing strain.

Population Density: The number of dots in the uninfected population on the screen. Inherently a density due to the set size of the simulation window.

Transmission Rate: The probability that on collision with a transmissible infected individual, an unvaccinated individual will become infected. (Inherent limitations, see: Collisions)

Vaccine Effectiveness: The probability that on collision with a transmissible infected individual, a vaccinated individual will not become infected. Effectively a "Vaccine success rate". (Inherent limitations, see: Collisions)

Vaccination Rate: The proportion of uninfected individuals generated on the screen (*by Population Density*) that are vaccinated.

Contagious Period: Can be enabled or disabled. (Disabled by default.) The amount of simulation ticks (a measure of time) that a newly infected individual is able to infect others before becoming non-contagious. (Non-contagious individuals are visually different)

Incubation Period: Can be enabled or disabled. (Disabled by default.) The amount of simulation ticks that a newly infected individual waits before becoming contagious. (Incubating individuals are visually different)

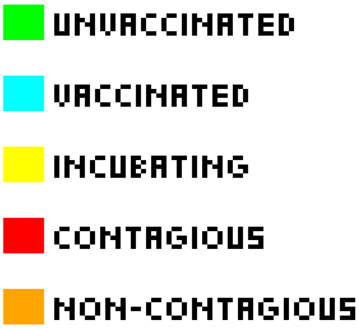
States of Individuals

Primary states of characters

- Blue (Vaccinated)
- Green (Unvaccinated)
- Red (Infectious)

Optional, additional states of characters

- Yellow (Infected, incubation period)
- Orange (Infected, but non-contagious)



(Image credit: Liz Phillips)

Movement and Collisions

Game ticks and movement

Each individual has a predetermined discrete x and y movement value between -2 and 3, “rolled” on creation. The individuals are created when the simulation is started.

These movement values don’t change until the program resets or is restarted. It’s possible to roll both as zero and the dot won’t move. This is allowed to happen to accommodate for people that are sedentary, or unable to move particularly far.

When dots hit the side, their inherent x and y invert. This means they will fundamentally follow a predetermined path. At very low population densities this means dots can never interact (*The high values we use somewhat mitigates this, See: Caveats.)*

While it would have been relatively simple to do, the movement paths were not made pseudorandom, as people do not follow random paths in real life, and have routines. The limitation of this is the aforementioned possibility to never collide at low population densities (See: Caveats)

Every dot moves round the screen each “tick”- the counter of simulation time. Every tick, each dot moves their movement values.

Collisions

As part of each simulation tick, individuals are checked to see if an uninfected individual has collided (overlapped) with a contagious individual. If this is true, the uninfected individual will have a possibility of becoming contagious (*or Incubating, if this is enabled. See: Incubation Period, above*).

As the individuals are 10x10 pixels, it would possible for one overlapping to count as multiple collisions. To prevent this, once collided, an uninfected individual has a low preset number of ticks before its collisions will be “acknowledged” by the program again.

Running the Program

Starting the Program and Cycling

The program has two start buttons:

- “Start/Stop” - Single Cycle mode
- “Run Forever” - Repeating Cycle mode

The program will not begin simulating until one of these buttons has been pressed. The title of the program window will change to reflect that the program is running.

When the program starts simulating, all slider values are used to create the simulation.

While running, sliders are immutable. This stops messed up results if they’re interrupted while running. Depending on whether the program is running in Single or Repeating Cycle, the other mode’s start/stop button will be disabled. This means that no values can be changed during the program’s operation to preserve the integrity of the results.

When “running forever” in Repeated Cycle mode, the system will reset and randomly generate new individuals according to chosen slider values. The previously generated individuals movement values do not persist, and so the pathing and locations of the individuals is entirely new.

Beginning the Infection

New individuals can be created, called “Patient Zero”. This is how contagious individuals are added to the uninfected population at a random location and movement set, the same way uninfected individuals are generated. (*It is therefore also possible for Patient Zero to be stationar*y.) The tick counter begins the moment the first *Patient Zero* is added to the simulation.

In Single Cycle mode, more than one *Patient Zero* can be spawned in as the button is not locked. The interface is updated with an “Outbreaks” counter to track the number of new initial contagious individuals added to the population.

In Repeated Cycle mode, there is only one *Patient Zero*, generated immediately for efficiency.

Ending the simulation and Extracting Data

Restarting in Repeated Cycle Mode

In *Repeated Cycle* mode, the simulation runs for a total of 7500 ticks before restarting. This is the "timeout point" and is hard coded. The majority of non-extreme scenarios will have reached epidemic levels by this point.

Measuring Infection Rate

The most useful information generated through the program is that used to measure infection rate. This can be seen in two ways, both of which are available in the program's outputs:

- Time to hit epidemic levels, the number of ticks when 50% of the initial population (not including *Patient Zero* or its duplicates in *Single Cycle*) are infected.
- The amount of individuals in infected states when the program "times out" at 7500 ticks in *Repeated Cycle Mode*. As the timeout introduces a time frame, this makes the simple number of infected individuals a measurement of infection rate in itself.

Both of these have their advantages and disadvantages, so both are available to provide the user greater freedom when choosing their parameters of their experiments.

- Counting the infected individuals is less effective for High Susceptibility populations (eg. low vaccination rates), as it's more likely they all will be infected by the *Timeout Time*.
- *Epidemic Time* is less effective for Low Susceptibility populations (eg. the highest vaccination rates) as they may never hit epidemic levels by the *Timeout Time*.

Extracting Data

In *Repeated Cycle mode*, information is recorded and exported for efficient handling of mass repeats from the same experimental parameters. The main purpose of *Repeated Cycle* mode is to generate experimental data, so recording this data automatically saves time by beginning the next cycle immediately.

When the simulation cycle is complete, information is exported to a new line in a text file called "log" in the same directory as the program.

Each simulation cycle's output is a new line, including:

- The values of all sliders (as explanatory metadata for the generated information)
- The number of *outbreaks* (Should always be 1)
- The numbers of individuals in each state
- The *Epidemic Time* (if hit, else the *Timeout Time*, 7500 ticks)

Significantly, this wide variety of outputs allows the user a wide range of freedom to generate analyse data in the form that they want.

In the Context of Measles

The group project this software has been created for has been investigating the measles virus. For this, there is a button in the program that sets the slider values/parameters to reflect that of the measles virus. The appropriate parameters have been estimated by the following:

Transmission rate - 90.0%

CDC: Transmission of Measles [Internet]; 2018 Feb 5 [cited 2019 Nov 08]. Available from: https://www.cdc.gov/measles/transmission.html

Vaccine effectiveness - 97.0%

Estimated from two doses of the MMR vaccine.

CDC: Measles, Mumps, and Rubella (MMR) Vaccination: What Everyone Should Know [Internet]; 2019 Mar 28 [cited 2019 Nov 08]. Available from: https://www.cdc.gov/vaccines/vpd/mmr/public/index.html

Incubation period - 800
Contagious period - 1000

The ratio between incubation and contagious is what is important- it is proportional to the difference in number of real life days. Due to the arbitrary nature of the simulation tick, no direct parallel is drawn to the day from the tick, but the ratio can be directly applied.

Clustering of Unvaccinated Individuals

The Clustering Parameter applies a “Clustering” element to a proportion of unvaccinated individuals. ***When in various stages of infection, clustered individuals will still obey clustering- it is an attribute of the individual and not the individual’s state.***

It is important to note that the *Clustering Parameter* won’t have any effect on a fully vaccinated population.

The Clustering Parameter defines:

- The proportion of initially unvaccinated individuals that will “obey” clustering. Above 0.5, all initially unvaccinated individuals will.
- The size of the clustering region- the individual can move around the central point, but no further in x or y than 250 multiplied by the C*lustering Parameter*. This means that a smaller decimal Clustering Parameter will increase the clustering area.

These numbers were chosen to try and prevent the *Clustering Parameter* from changing the extent of clustering exponentially and give a more visually linear aspect to the *Clustering Parameter.*

The central point of clustering is a random location each simulation cycle. The location selection is limited to ensure that the clustering area will not cross over the natural edges of the simulation and skew the results.

Individuals obeying clustering will treat the edge of the cluster region as they would the simulation's edges/walls.

Caveats of the Model

Infection spread is a complex process, and depends on many factors that are naturally out of the scope and design of a simple, highly abstract model such as this- the model certainly can't reflect the real world! I can't list them all here, but below are some of interest.

Some Inherent Caveats

The infected are likely to isolate themselves at home, or quarantine themselves in a hospital in infection spread limiting conditions.

The age, location and socioeconomic status of individuals can impact their ability to access healthcare, and so impact their possibility of being cured and their chance of transmitting the virus.

The level of resistance of a vaccinated person will likely vary- for example, with age.

There is an assumption that the vaccines are administered correctly and that individuals are given the full course of vaccinations, and by extension that the vaccine works on everyone.

Immunocompromised people that cannot be vaccinated aren't considered.

Cases are ignored in which people develop measles from the vaccine. This is reasonable, as this is a reduced effect and they are unlikely to transmit.

List of Model Specific Caveats

The *Clustering Parameter* won't do anything to a fully vaccinated population. Any experimental design using this program should take notice of this.

Due to the predetermined, and therefore not random, pathing of individuals it is possible that they will never interact at very low population densities. Experiments using this software shouldn't use very small populations to mitigate this.

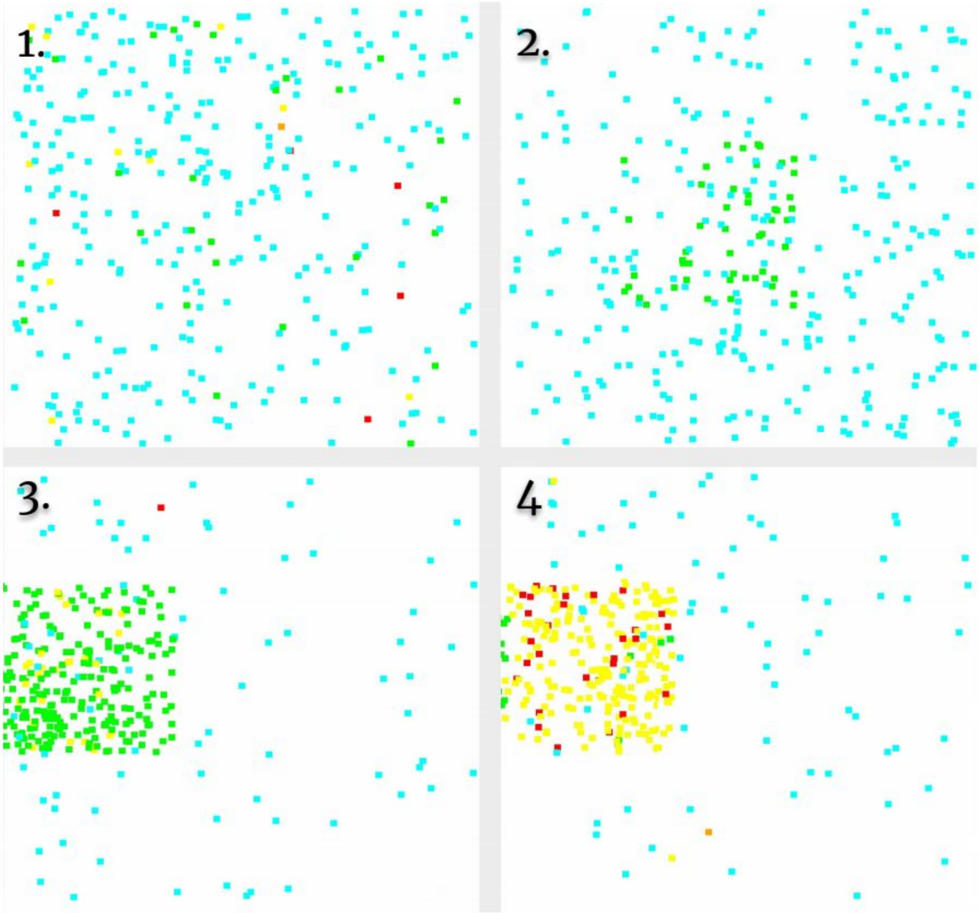
As previously mentioned, the fact that there’s no attempt at pseudorandom movement could be construed as a limitation, but was left in real life people follow routines anyway.

The model will allow disease to start in isolated healthy communities, which given human-human transmission shouldn’t be possible. This can be somewhat mitigated by a large number of repeats.

Clustered communities contain only unvaccinated individuals.

As previously mentioned, there is no relation between the simulation tick and the day, though infectious period and incubation period can be aligned proportionately to each other. (*See: In the Context of Measles*)

Example Simulations



1: High vaccination rate, no clustering, 1502 ticks.
2: High vaccination rate, clustering, no ticks.
3: Low vaccination rate, clustering, 551 ticks.
4: Low vaccination rate, clustering, 1405 ticks.