

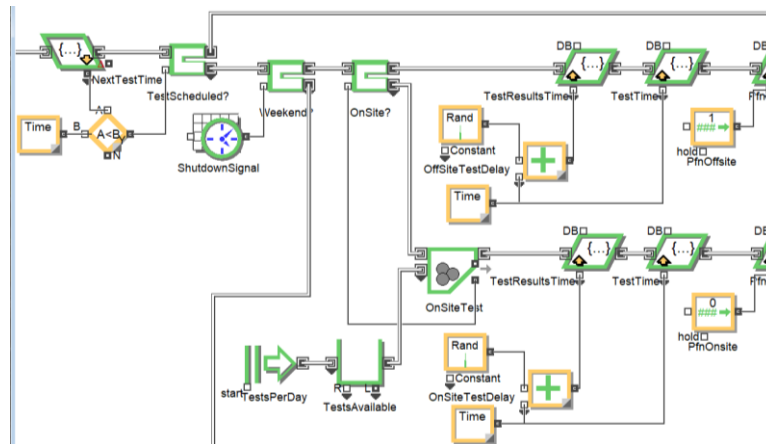
Agent-based Disease Surveillance and Transmission Model

Thomas Edmunds*
COVID Systems Analysis Group

LLNL-TR-829804

December 2021

*edmunds@llnl.gov, (925) 423-8982



Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

Acknowledgements

The author would like to acknowledge the review and recommendations of the other members of the COVID Systems Analysis Group at LLNL: Thomas Bates, Matthew Dombroski, and Richard Wheeler.

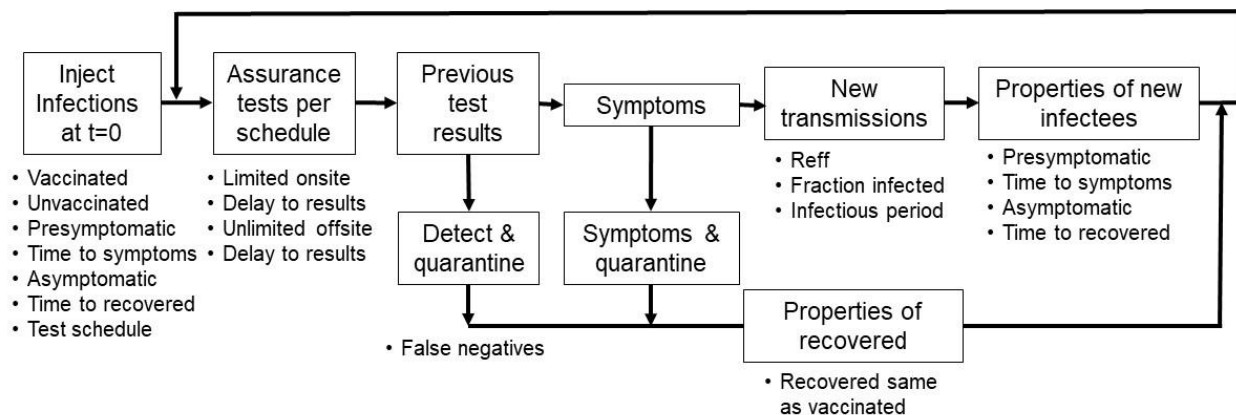
Executive Summary

The SARS-CoV-2 pandemic has driven a need for models that can predict transmission among populations and test the effectiveness of proposed interventions to control it such as testing, quarantining, vaccinations, and nonpharmaceutical interventions. Some questions of interest include:

1. How rapidly will the virus spread if no action is taken?
2. How frequently should people be tested and quarantined if necessary?
3. Should unvaccinated persons be tested more frequently than the vaccinated?
4. If test results are delayed, how will that affect the spread?
5. How do test false negative rates affect the overall efficacy of testing programs?
6. How do constraints on testing resources impact the overall efficacy of testing programs?
7. How does the patient's failure to recognize COVID symptoms and quarantine affect the spread?
8. What level of vaccination coverage is needed to suppress the epidemic?
9. What is the risk of a large outbreak (probability exceeding x infections)?
10. What is the cost effectiveness of interventions (dollars per avoided infection)?

This report describes the structure of a discrete event simulation model that provides actionable information to address these questions. Parameter inputs and results of analyses have been reported in briefings provided to LLNL management and are not discussed here.

The overall flow of agents in the model is shown in the figure below. Key variables that are initialized or used at each of the process steps are noted. As indicated on the left side of the figure, infected persons are inserted into the overall population at time zero. Next, assurance tests are conducted per the individualized schedule of each person in the model. If test results from previous days are available, individuals who tested positive are quarantined. Next, individuals who present symptoms and recognize them as COVID-19 are quarantined. Finally, new transmissions are calculated and the properties of the newly infected are initialized. As indicated at the far right of the figure, all persons in the model begin the next 24-hour period.



[This page intentionally left blank.]

Table of Contents

1	Background and Model Overview.....	1
2	Injection of Subpopulations	3
3	Assurance Testing	5
4	Test Results and Quarantine	7
5	Symptomatic and Quarantine.....	8
6	Transmission of New Infections.....	9
7	Control Panel for Running Model.....	11
8	Sample Results	13
	References	15
	Glossary of Icons Used in the Model.....	16

List of Figures

Figure 1 Overall flow of agents in model	2
Figure 2 Times of key events in model.....	2
Figure 3 Insertion of four subpopulations and parameter assignments.....	4
Figure 4 Resource-limited onsite and unlimited offsite testing	6
Figure 5 Status when sampled and scheduling next test for unvaccinated and vaccinated	7
Figure 6 Reading test results, quarantine, and computing fraction of population infected	8
Figure 7 Quarantine based on test results or recognition of symptoms	9
Figure 8 Disease transmission among vaccinated and unvaccinated subpopulations	10
Figure 9 Details of infection probability calculation for unvaccinated population.....	10
Figure 10 Control panel for specifying inputs and running model	12
Figure 11 Box and whiskers plot of simulation results	13
Figure 12 Cumulative distribution of infections.....	14

1 Background and Model Overview

The SARS-CoV-2 pandemic has driven a need for models that can predict transmission among populations and can test the effectiveness of proposed interventions to control it such as testing, quarantining, vaccinations, and nonpharmaceutical interventions. Some questions of interest include:

1. How rapidly will the virus spread if no action is taken?
2. How frequently should people be tested and quarantined if necessary?
3. Should unvaccinated persons be tested more frequently than the vaccinated?
4. If test results are delayed, how will that affect the spread?
5. How do test false negative rates affect the overall efficacy of testing programs?
6. How do constraints on testing resources impact the overall efficacy of testing programs?
7. How does the patient's failure to recognize COVID symptoms and quarantine affect the spread?
8. What level of vaccination coverage is needed to suppress the epidemic?
9. What is the risk of a large outbreak (probability exceeding x infections)?
10. What is the cost effectiveness of interventions (dollars per avoided infection)?

There are two general approaches for modeling disease transmission. One approach is to formulate a set of coupled ordinary differential equations (ODEs) representing subpopulations. The population is typically divided into four groups: susceptible, exposed, infected, and recovered (an SEIR ODE model). The four coupled differential equations are integrated to predict the spread of the disease [Jason 2020]. This modeling approach assumes homogeneous mixing of all four subpopulations. Given this assumption, it is difficult to represent the transient super spreader events that have characterized the COVID-19 pandemic.

A second general approach is to develop an agent-based model (ABM). ABMs represent each individual in the population as an independent entity that interacts with all other agents. As such, detailed information about each person's behavior at all times is required to instantiate and run the model. Contact frequency, duration, proximity, and other detailed data are required by the model. This need for detailed information about the behavior of each agent is a challenge. An ABM of the City of Chicago was recently developed to predict the spread of COVID-19 [ANL 2021].

This report describes a hybrid approach that is mostly agent based, but utilizes R_0 and time frames from ODE models to estimate infection rates and reduce data requirements. This paper describes the structure and mechanics of the discrete event simulation model. The model has evolved over time to address different questions posed to the analysis team at LLNL. This paper describes the November 23, 2021 version of the model. Parameter inputs and results of analyses have been reported in briefings provided to LLNL management and are not covered in this report.

The overall flow of agents in the model is shown in **Figure 1**. Key variables that are initialized or used at each of the process steps are noted. As indicated on the left side of the figure, infected

persons are inserted into the overall population at time zero. Vaccination status, characteristics of infection, testing schedule, and other attributes are generated and attached to each individual.

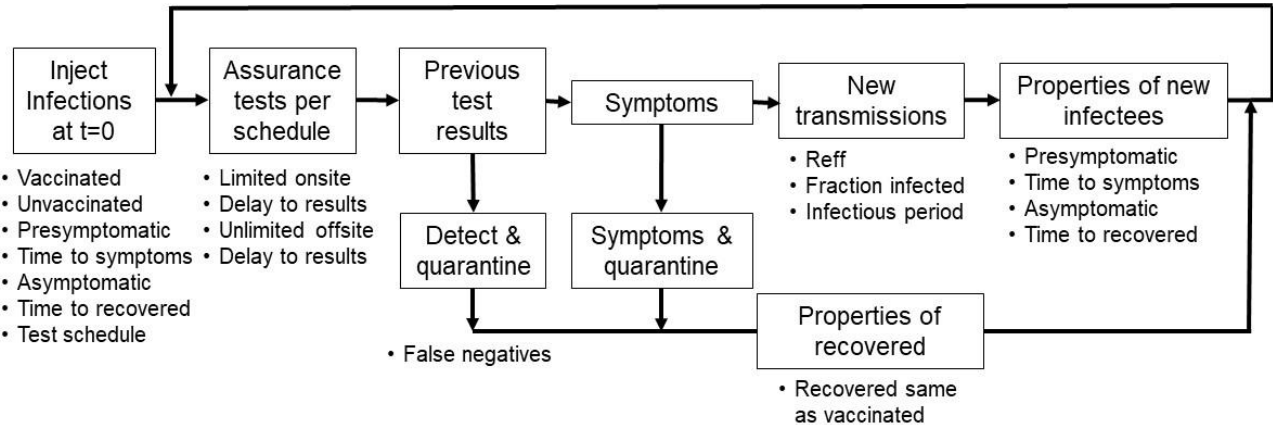


Figure 1 Overall flow of agents in model

Next, tests are conducted per the individualized schedule of each person in the model. First, limited onsite tests are allocated to individuals who are scheduled to be tested. Additional tests are conducted using offsite resources. The time of sample collection and delays for onsite and offsite test results are used to compute when test results will be available. If test results from samples taken on previous days are available, individuals who tested positive are quarantined. In addition, individuals who have presented symptoms recognized them as COVID-19 are quarantined. Finally, new transmissions are calculated and the properties of the newly infected are initialized. As indicated at the far right of the figure, all persons in the model begin the next 24-hour period.

The times of some key events in the model are shown in **Figure 2**. Sample collection occurs at 8:00 am and results from previous tests are checked at that time. If a previous test result is positive, the employee is immediately quarantined and does not enter the work force that day. The tally of infected and uninfected in the work force not in quarantine is conducted at noon. At midnight, each person checks for COVID-19 symptoms and enters quarantine if they are present. Also, new infections are computed at midnight. As indicated in the figure, the fraction of the population that is infected at noon is a key driver of the new infections at the end of the day.

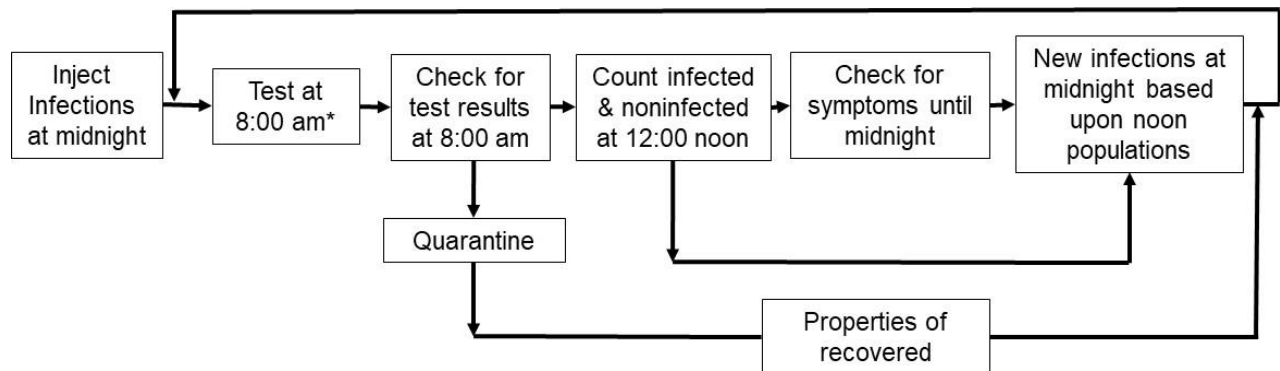


Figure 2 Times of key events in model

*Test results are not available until next day or later.

2 Injection of Subpopulations

It can be a challenge to define appropriate boundary conditions for a simulation model. In this case, the boundary between the population working on the LLNL site and the surrounding community must be defined. This is accomplished by seeding the model with infections from the external community. There are two approaches to seeding the model with infected individuals.

One approach is to inject infected individuals only at time zero, then let the disease spread among the simulated population with no further infections introduced. Disease spreads among the modeled population based on a 24-hour clock.

Computation of the number of infected persons to inject into the LLNL population at time zero can be based on the number of daily reported cases at the LLNL site at time zero. Note that the one-time injection must consider the number of reported cases at time zero and number of infectious cases that have yet to present symptoms or are asymptomatic but are “in the pipeline.” For example, assume a two day infectious period for presymptomatic patients, a seven day infectious period for asymptomatics, 50% of infected persons are asymptomatic, and one reported case per day. With these assumptions the computations proceed as follows:

- (1 reported case) + (1 presymptomatic-infectious case in the pipeline) = 2 infectious-presymptomatic cases
- (1 reported case) x (1 asymptomatic per presymptomatic) + (6 asymptomatic infections in the pipeline) = 7 infectious-asymptomatic cases

Because infection of an unvaccinated person is about ten times more likely than a vaccinated person, one could assume that all of the initial infections are in unvaccinated persons. However, if the vaccinated cohort is much larger than the unvaccinated cohort, the initial infections may be more likely to be present in the vaccinated cohort due to its size. Under these circumstances, the initial infections could be allocated to the vaccinated and unvaccinated in proportion to the size of the two cohorts and the relative risk of infection.

A second approach is to continuously inject infected persons based upon reported cases at the simulated site. Under this approach, the ratio of asymptomatic to symptomatic infections would be applied to the reported number of cases per day and that number would be injected each day of the simulation. In the example above with one reported case, two cases per day would be injected into the simulation each day – one presymptomatic and one asymptomatic. Other assumptions could be used.

A screen capture of a portion of the discrete event simulation model¹ is shown in **Figure 3**. The software used to build the model provides a graphical user interface to instantiate and connect code modules that comprise the simulation model. Infected persons can be inserted at one point in time or can be inserted each day in accordance with the procedures described above.

¹ The model was built with the ExtendSim software package <https://extendsim.com/>. The icons and code modules used to build this model are described in the glossary at the end of the paper.

As indicated in the figure, four subpopulations of individuals comprise the population simulated: 1) unvaccinated-not infected, 2) vaccinated-not infected, 3) unvaccinated-infected-presymptomatic, 4) unvaccinated-infected-asymptomatic. The four “Create” code modules that generate these subpopulations are shown as arrowhead icons on the left-hand side of the figure.

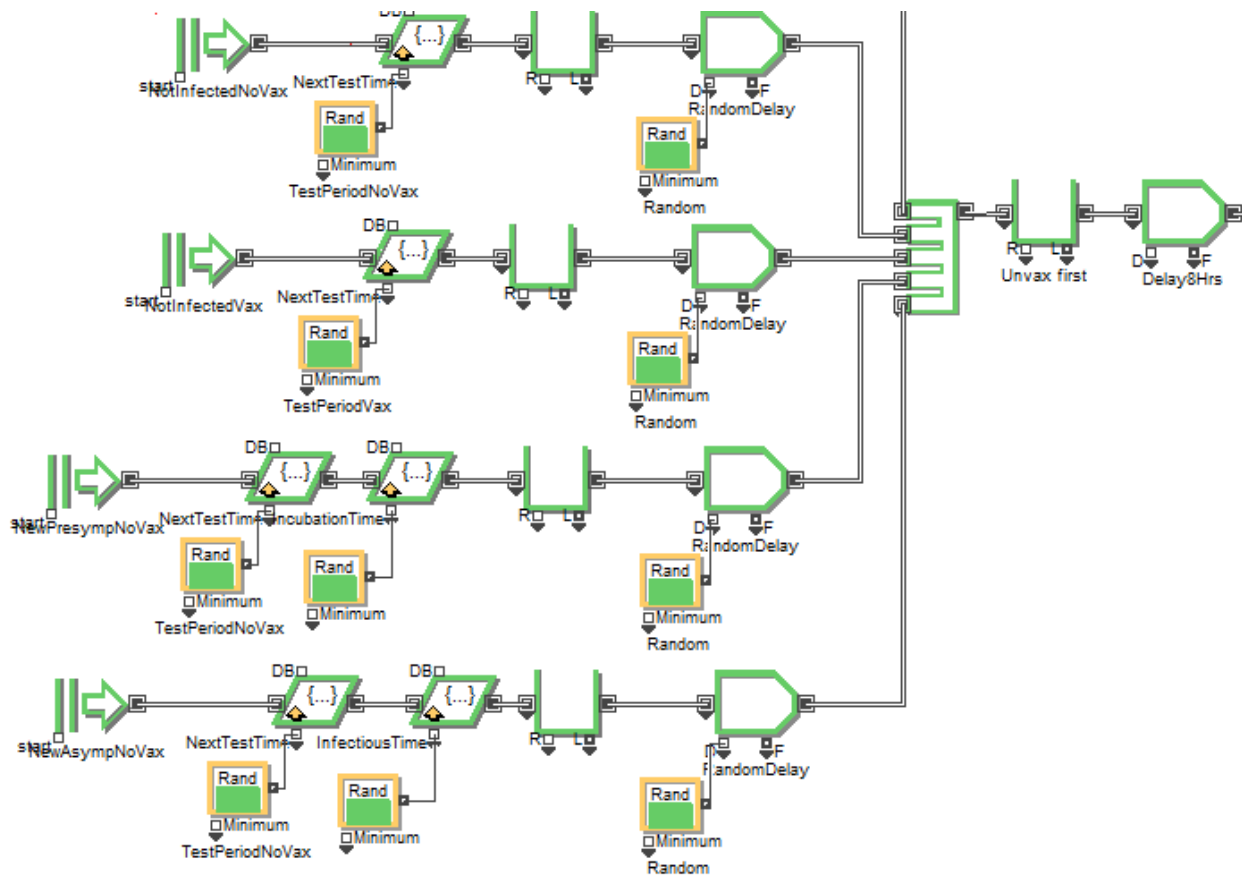


Figure 3 Insertion of four subpopulations and parameter assignments

The following attributes are assigned to each individual generated in the Create modules:

- Time – hour when person enters system (zero)
- Quantity – number of persons of given type generated (not infected-unvaccinated, not infected-vaccinated, infected-presymptomatic, infected-asymptomatic)
- Vaccinated – 1 if vaccinated, 0 otherwise
- IncubationTime – hours between infection and presentation of symptoms (also called latent period)
- InfectiousTime – number of hours an asymptomatic person is infectious
- Infected – 1 if infected, 0 otherwise
- TestResultsTime – hour when test results are available
- AwaitingResults – 1 if waiting for test results, 0 otherwise
- PreSymp – 1 if infected person is presymptomatic, 0 otherwise

Asymp – 1 if infected person is asymptomatic, 0 otherwise
 InfectedTime – hour when person is infected
 TestTime – hour when sample is taken
 Pfn – probability of false negative with test

As indicated at the next icon downstream from the Create node, the variable NextTestTime is assigned for each person. (A parallelogram that leans right assigns a value to a parameter.) The value for the variable NextTestTime is randomly assigned between zero and the test period for that subpopulation. For example, if individuals in the subpopulation are to be sampled once per week (168 hours), the test for individuals will be scheduled between 0 and 168 hours. This random assignment process will tend to uniformly distribute the test times over the testing interval ($1/7^{\text{th}}$ of the population will be scheduled for testing each day). When the current time is greater than the NextTestTime, the person is routed into a section of the model where testing is conducted.

Finally, there is a queue for diagnostic purposes, then a short, randomized delay of individuals so that the four input streams are uniformly mixed before entering the rest of the model. Note that this initialization process occurs only once. After this initialization, the simulation model follows a regular 24-hour cycle.

As indicated on the right-hand side of the figure, the four streams are aggregated. Immediately after the aggregation point, all individuals are placed in a queue and sorted so that the unvaccinated persons are released first. This sorting guarantees that the unvaccinated persons are first in line to be assigned available on-site test kits. Finally, all persons are delayed from time zero (midnight) until 8:00 am when samples are taken.

3 Assurance Testing

The testing program is depicted in **Figure 4**. There are four routing decisions in this portion of the model that are driven by object or system attributes. Each of these four routing actions is performed by a module that reads an attribute and transmits the value of the attribute to the routing module that immediately follows it. The routing decision, whether to take the upper or lower path, is based on the value of the attribute.

The first routing decision depends upon whether or not the individual is waiting for the results of a previous test². This is accomplished by reading the binary attribute AwaitingResults (the value is 1.0 if waiting for results and zero otherwise) and transmitting the value to the forked routing icon. If the value is 1.0, the COVID-19 test allocation structure in the model is bypassed by taking the lower line emanating from the routing icon. If the value is zero, the person is routed on the top line.

² A parallelogram that leans left retrieves a value of a parameter that has already been assigned to the agent. This value is transmitted to the routing node on the right. See glossary for a complete description of icons.

The second routing decision depends upon whether or not the person is scheduled to be tested. This is accomplished by reading the attribute NextTestTime (the hour when the test is scheduled) and transmitting the value on the lower line to decision logic represented by the diamond-shaped icon with the inequality $A < B$. As indicated in the figure, the decision logic compares NextTestTime (the A input) to the current time (the B input). If NextTestTime is less than (before) the current time, output is routed to the lower branch feeding the logic for onsite and offsite COVID-19 testing. If NextTestTime is greater (later) than the current time, output is routed to the upper line and bypasses the COVID-19 testing portion of the model.

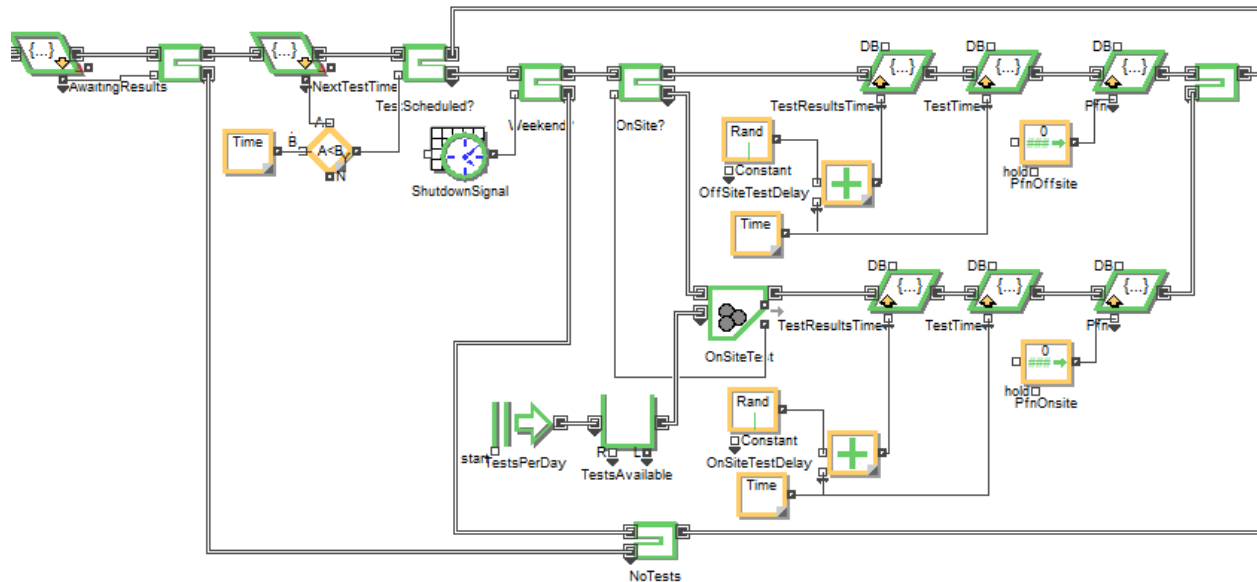


Figure 4 Resource-limited onsite and unlimited offsite testing

The third routing decision is driven by the day of the week. It is assumed that onsite and offsite testing can only be conducted on weekdays. The weekday/weekend schedule is stored in the clock icon in the figure. The node generates a value of 1.0 during the weekend. This signal is read by the routing node and all output is routed to the lower line. If it is not the weekend, output is routed to the top line where onsite or offsite COVID-19 testing is performed.

The fourth routing decision is driven by the availability of onsite testing kits. The Create node near the bottom of the figure generates a new batch of test kits each weekday, which are transmitted to a Batch node. This Batch node matches individuals to be tested with test kits. The Batch node keeps track of available test kits and transmits this number to the routing node. When the number of test kits reaches zero, the routing node switches output from the bottom line (onsite tests) to the top line (offsite tests). This logic ensures that tests are conducted using these onsite resources until the daily supply has been exhausted.

As indicated in the figure, data are recorded that specify when the sample was taken, when results will be available, and the false negative rate of the test. Parameters that are defined in this section of the model include:

ShutdownSignal – 1 if weekend, 0 otherwise (assuming no sample collection on weekends)

TestsPerDay – number of test kits available per day for onsite testing, switch to offsite testing when on site kits are no longer available, number of offsite tests is unlimited
 OnSiteTestDelay – hours between sample collection and test results reported when testing onsite
 OffSiteTestDelay - hours between sample collection and test results reported when testing offsite
 PfnOffsite – probability of false negative for offsite tests
 PfnOnsite – probability of false negative for onsite tests

After samples have been collected, the infection status of the person at the time they were sampled is recorded and the schedule for the next test is specified. **Figure 5** shows how these procedures are modeled. As indicated on the left, if the person was infected when sampled, the attribute TestedInfected is set to 1.0 on the lower branch. If not infected, the attribute is set to zero on the upper branch.

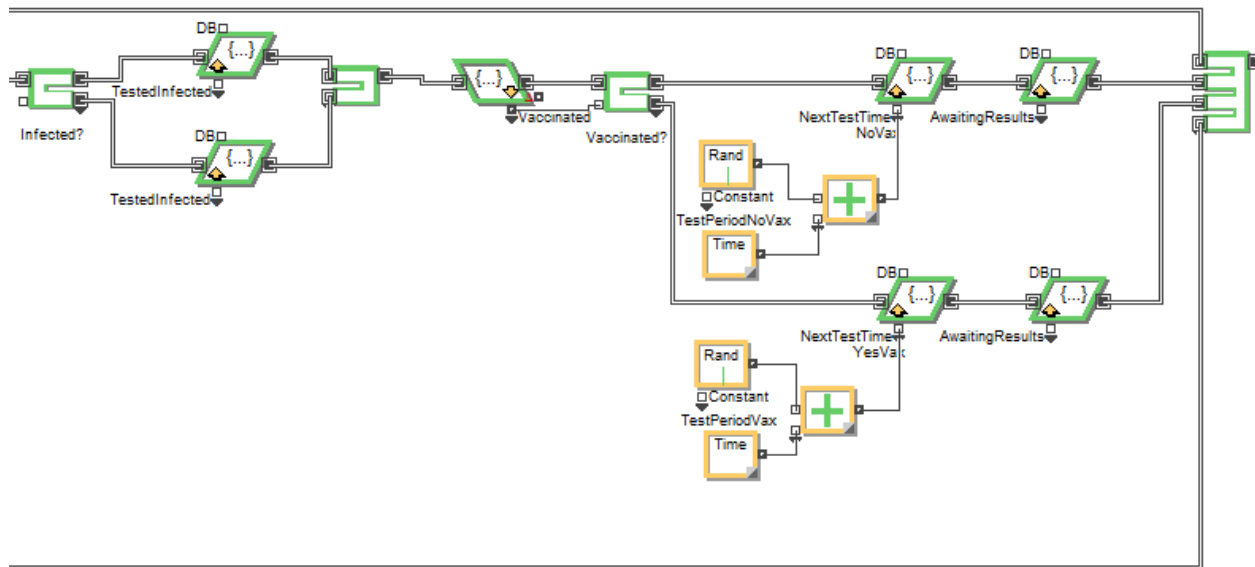


Figure 5 Status when sampled and scheduling next test for unvaccinated and vaccinated

The next test is for the vaccination status. If the person is not vaccinated, the upper branch is taken. The NextTestTime is assigned a value that is the sum of the test period for unvaccinated individuals and the current time. The AwaitingResults binary flag is set to 1 and the person proceeds to the right. If the person is vaccinated, the lower branch is taken and similar parameter assignments occur. These two streams are combined with the two bypass streams shown in the previous figure, as shown on the right.

4 Test Results and Quarantine

The model structure for recording and responding to test results is shown in **Figure 6**. As indicated in the figure, three routing decisions are performed. First, there is a check to see if test results are available. To accomplish this, the attribute TestResultsTime is compared to the current time and results are transmitted to the routing node. If TestResultsTime is less than (before) the current time, test results are available, a value of 1.0 is transmitted to the routing node, and the lower branch is taken. The AwaitingResults attribute is reset to zero.

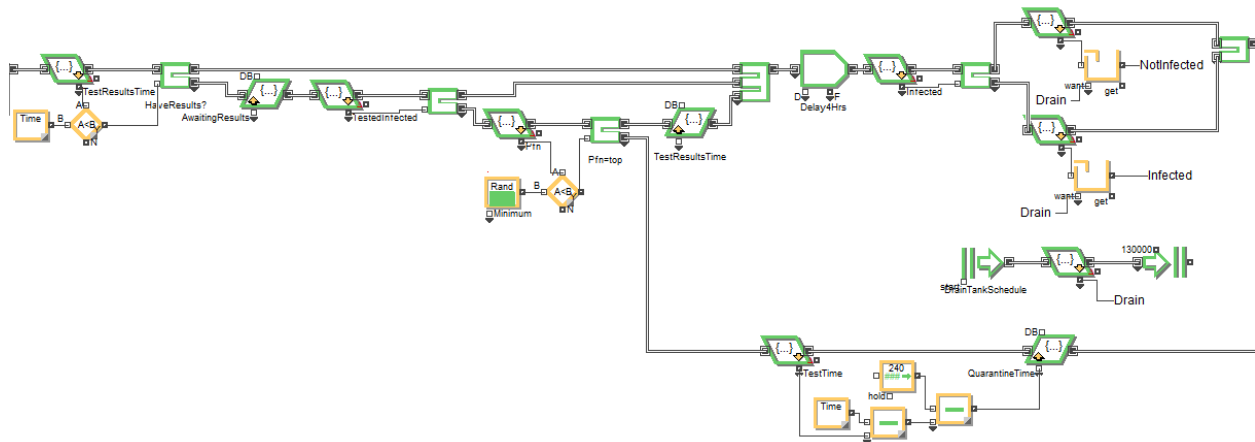


Figure 6 Reading test results, quarantine, and computing fraction of population infected

The second routing decision is driven by the infection status of the person when the sample was taken. If the TestedInfected attribute is zero, the sample was collected when the person was not infected and the upper branch is taken because test results are negative. If TestedInfected is 1.0, the lower branch is taken.

The third routing decision accounts for false negatives. As indicated in the figure, a random number uniformly distributed between zero and 1.0 is compared to the false negative rate to see if a detection occurs. If detection occurs, the person is quarantined.

The bottom branch in the figure computes the quarantine period. If test results have been delayed, the quarantine time is reduced by the elapsed time after the sample was taken. For example, if the required quarantine period is 10 days and the sample was taken 2 days ago, the person is only required to quarantine for 8 days.

The structure in the upper right of the figure computes the fraction of the population that is infected. First, the clock is advanced four hours to noon. The infection status of each person is read, and they are routed to the upper or lower branch accordingly. The numbers of uninfected and infected individuals in the population are accumulated and recorded as the variables NotInfected and Infected. These variables are used later in the model to predict transmission rates among vaccinated and unvaccinated subpopulations.

5 Symptomatic and Quarantine

In addition to detection by testing, presymptomatic persons can recognize symptoms and initiate quarantine. This logic is shown in **Figure 7**. Note that symptoms are not present until the incubation time stamp has been exceeded and that there is a non-zero probability that the symptoms will not be recognized as COVID-19 on that day. Finally, the model treats someone

who has recovered from the disease as equivalent to being vaccinated. This is performed by resetting parameters in the set attribute blocks that follow the quarantine activities.

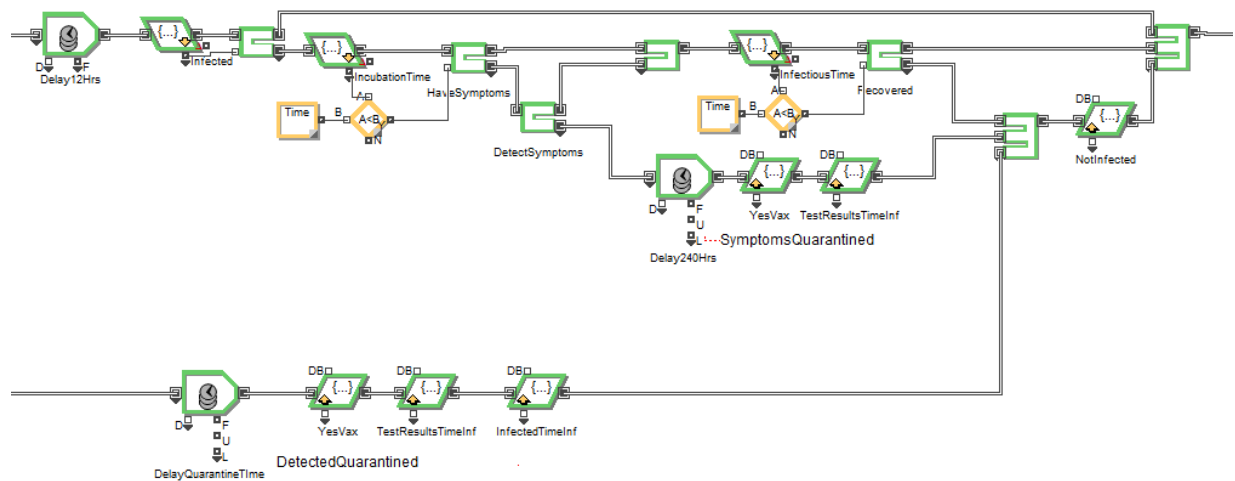


Figure 7 Quarantine based on test results or recognition of symptoms

6 Transmission of New Infections

As indicated on the left side of **Figure 8**, the population is first separated into vaccinated and unvaccinated cohorts. The unvaccinated are routed to the top branch and vaccinated persons are routed to the bottom. For each of the two cohorts of the model, the probability of transmission is computed, and a random number is compared to this probability to see if transmission occurs. Details of this process are shown in **Figure 9**.

The time of the infection is recoded as the attribute `InfectedTime`, then the person is added to the infected queue. Throughout the simulation, the cumulative number of unvaccinated persons that have been infected is output as the variable `NewUnvaxInfections`.

Following this queue, the newly infected are split into asymptomatic cases in the top line and presymptomatic cases in the bottom line using their corresponding probability. The asymptomatic cases are assigned an `InfectiousTime`. The incubation time is set to infinity because it is only a property of the presymptomatic cases in this model. Finally, the `Infected` attribute is set to 1.0 and the person is routed to the beginning of the next 24-hour simulation period. A similar process is followed for the unvaccinated and presymptomatic cases.

The vaccinated persons are routed to the bottom part of the model and processed in a similar manner. The cumulative number of vaccinated persons that have been infected is output as the variable `NewVaxInfections`.

At the end of the entire simulation run, the total number of infected individuals for the duration of the simulation is computed as the sum of `NewUnvaxInfections` and `NewVaxInfections`. This

total is written out to a file at the end of the run, as indicated at the bottom of the figure. At indicated on the right side of the figure, all streams are combined and routed to the beginning of the model shown in **Figure 3** for simulation of the next 24-hour period.

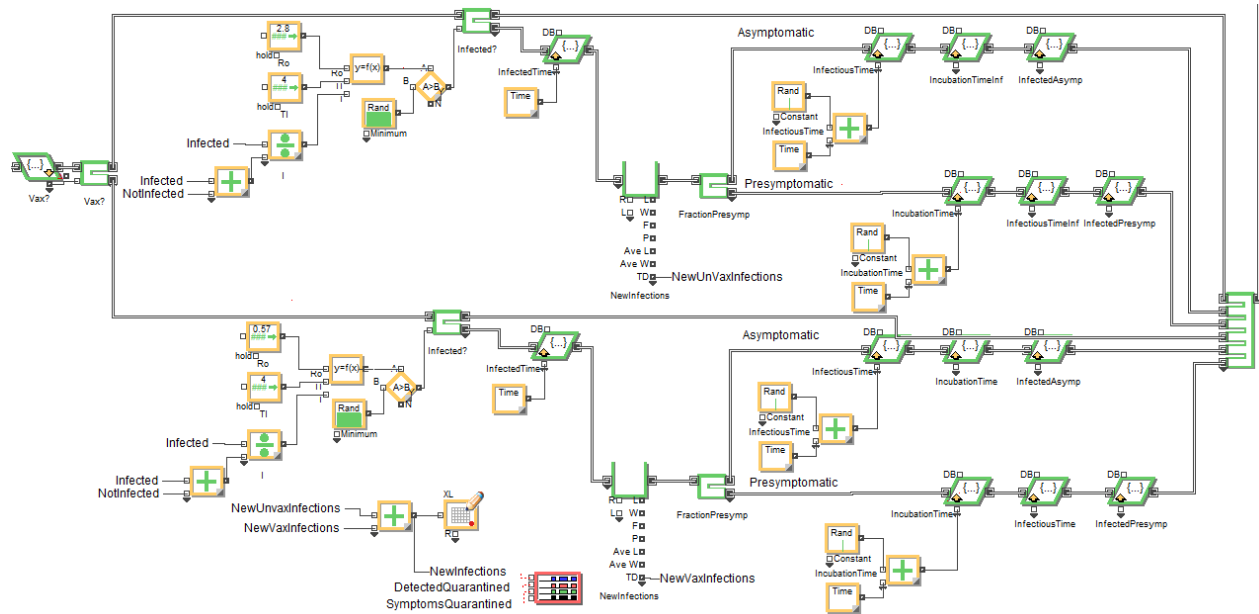


Figure 8 Disease transmission among vaccinated and unvaccinated subpopulations

Details of the process for computing the probability of infection are shown in **Figure 9**. The probability calculation is an implementation of the term in the SEIR ODE model that controls the flow from the susceptible state to the exposed state [Jason 2020]. The rate of transition from the susceptible state is computed as:

$$dS/dt = -1/T_i R_o S I$$

where

S = fraction of the population that is susceptible

T_i = time that person is infectious

R_o = reproduction number

I = fraction of population that is infectious

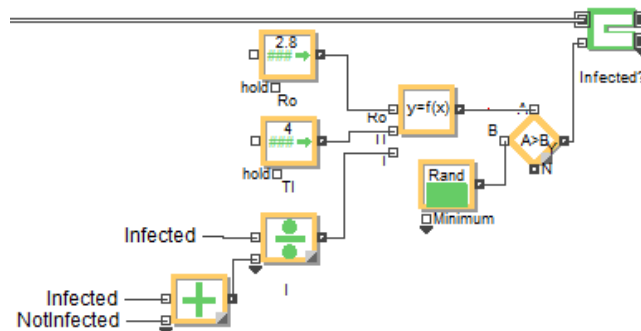


Figure 9 Details of infection probability calculation for unvaccinated population

In the model, the value used for T_i is an average of the infectious time for presymptomatic and asymptomatic persons. The R_0 value is an effective value for either a vaccinated or unvaccinated population. This equation is implemented in the “ $y=f(x)$ ” module in the simulation model.

The two R_0 parameters for vaccinated and unvaccinated populations used in the model should reflect the mobility of the local population, the effects of non-pharmaceutical interventions in place at the time, and other factors that affect the spread of the disease. Further, the two parameters need to reflect the relative transmissibility of the disease for the vaccinated and unvaccinated subpopulations. One approach for developing estimates for them is to calculate R_{vax} and R_{unvax} using the composite R_0 value for the local population, the transmissibility ratio T , and the fraction of the local population that is vaccinated f_{vax} . These conditions yield the following two simultaneous equations that can be solved for the needed parameters R_{vax} and R_{unvax} .

$$\begin{aligned} R_0 &= f_{vax} R_{vax} + (1 - f_{vax}) R_{unvax} \\ R_{unvax}/R_{vax} &= T \end{aligned}$$

Solving for R_{unvax} and R_{vax} yields the following equations in terms of the known parameters R_0 , f_{vax} , and T .

$$\begin{aligned} R_0 &= f_{vax} R_{vax} + (1 - f_{vax}) TR_{vax} \\ R_{vax} &= R_0 / (f_{vax} + (1 - f_{vax})T) \\ R_{unvax} &= TR_{vax} \end{aligned}$$

Alternative formulations of the infection process could be implemented. One simple alternative formulation is to combine the unvaccinated and vaccinated cohorts and use a weighted average value for R_0 in the calculations of infection probability. A second approach would be to use transmission probabilities for infected/vaccinated and infected unvaccinated. A two-by-two matrix of transmission probabilities per day would be required.

From	Vaccinated	Unvaccinated
Infected/vaccinated	P_{11}	P_{12}
Infected/unvaccinated	P_{21}	P_{22}

7 Control Panel for Running Model

Most of the key parameters to configure a simulation run are exposed in the control panel shown in **Figure 10**. Input in the control panel is copied to the appropriate code modules throughout the model. Note that test periods for the four subpopulations are specified for the initial insertion of individuals and for subsequent scheduled tests. In the example in the figure, the test period for uninfected, unvaccinated persons (labeled NotInfectedNoVax) initially injected into the model is randomly assigned in the range zero to 84 hours. Subsequent tests are scheduled to occur every 84 hours (3.5 days).

Notebook - COVID-19 simulator 10-26-21.mox

Test period (hours)		Epi parameters	
Initial		Unvaccinated	
NotInfectedNoVax	Scheduled	R_0 2.8	
min 0	84 Hours	T_i 4 Days	
max 84		Vaccinated	
NotInfectedVax		R_0 0.57	
min 9999	9999 Hours	T_i 4 Days	
max 9999		Output to SimOutput.xlsx	
PresympNoVax		Column # 26	TotalNewInfections
min 0	Hours	Test results delay	
max 84		Onsite 12 hours	
AsympNoVax		Offsite 12 hours	
min 0	Hours	Prob(false negative)	
max 84		Offsite 1	
Offsite		Onsite 0	
Onsite			

Figure 10 Control panel for specifying inputs and running model

Probabilities of a false negative for offsite and onsite tests are entered at the bottom left. The R_0 and T_i parameters for the SEIR model terms are shown at the top right. There are separate terms for unvaccinated and vaccinated individuals. At the end of each simulation run, the total number of persons infected is written to an Excel file named SimOutput.xlsx in column 26. Finally, the delay from sample collection to test results for onsite and offsite tests are specified.

8 Sample Results

The simulation model is intended to inform management decisions regarding testing and quarantine policies and technologies. Some key questions that the model could address include:

1. How rapidly will the virus spread if no action is taken?
2. How frequently should people be tested and quarantined if necessary?
3. Should unvaccinated persons be tested more frequently than the vaccinated?
4. If test results are delayed, how will that affect the spread?
5. How do test false negative rates affect the overall efficacy of testing programs?
6. How do constraints on testing resources impact the overall efficacy of testing programs?
7. How does the patient's failure to recognize COVID symptoms and quarantine affect the spread?
8. What level of vaccination coverage is needed to suppress the epidemic?
9. What is the risk of a large outbreak (probability exceeding x infections)?
10. What is the cost effectiveness of interventions (dollars per avoided infection)?

Results from the simulation model can be displayed in a number of different formats to inform these decisions. A standard Excel box and whiskers plot displays results of each of the simulation runs for a given configuration of the model. An example is shown in **Figure 11**.

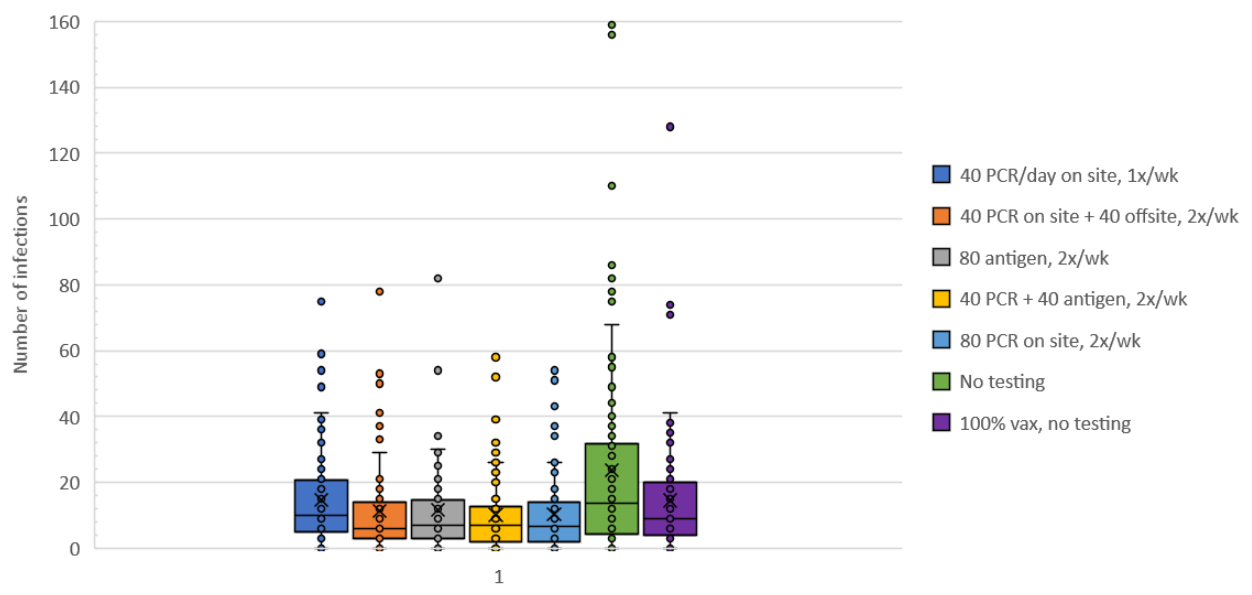


Figure 11 Box and whiskers plot of simulation results

Each point in the figure represents one realization out of 100 simulated. Risks of outbreaks exceeding a certain level can be read directly from the graphic. For example, if no tests are conducted (green bar), the risk of an outbreak exceeding 100 persons is 3%. The data are displayed as quartiles, with the middle two quartiles inside of the box. The line inside the box is the median value and the x is the mean.

Cumulative probability distribution plots provide another view of the infection risk profile. The risk of an outbreak exceeding n persons can also be read from this plot. For example, the probability of an outbreak exceeding 40 is 15% if no one is vaccinated (green curve).

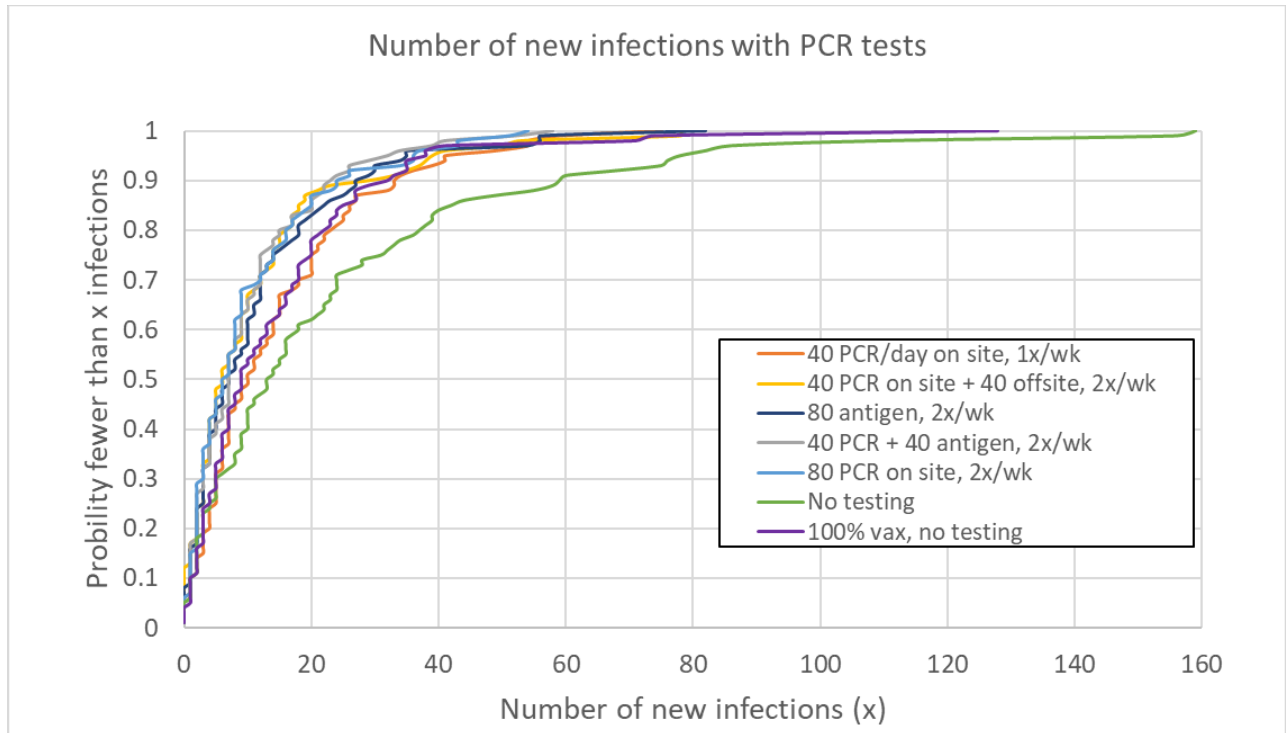


Figure 12 Cumulative distribution of infections


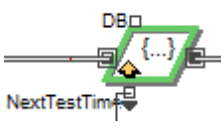
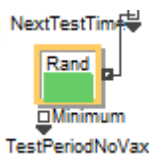
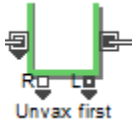
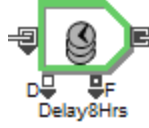

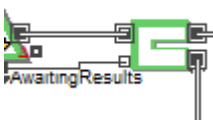

Finally, a useful metric for assessing the cost effectiveness of a testing program can be computed as $\text{cost}/(\text{infections without testing} - \text{infections with testing})$. This is the dollar cost per avoided infection with testing. This risk reduction metric can be compared to other workforce safety investments to improve resource allocation.

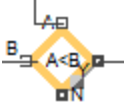

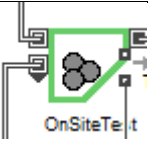
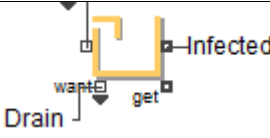
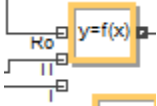
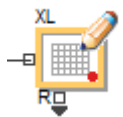

References

- [ANL 2021] CityCOVID – About the Model, <https://www.anl.gov/dis/citycovid-about-the-model>
- [ExtendSim 2021] www.extendsim.com
- [Jason 2020] JASON/MITRE Corp., Managing the Risk From COVID-19 During a Return to On-Site University Research, JSR-20-NS1, pg. 69 (July 2, 2020).

Glossary of Icons Used in the Model

The graphical icons in the figures correspond to code modules in the Extendsim library. The modules used in this model are briefly described here. See [Extendsim 2021] for more details.

Icon	Name	Function
	Create	This module generates objects (persons or test kits in this model) in accordance with a schedule. This node generates the population of persons that are not infected and not vaccinated at time zero. Attributes such as vaccination status are also assigned to each person. Fourteen attributes are assigned in this model.
	Set attribute	This module sets an attribute of each person that passes through it. In this example, the attribute NextTestTime is assigned. The value of the attribute is specified by the data on the line connected to the bottom of the icon.
	Random number	This module generates a random number from the user-specified distribution. In this example, the test period for unvaccinated persons is uniformly distributed between zero and the user specified test period (e.g., 168 hours). This random assignment of the first test time spreads out the tests to reflect an equilibrium state for the testing program.
	Queue	This module holds objects temporarily until called by downstream modules. Objects in the queue can be sorted. In this example, the unvaccinated persons are first in the queue.
	Activity	This module represents a workstation where some action can be performed. The duration of the activity is specified.
	Get attribute	This module reads the value of an attribute as the object passes through it. In this example, the value of the binary variable AwaitingResults is read and passed to the next module in the model.
	Select item out	This module routes objects as directed by the input value of AwaitingResults. If the person is not waiting for test results (AwaitingResults = 0). Then the person is routed to the top line. If not, the bottom line is selected.
	System time	This module output the current system time.

	Decision	<p>This module compares the values A and B. If $A < B$, then the node outputs a value of 1 on the line to the right. Output is zero otherwise. The module is frequently used to compare a timestamp attribute to the current time to see if an event is scheduled to occur (e.g., is it time for a test).</p>
	Shift	<p>This module generates a binary value in accordance with a user-specified schedule. In this model, it is used to designate a weekend day when testing is not performed.</p>
	Batch	<p>This module pairs two inputs. In this model, persons are paired with onsite test kits if they are available. If no test kits are available onsite, a zero signal is generated on the line at the lower right of the icon. This signal is used to route persons to off-site tests.</p>
	Holding tank	<p>This module accumulates objects that enter through the top line. The current number of objects in the tank is assigned to the variable Infected. The tank is drained each night to reset the count of the infected population.</p>
	Equation	<p>This module implements an arbitrary equation specified by the user. In this example, the module generates the probability of an infection using the inputs R_o, T_I, and I.</p>
	Write	<p>This model writes model output to an Excel file. Any defined variable in the model can be written during model execution or at the end of the run.</p>
	DE Plotter	<p>This model plots variables in the model during execution or at the end of the run.</p>