



SOFTWARE TOOL ARTICLE

VTES: a stochastic Python-based tool to simulate viral transmission [version 1; peer review: awaiting peer review]

Danielle L. Kurtin ¹, Daniel A.J. Parsons ², Scott M. Stagg^{3,4}

¹Department of Psychology, University of Surrey, Guildford, GU2 7XH, UK

²School of Life Sciences, Pharmacy and Chemistry, Kingston University, London, KT1 2EE, UK

³Institute of Molecular Biophysics, Florida State University, Tallahassee, Florida, 32306, USA

⁴Department of Chemistry and Biochemistry, Florida State University, Tallahassee, FL, USA

V1 First published: 05 Oct 2020, 9:1198
<https://doi.org/10.12688/f1000research.26786.1>

Latest published: 05 Oct 2020, 9:1198
<https://doi.org/10.12688/f1000research.26786.1>

Abstract

The spread of diseases like severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) in human populations involve a large number of variables, making it difficult to predict how it will spread across communities and populations. Reduced representation simulations allow us to reduce the complexity of disease spread and model transmission based on a few key variables. Here we have created a Viral Transmission Education Simulator (VTES) that simulates the spread of disease through the interactions between circles representing individual people bouncing around a bounded, 2D plane. Infections are transmitted via person-to-person contact and the course of an outbreak can be tracked over time. Using this approach, we are able to simulate the influence of variables like infectivity, population density, and social distancing on the course of an outbreak. We also describe how VTES's code can be used to calculate R_0 for the simulated pandemic. VTES is useful for modeling how small changes in variables that influence disease transmission can have large changes on the outcome of an epidemic. Additionally, VTES serves as an educational tool where users can easily visualize how disease spreads, and test how interventions, like masking, can influence an outbreak. VTES is designed to be simple and clear to encourage user modifications. These properties make VTES an educational tool that uses accessible, clear code and dynamic simulations to provide a richer understanding of the behaviors and factors underpinning a pandemic. VTES is available from: <https://github.com/sstagg/disease-transmission>.

Keywords

simulation, modelling, Python, SARS-CoV-2, disease model, educational tool

Peer review discontinued

At the request of the author(s), this article is no longer under peer review. [What does this mean?](#)

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the **Pathogens** gateway.



This article is included in the **Python** collection.



This article is included in the **Mini Hacks** collection.

Corresponding author: Scott M. Stagg (sstagg@fsu.edu)

Author roles: Kurtin DL: Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Parsons DAJ: Writing – Original Draft Preparation, Writing – Review & Editing; Stagg SM: Conceptualization, Software, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: SS was supported in part by the National Institutes of Health [R01 GM108753]. DK was supported in part by the Vice Chancellor Studentship at the University of Surrey.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2020 Kurtin DL *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Kurtin DL, Parsons DAJ and Stagg SM. **VTES: a stochastic Python-based tool to simulate viral transmission [version 1; peer review: awaiting peer review]** F1000Research 2020, 9:1198 <https://doi.org/10.12688/f1000research.26786.1>

First published: 05 Oct 2020, 9:1198 <https://doi.org/10.12688/f1000research.26786.1>

Introduction

As of the start of this year (2020) a novel coronavirus originating out of Wuhan-China (which became known as severe acute respiratory syndrome coronavirus 2, or SARS-CoV-2), rapidly spread worldwide and became a global pandemic, the magnitude of which had not been seen since the Spanish influenza pandemic a century ago. At the time of writing (09/2020), SARS-CoV-2 has resulted in >27 million cases and >900,000 deaths in 188 countries worldwide¹, with a case-fatality rate of >15% in some areas^{1,2}.

One key factor of the disease's devastating impact is how rapidly and pervasively it spreads throughout a population. The factors that influence the spread of infectious diseases are enormously complex, involving numerous variables, which are often initially unknown to researchers. Such variables include the rate of transmission, population density, and the disease mortality rate in a population. Therefore, it is unsurprising that dubious science and confounding arguments has resulted in mixed messages and false claims being made on how to limit the spread of this viral disease³. Misinterpreted science and these misleading reports have generally resulted in mass confusion, and in some cases, resistance to many crucial disease control techniques, such as mask wearing, social distancing, and self-isolation. This has predictably lead to a substantial rise in SARS-CoV-2 cases and deaths⁴.

Though the complexities of a pandemic are difficult to account for, simulations and models offer us a chance to both explore the mechanisms underpinning a pandemic and potential solutions to mitigate its threat. Another benefit of simulations is that, by reducing the complexity of disease spread, they can be useful educational tools for clearly demonstrating how recommendations from medical professionals will affect the course of an epidemic. Along those lines, medically-based simulations are increasingly being developed to teach and demonstrate the effects of differing courses of actions on an outcome, and the rise in SARS-Cov-2 has provided more use for these models than ever before. One example is the Washington Post's "Coronavirus Simulator"⁵. The article utilizes several simulations and infographics to effectively demonstrate how quickly an unmitigated virus will spread within a population, and explains the basis behind different containment methods used to prevent a widespread infection. Inspired by this approach, we coded a Viral Transmission Education Simulator (VTES) - a reduced representation simulation that replicates the effects of different variables on viral transmission in a population.

VTES was designed to be modular, simple, and clear, to encourage users to interact with the script that creates the simulations. By interacting with VTES, and enabling visualization of the results, we aim to help individuals understand key variables that influence the course of the spread of the disease, including population density, population mobility, interventions that reduce transmission, and a disease's mortality rate. In this article, we show the impact of decreasing transmission chance (achieved through actions such as social distancing, masking and self-quarantining), the process of

becoming "immune" to SARS-CoV-2, and how to calculate R0. We hope VTES will help foster a deeper understanding of the parameters underpinning the spread of a virus such as SARS-CoV-2 and will be used to clearly educate those unfamiliar with such parameters, as well as in exploring how small changes to variables that influence virus spread can have large impacts on the course of a pandemic.

Methods Implementation

The disease spread simulation script was written in Python 3 using standard libraries. The code was written in a way so that it is straightforward for programming novices to change the values for variables that influence the disease spread, and was written in a modular fashion with sections, each headed by a title and description in comments. Below we provide an explanation and describe each section of the code.

The script first introduces a function to **createperson**. This defines a dictionary of attributes for a "person", or dot, and how they move around in the simulated environment. They begin as alive, uninfected, and not immune (i.e. susceptible to infection).

The next section introduces **variables that influence an outbreak**. These lines are what we suggest modifying to explore the effects of different variables on viral transmission in the simulated population. For example, by changing *transmission_chance* = 0.90 to *transmission_chance* = 0.70, the simulation will decrease the probability the disease is transmitted from person to person. This will result in an decreased number of "infected people" overall.

The next section **creates a window**, forms an environment for the simulation, generates a population, and begins the simulation by changing one "dot's" status to *person ['infected'] = True*. The code **begins a loop** to move dots around according to their velocity, and bouncing off encountered walls or other dots. The parameter *velocity* determines how quickly the dots move. Users may change the velocity to simulate how measures like social distancing impact viral transmission.

Within the loop, the section headed **make a person better over infection duration** determines whether or not a person recovers over the duration of infection (thus becoming non-susceptible), and if so, reassigns their status accordingly.

The next section **check the interactions** of all the dots at every step in the simulation. This is where transmission is determined; as dots collide, their probability of becoming infected is determined by *transmission_chance*. Deceased and immune (recovered/non-susceptible) dots are ignored by infectious dots, and the window is then updated.

The final section **calculates statistics** and plots the figures (see example in [Figure 1](#)). Statistics include the total number of cases, people infected, the total number of deceased, and the length of the pandemic in days.

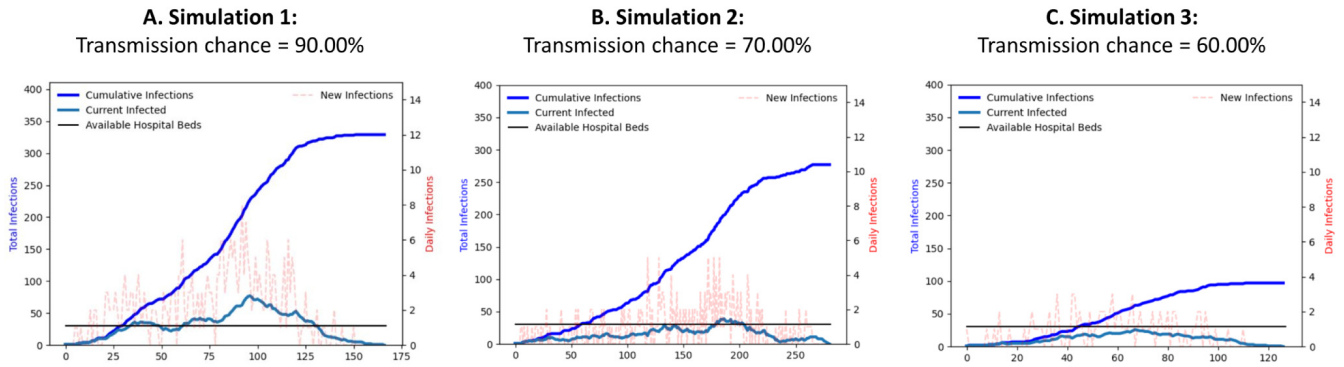


Figure 1. Plots outlining the epidemiological statistics output at the end of (A) Simulation 1, (B) Simulation 2, and (C) Simulation 3.

Operation

The code is fully executable in a Python 3 environment with the following packages installed: turtle, sys, random, math, and matplotlib. For those who do not have Python downloaded, we encourage the use of [https://repl.it/ languages/python3](https://repl.it/languages/python3), a free, online Python interpreter that can run the stimulation simply by copying and pasting the code into the command line in the browser.

VTES is designed such that the user can test the influence of various factors on the spread of the virus, such as the following: masking, population density, population mobility, and calculating R0. Below we describe methods to observe the influence of these factors on the spread of the virus in the simulated population.

The impact of masking

Since SARS-Cov-2 is spread by respiratory, air-borne droplets, wearing a mask serves to protect the person wearing the mask (by filtering out viral particles) and others (by trapping viral particles)⁶. This serves to reduce the chance of transmission by reducing viral load. By modulating *transmission_chance* users can observe the impact of how measures that aim to reduce transmission-such as wearing a mask- impact the spread of a virus.

The impact of population density

High density populations experience greater contact rates between individuals than low density populations, leading to more severe and pervasive outbreaks in high density populations⁷. For example, an outbreak on a cruise ship with a population density four times higher than that of Wuhan, China (the origin of the SARS-Cov-2 outbreak) also reported quadrupled R0 values^{7,8}. Users can observe the impact of population density on viral transmission by modulating *npeople*.

The impact of social distancing

One powerful method of controlling the spread of a disease is by social distancing or avoiding contact with other people. In the simulation, this can be thought of as limiting the number of collisions of the bouncing dots. Users can simulate the impact of social distancing in the code by reducing the *maxvelocity* variable.

Visualizing how different variables flatten the curve

Guidelines such as social distancing and wearing a mask are attempts to slow viral spread within a population - referred to as “flattening the curve”⁹. The aim of flattening the curve is to reduce the stress on healthcare systems, namely their ICUs (Intensive Care Units) through the over-admittance of severely ill individuals. Hospitals can more effectively respond to cases distributed over time, as opposed to a spike in cases that may overwhelm the healthcare system⁹. We model the capacity of our simulated population’s healthcare system via the *hospital_beds* parameter. This threshold is included in the figure output after the simulation as a solid line **Figure 1**. At times where the number of *currentinfected* exceeds the threshold, users can infer that the simulated population has more cases than their healthcare system can handle. We have set *hospital_beds* = 30; i.e., there is roughly 1 bed for every 13 people, or roughly 77 per 1000. This is much higher than most developed countries: Japan had 13 beds per 1000 people as of 2017, and the United States had 2.77 beds per 1000 people as of 2016¹⁰. We encourage users to change the number of hospital beds available to view how flattening a curve may keep the current number of infections lower than the number of available beds.

Calculating R0

R0 describes the number of secondary cases transmitted from a single infected person in a virus-naive population. For example, if R0 = 2, and one person becomes infected, it is expected that person will spread the disease to 2 more people on average. The equation to calculate R0 is

$$R0 = \tau \cdot \bar{c} \cdot d$$

where τ is transmissibility, \bar{c} is the average rate of contact between infected and non-infected people, and d is the duration of infections (the length of time someone is able to communicate an infection)¹¹. This may be written as:

$$R0 \propto \left(\frac{\text{infection}}{\text{contact}}\right) \cdot \left(\frac{\text{contact}}{\text{time}}\right) \cdot \left(\frac{\text{time}}{\text{infection}}\right)$$

We can directly calculate R0 using VTES. τ is simply the value corresponding to the *transmission_chance* variable and d is the *infectionduration* variable. Since \bar{c} is the average

contact rate, it needs to be calculated by running the simulation multiple times. This can be accomplished by setting *Record_contacts = True* in the code. Each run will then calculate and output *Contact rate for uninfected individuals with patient zero*. We recommend running the simulation >15 times and averaging the resulting contact rates to produce \bar{c} . This results in the following equation:

$$R0 \approx (transmission_chance) \cdot (\bar{c}) (infection_duration)$$

R0 is an average representation of how transmittable the virus is. In real populations, calculating R0 is complicated, and requires record-keeping measures such as contact tracing¹². Furthermore, it does not account for variance in transmission, where some “super spreaders” are responsible for spreading the disease more prolifically than the average infected person¹². Yet the strength of VTES lies in the ability to control these factors. We can confidently calculate a precise R0 using VTES and it serves as a powerful teaching tool for understanding the meaning of R0 and the variables that influence it.

Use Case 1: observing the effects of modulating VTES’s parameters

We now introduce an example of how to interact with the simulation’s code, and detail the resultant effects it has on the simulation of viral transmission. We will modulate the *transmission_chance* from *transmission_chance = 0.90* to *transmission_chance = 0.70* and *transmission_chance = 0.60*, reducing the probability of transmission by 20% and 30%, respectively. On average, decreasing the chance of transmission will reduce the total number of both people infected and fatalities resulting from infection, as well as flatten the curve.

Results

We have provided a video of Simulation 1 (*Extended data*¹³) to provide an example of what the simulation looks like.

In Simulation 1, where *transmission_chance = 0.90*, the pandemic lasted a total of 167 days, infecting 87.73% of the 375 simulated “people”, leaving 5.07% deceased **Figure 2(A)**. In Simulation 2, where *transmission_chance = 0.70* **Figure 2(B)**, the pandemic lasted 282 days, and resulted in a lower total number of people infected (73.87%) or deceased (3.73% of the simulated population). Simulation 3, where *transmission_chance = 0.60*, the pandemic lasted 128 days and displayed the lowest total number of people infected (25.87%) and deceased (1.87%) **Figure 2(C)**.

Decreasing the chance of transmission had an inverse relationship between outbreak severity and the size of the curve. The higher the chance of transmission, the more people became infected, resulting in a curve that extended beyond the hospital capacity. As transmission chance lowered, so did the number of people infected or killed by the virus, resulting in a flattened curve **Figure 1**.

The difference in the progression of the virus may be seen through snapshots of the Simulation 1 and Simulation 2’s progression. At first, both Simulation 1 and 2 started with one person infected **Figure 3(A)**. However, 25 seconds into the pandemics, we can see Simulation 1 has more “people” who can continue spreading the virus, whereas Simulation 2 has fewer actively infectious “people” and more recovered\immune people **Figure 3(B)**.

We can see examples of herd immunity in both Simulation 1 and Simulation 2 **Figure 4**. Herd immunity occurs when a large enough proportion of the population has acquired individual immunity to the virus¹⁴. Since a virus cannot propagate through a non-susceptible population (in the case of SARS-CoV-2), herd immunity creates a buffer for those who are not yet immune, essentially protecting them from infection¹⁴. This can be seen

	Total number of people infected	Total number of people deceased	Length of pandemic (days)
A. Transmission chance = 0.90	329/375 or 87.73%	19/375 or 5.07%	167
B. Transmission chance = 0.70	277/375 or 73.87%	14/375 or 3.73%	282
C. Transmission chance = 0.60	97/375 or 25.87%	7/375 or 1.87%	128

Figure 2. Statistics generated through running (A) Simulation 1, (B) Simulation 2, and (C) Simulation 3.

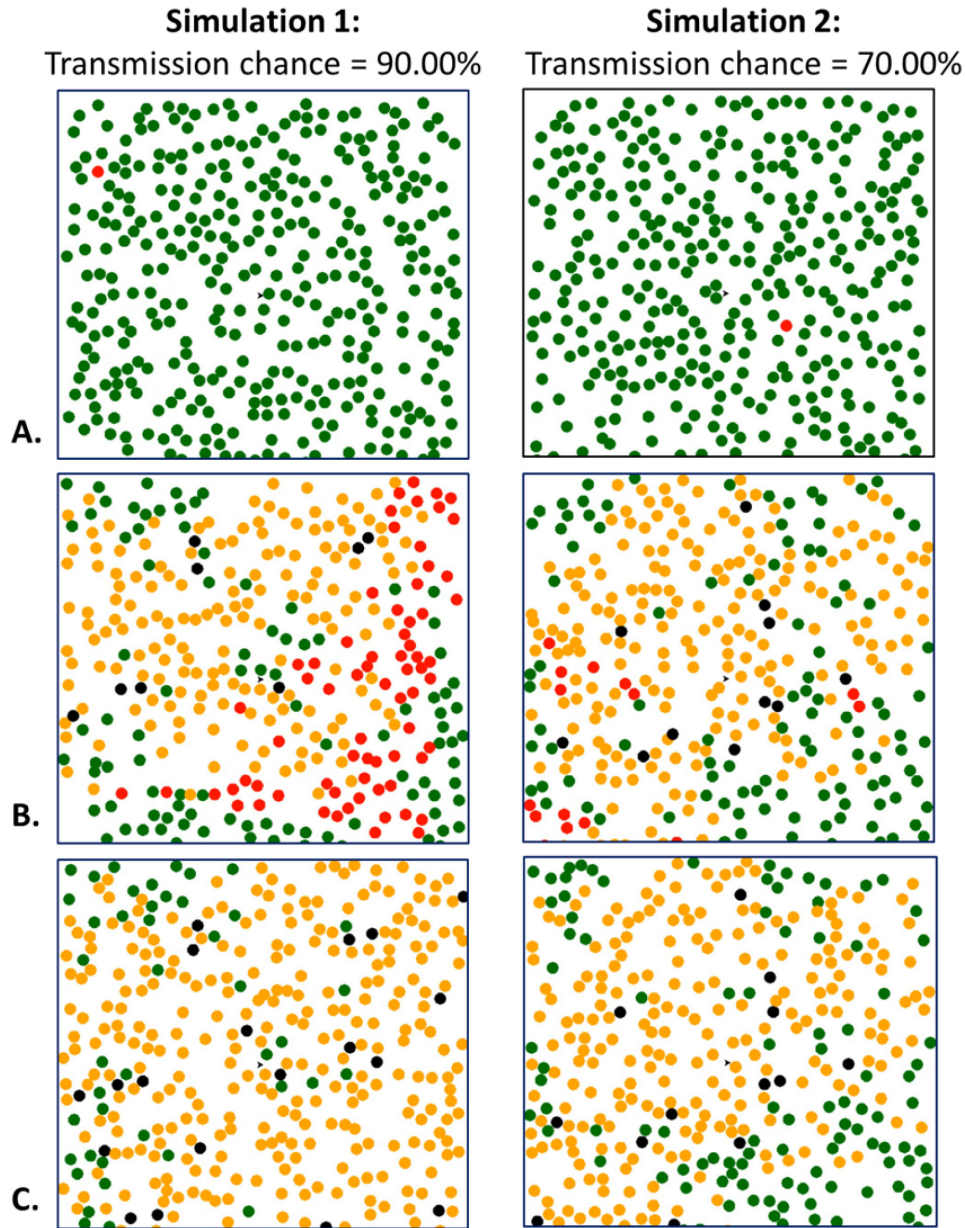


Figure 3. Snapshots of Simulation 1 and Simulation 2 at the start of the simulation (A), 25 seconds in to the simulation (B), and at the end of the simulations (C). Total run time was 42 seconds for Simulation 1 and 35 seconds for Simulation 2. Green dots represent those who have not been infected yet (susceptible individuals), red dots indicate infected “people”, yellow dots indicate the number of recovered or immune “people” (non-susceptible individuals), and the black dots represent those deceased individuals.

in Figure 4, where individuals susceptible to the virus do not come into contact with infected peoples capable of transmitting the disease, thanks to the buffer created by non-susceptible others.

Use Case 2: calculating R0

In order to calculate R0, we set *record_contacts = True*. We then ran the simulation 15 times, recording each output, the *Contact_rate_for_uninfected_individuals_with_patient_zero*

and averaging them to produce \bar{c} . This resulted in $\bar{c} = 0.1808$. With *transmission_chance = 0.9* and the *infection_duration = 14*, we show

$$R0 \propto (0.9) \cdot (0.1808) \cdot (14) = 2.27$$

This is slightly lower R0 for SARS-Cov-2 as calculated in a review by Liu *et al*, who reported a mean R0 of 3.28 and a median of 2.79¹⁵.

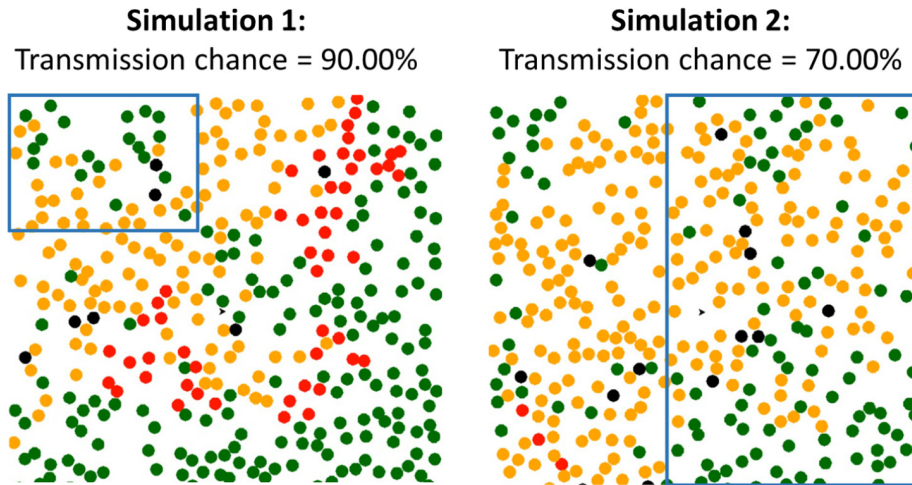


Figure 4. Example of susceptible individuals protected by those individuals that have 'recovered', thereby illustrating the process of herd immunity in the simulation (in blue border).

Discussion

In Use Case 1, we show that lowering the *transmission_chance* results in fewer numbers of infected and deceased individuals in a population experiencing a viral outbreak. By modelling a reduction in transmission chance, we can draw parallels to real-world efforts to reduce the chance of transmission, such as social distancing or wearing masks. A recent meta-analysis by Chu *et al* (2020), aimed at investigating the optimal distance for limiting person-to-person transmission and the impact of wearing a mask on preventing transmission of CoV-1, CoV-2 and Middle East Respiratory Syndrome (MERS-CoV), found that wearing a mask may lower the chance of viral transmission and infection by 14.3% and that the risk of being infected (i.e. chance of transmission) is only 3% when 2 individuals are over 1 metre apart¹⁶. These real-world studies suggest statistics that can be used to alter the parameters of the simulation, where users can observe the effects of these alterations, and more importantly, the behaviours or processes that underpin them.

In Use Case 2, we calculated R_0 and found it to be reasonably close to the established R_0 for SARS-Cov-2, especially considering we did not intentionally nor precisely modify the parameters to reflect SARS-Cov-2. We note there are numerous means of calculating R_0 at various time-points throughout the progression of a viral outbreak, with each equation building on the complexity of the last. Our definition was drawn from Jones¹¹. Furthermore, calculating R_0 in the real world via contact tracing may not always reflect the precise infectivity of a virus¹². However, we have the luxury of running the simulation multiple times in order to obtain an average \bar{c} , providing a more robust R_0 calculation. The strength of VTES is that it simplifies messy, complex, real-world variability. Our reduced representation can help users gain a deeper understanding of

R_0 - a popular, though not well understood, metric used to highlight the severity of the current SARS-Cov-2 pandemic, as well as other communicable diseases.

VTES clearly demonstrates the link between transmission control measures (e.g. wearing a mask) and its effects on slowing viral spread and reducing the number of infected individuals (Figure 2). By modifying one of the simulations parameters, we can compare the behaviour of VTES's simulations to current literature on SARS-CoV-2. This reduces the complexity of the real world, where dozens of factors influence the outcomes of viral transmission, to just a few key factors. Manipulating these factors allows users to draw insights into the mechanisms underpinning disease transmission. The dynamic nature of these simulations may therefore carry greater educational impact than impersonal and poorly described statistics. The simulations generated by VTES serve as tools that can help educate, train, and inform individuals on how a disease can propagate through a population.

While VTES models the parameters that underpin the dynamics of a pandemic at a high level, they do not capture many of the variables that influence disease spread in real populations. For instance, while the bouncing circles resemble the interaction between individuals, they do not fully, nor accurately, represent the way in which humans move about in a population. The software for VTES was created as a simple, modular tool that models the dynamics of viral transmission within a population, and encourages modification by users. The lessons learned from the simulations may be scaled up and are applicable to a larger population, and we hope future users who interact with the code add their own improvements. This may generate new and exciting insights that our simple, baseline simulation may not provide.

Many other features could be added by the adventurous programmer. For instance, one aspect of disease control that we currently do not account for is testing and quarantining. It would be straightforward to simulate the process of quarantining by adding to the code such that after a certain number of days of infection an infected individual would stop moving, and would thus have fewer collisions, and therefore chances to spread the disease.

Conclusions

In summary, we have used the programming language Python to build a stochastic, epidemic simulator that has dual roles as a tool for modeling the impact of different factors on the spread of disease and as an educational tool. By providing Use Cases where the *transmission_chance* parameter was modified and R_0 was calculated, we provided examples of the sort of exploratory use we encourage from future users, and how this tool can be used for educational purposes. We hope VTES will be amended and improved per the needs of each user, and that the simulations it generates provide visual explanations for the efficacy of, and rationale behind, varying methods of slowing viral spread.

Software availability

Source code available from: <https://github.com/sstagg/disease-transmission>

Archived source code as at time of publication: <http://doi.org/10.5281/zenodo.4041821>¹⁷

License: CC-BY 4.0

Data availability

Underlying data

All data underlying the results are available as part of the article and no additional source data are required.

Extended data

Zenodo: Video1, <http://doi.org/10.5281/zenodo.4043361>¹³.

This project contains a video, showing what the simulation produced by VTES looks like.

Data are available under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/) (CC-BY 4.0).

References

- Dong E, Du H, Gardner L: **An interactive web-based dashboard to track COVID-19 in real time.** *Lancet Infect Dis.* 2020; **20**(5): 533–534.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Coronavirus resource center, 2020.** [Online; accessed 09-September-2020].
[Reference Source](#)
- Kasprak A: **Will sipping water every 15 minutes prevent a coronavirus infection?** 2020.
[Reference Source](#)
- Wu C: **Anti-mask activists protest in florida as covid-19 cases continue to soar.** 2020.
[Reference Source](#)
- Stevens H: **Why outbreaks like coronavirus spread exponentially, and how to “flatten the curve”.** *The Washington Post.* 2020.
[Reference Source](#)
- Leung CC, Cheng KK, Lam TH, *et al.*: **Mask wearing to complement social distancing and save lives during COVID-19.** *Int J Tuberc Lung D.* 2020; **24**(6): 556–558.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Rocklöv J, Sjödin H: **High population densities catalyse the spread of COVID-19.** *J Travel Med.* 2020; **27**(3): taaa038.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Rocklöv J, Sjödin H, Wilder-Smith A: **COVID-19 outbreak on the Diamond Princess cruise ship: estimating the epidemic potential and effectiveness of public health countermeasures.** *J Travel Med.* 2020; **27**(3): taaa030.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Specktor B: **Coronavirus: What is ‘flattening the curve,’ and will it work.** *Live Science.* 2020.
[Reference Source](#)
- Wikipedia contributors: **List of countries by hospital beds — Wikipedia, the free encyclopedia, 2020.** [Online; accessed 28-August-2020].
[Reference Source](#)
- Holland J: **Notes on r_0 .** *California: Department of Anthropological Sciences.* 2007; **323**: 1–19.
[Reference Source](#)
- Breban R, Vardavas R, Blower S: **Theory versus data: how to calculate R_0 ?** *PLoS One.* 2007; **2**(3): e282.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Kurtin D: **Video1 (Version v1.0).** *Zenodo.* 2020.
<http://www.doi.org/10.5281/zenodo.4043361>
- Randolph HE, Barreiro LB: **Herd Immunity: Understanding COVID-19.** *Immunity.* 2020; **52**(5): 737–741.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Liu Y, Gayle AA, Wilder-Smith A, *et al.*: **The reproductive number of COVID-19 is higher compared to SARS coronavirus.** *J Travel Med.* 2020; **27**(2): taaa021.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chu DK, Akl EA, Duda S, *et al.*: **Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis.** *Lancet.* 2020; **395**(10242): 1973–1987.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- sstagg, Kurtin D: **sstagg/disease-transmission: VTES: Viral Transmission Education Simulator (Version v1.0).** *Zenodo.* 2020.
<http://www.doi.org/10.5281/zenodo.4041821>

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research