U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND
CHEMICAL BIOLOGICAL CENTER

ABERDEEN PROVING GROUND, MD 21010-5424

DEVCOM CBC-TR-1747

Estimators for Disease Dynamics with Imperfect Surveillance

Tom Ingersoll
RESEARCH AND TECHNOLOGY DIRECTORATE

November 2021

Approved for public release: distribution unlimited.

Disclaimer

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorizing documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 h per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
XX-11-2021	Final	Sep 2020–Feb 2021

4. TITLE AND SUBTITLE
Estimators for Disease Dynamics with Imperfect Surveillance

5a. CONTRACT NUMBER
.

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Ingersoll, Tom

5d. PROJECT NUMBER
CB11017

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Director, DEVCOM CBC, ATTN: FCDD-CBR-IM, APG, MD 21010-5424

8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM CBC-TR-1747

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Threat Reduction Agency, 8725 John J. Kingman Road, MSC 6201, Fort Belvoir, VA 22060-6201

10. SPONSOR/MONITOR'S ACRONYM(S)
DTRA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT:
This report presents a method for estimating infective disease-dynamics parameters when contact rates are uneven, surveillance data are not systematically sampled, and cases are underreported. An important parameter for predictive infectious disease models is the effective reproductive number (R_e). R_e determines the rate at which new infections occur and respond to intervention strategies, such as vaccination, quarantine, and social distancing. However, accurate estimation of R_e is complicated by shortcomings in surveillance data collection, and these shortcomings are difficult to mitigate through changes in sampling methods. The author proposes that estimation of R_e is not necessary to model changes in disease dynamics; rather, the basic reproductive number R_0 may be used along with contact parameters derived from network characteristics within the host population. In addition, estimates of R_0 can be derived from imperfect surveillance data through application of hierarchical methods that correct for underreporting by using explicit estimates of detection probabilities. A hierarchical data-assimilative method for improving parameter estimates in predictive models when data are imperfectly collected is demonstrated in this report. Accurate estimates of changes in disease dynamics can inform management decisions and mitigation strategies.

15. SUBJECT TERMS
Coronavirus disease 2019 (COVID-19)
Maximum likelihood
Susceptible, exposed, infective, or resistant (SEIR) model
Network model
Underreporting
Surveillance
Parameter estimation
Disease dynamics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Renu Rastogi
U	U	U	UU	48	**19b. TELEPHONE NUMBER** *(include area code)* (410) 436-7545

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Blank

PREFACE

The work described in this report was authorized under project number CB11017. The work was started in September 2020 and completed in February 2021. This work was performed at the U.S. Army Combat Capabilities Development Command Chemical Biological Center (DEVCOM CBC; Aberdeen Proving Ground, MD).

The use of either trade or manufacturers' names in this report does not constitute an official endorsement of any commercial products. This report may not be cited for purposes of advertisement.

This report has been approved for public release.

Acknowledgments

The author thanks Michel Kierzewski (DEVCOM CBC) for help with report content, Dr. Sweta Batni (Defense Threat Reduction Agency; Fort Belvoir, VA) for funding, and Kelly Zaledonis (Battelle; Columbus, OH) for editing.

Blank

CONTENTS

FIGURES

ESTIMATORS FOR DISEASE DYNAMICS WITH IMPERFECT SURVEILLANCE

1. INTRODUCTION

1.1 Purpose

This report presents a maximum-likelihood method for estimating infective disease-dynamics parameters when contact rates are uneven, surveillance data are not systematically sampled, and cases are underreported. A critical and difficult to estimate parameter for predictive infectious disease models is the effective reproductive number (R_e). R_e determines the rate at which new infections occur and respond to intervention strategies, such as vaccination, quarantine, and social distancing. However, accurate estimation of R_e is complicated by shortcomings in surveillance data collection, and these shortcomings are difficult to mitigate through changes in sampling methods. The author proposes that estimation of R_e is not necessary to model changes in disease dynamics; rather, the basic reproductive number (R_0) may be used along with contact parameters derived from network characteristics within the host population. In addition, estimates of R_0 can be derived from imperfect surveillance data through the application of hierarchical methods that correct for underreporting through explicit estimation of detection probabilities. A hierarchical data-assimilative method for improving parameter estimates in predictive models when data are imperfectly collected, Estimators for Disease Dynamics with Imperfect Surveillance (EDDIS), is demonstrated in this report. Accurate estimates of changes in disease dynamics can inform management decisions and mitigation strategies.

1.2 Background

Parameter estimates for predictive disease models rely on inadequately-sampled surveillance data. Sampling is typically uneven, limited to case reports of patients already known to be ill, and supplemented with specific tests directed at suspected cases (confirmatory sampling). Confirmatory samples may contain underreporting, where testing fails to detect infection, as well as uneven samples, where large portions of population are unrepresented. Because only suspected cases are included in sampling, asymptomatic and mild cases are disproportionally overlooked. This results in underestimates for the number of infected hosts and misrepresents the community at large. As a result, rate parameters for dynamic models do not represent the true underlying state of infection. Methods using hierarchical models are well established for correcting underreporting (1). These methods are particularly relevant for confirmatory samples because confirmatory samples tend toward high specificity, where false positives are limited (2). However, methods for correcting underrepresentation and the irregular collection of confirmatory samples are more elusive. Correcting for unsystematic sampling would require that we supplement confirmatory samples with information more inclusive of the broader population.

Predictive dynamic models can produce data ranging across the broader population and might be used to overcome problems with unsystematic data collection. This process would require that parameters be accurately estimated, that is, free of over and under

estimates. Dynamic models could be combined with confirmatory samples in an ensemble model to correct estimation inaccuracies while providing broad representation. Such models are called data-assimilative models, the invention of which can be credited to Gauss (3). Data assimilation combines the strengths of dynamic models and confirmatory sampling while mitigating their weaknesses. Data assimilation combined with hierarchical methodology that explicitly models detection probabilities could be used to adjust misestimates in dynamic model parameters, increasing reliability and the usefulness of model-based predictions.

To model dynamics of acute infectious disease in a host community, the populations are divided into categories based on the infectious state of individuals. The state of the host can be described as follows: susceptible, exposed, infective, or resistant (SEIR), and the model represents the count of individuals in each these states at any given time. Models containing these categories are called SEIR models, and a goal of dynamic models is to estimate transition rates between SEIR categories (4, 5), such as the infection rate (r). Susceptible individuals (S) are not yet infected but are available to become hosts to the disease. Exposed individuals (E) have been exposed to the disease but have not yet developed enough disease to transmit it to others; therefore, they are latent and not yet infective. Infective individuals (I) have acquired the disease and may pass it to others. Resistant individuals (R) have acquired immunity and can no longer become hosts or transmit the disease to others. Therefore, the cumulative count of cases at any time is the sum of I and R. SEIR models are mathematical approximations intended to predict population-level disease outcomes. The terms of the model are adjusted until model output begins to resemble surveillance observations, then these terms are used to predict the future trajectory of the disease. Terms may be adjusted by expert judgement or estimated using statistical methods. A SEIR model was successfully applied to medical planning during the 2003 Severe Acute Respiratory Syndrome (SARS) outbreak (6) using Bayesian methods to estimate terms. In this report, the SARS model is adapted to the closely related disease coronavirus disease 2019 (COVID-19). An important assumption of such models is that the population is well-mixed; that is, all individuals have an equal probability of coming into contact with an infective host (5). The author demonstrates a method for relaxing this assumption.

Isaac Newton's generalized binomial theorem (7) can be used as a method for estimating detection probability using predicted values from dynamic models and counting data from surveillance:

$$P = \binom{n}{k} (p)^k (1-p)^{n-k} \quad (1)$$

where, for our purposes, *n* is the day's predicted cumulative case count from the dynamic model (I + R), *k* is the day's count from the surveillance data, and *p* is the detection probability, which can be estimated via maximum likelihood. The result is a method for estimating underreporting expected to be present in the surveillance data (Figure 1).

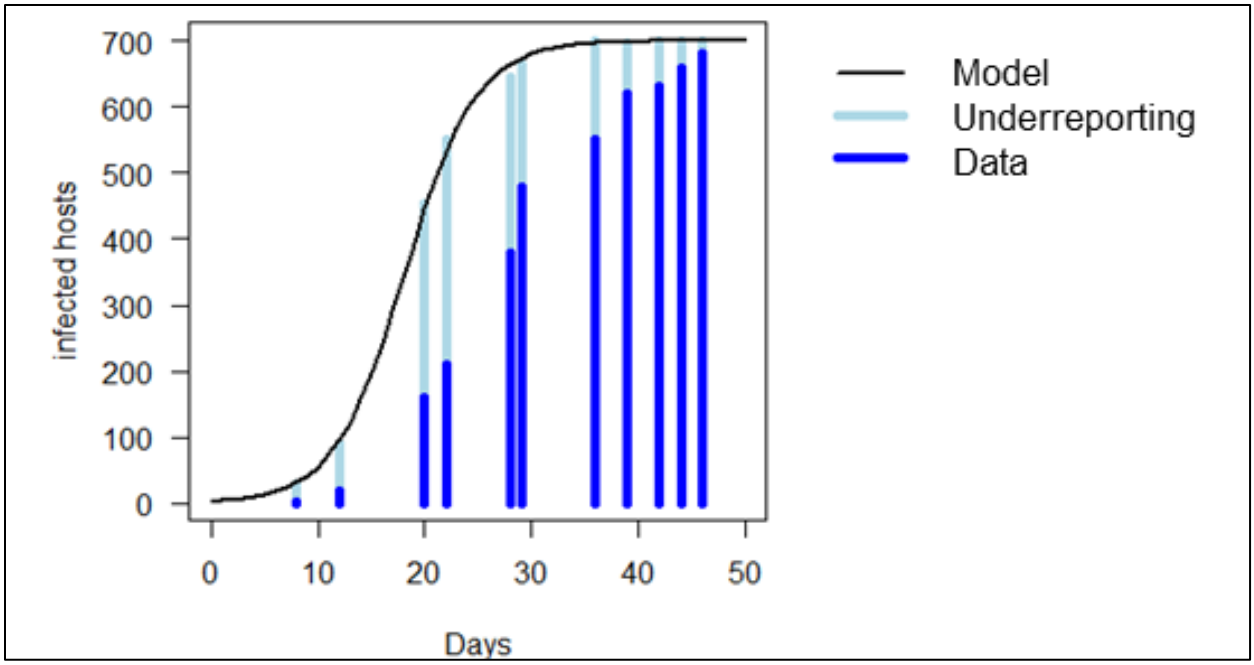

Figure 1. Hypothetical data with underreporting compared with the counts produced by a dynamic model.

The parameters for the dynamic model, which produce *n*, can also be adjusted with maximum likelihood. The result is data assimilation, where data are used to adjust a statistical model for detection, and a dynamic model for the count of cases.

Model output exceeding corresponding values in data can be explained by underreporting in the data; however, model values below the data produce probabilities of 0 according to Newton's theorem. Plausible model values must therefore meet or exceed corresponding values in data.

It is both critical and difficult to estimate the R_0 parameter for SEIR models (5). R_0 is the expected number of new infections arising from an infected host in an arbitrarily large, naive population. It is assumed that R_0 is a fixed rate characteristic of a pathogen strain and that changes in r can be attributed to changes in contact rates between hosts (5). This is sensible; many otherwise susceptible individuals are not available for contact with infective hosts because of spatial, environmental, or social distances, or mitigation strategies such as quarantine, each of which result in a population that is not well-mixed. Disease dynamics models for populations that are not well-mixed are called network models (8). The author proposes that changes in contact rates can be approximated by changes in the effective size of the susceptible population, resulting in an open-population model that behaves like a network model, where rates of infection are heterogeneous.

Network models have two fundamental components: nodes, which are subpopulations that exist in incomplete isolation from other nodes within a metapopulation, and links, which are corridors of contact between nodes (8). If we assume that R_0 is fixed, then changes in r can be explained by changes in the size of nodes, and the timing of changes can be

explained by contact with links. Changes in R_0 would therefore not occur without pathogen mutation and emergence of new pathogenic strains.

1.3 Data Sources

Open-source data for the COVID-19 pandemic are available for the Diamond Princess cruise ship (9). The Diamond Princess is a special case in which management decisions affecting quarantine, social distancing, and removal of passengers are well documented in a small population appropriate for network models. A hypothesis that a change in r is coincident with management strategies affecting the size of the exposed population can therefore be confronted with data. At the beginning of the sampling period, the Diamond Princess had a population of 2670 passengers and 1100 staff. By the end of the sampling period, all those infected had been removed shoreward into quarantine (10). Complete surveillance had also been attained by the end of the sampling period; all passengers had been repeatedly tested for COVID-19 using polymerase chain reaction (PCR) methods. The PCR method has high specificity (low false-positive rates), but it can result in underreporting because genetic material from the pathogen must be present in each sample. As a remedy, a second PCR test can be recommended to confirm negative results (11). Because testing frequency increased and all passengers were tested repeatedly by the end of the sampling period, a valid model should converge on testing results toward the end of the period, and estimated rates of detection should increase across the period toward 1.

2. METHOD DEVELOPMENT

Models were based on SEIR models developed for the 2003 SARS infection (6). These models were executed in the computational language R (12), using the R library of differential equation solvers *deSolve* (13) and R functions coded by the author (Appendix). Functions that estimate and model detection probability, based on Newton's theorem, were written by the author (Counts; Appendix).

2.1 Data

Data were extracted from daily reports prepared by the Japanese Health Ministry and published online by the Johns Hopkins University (Baltimore, MD) (9) (Figure 2).

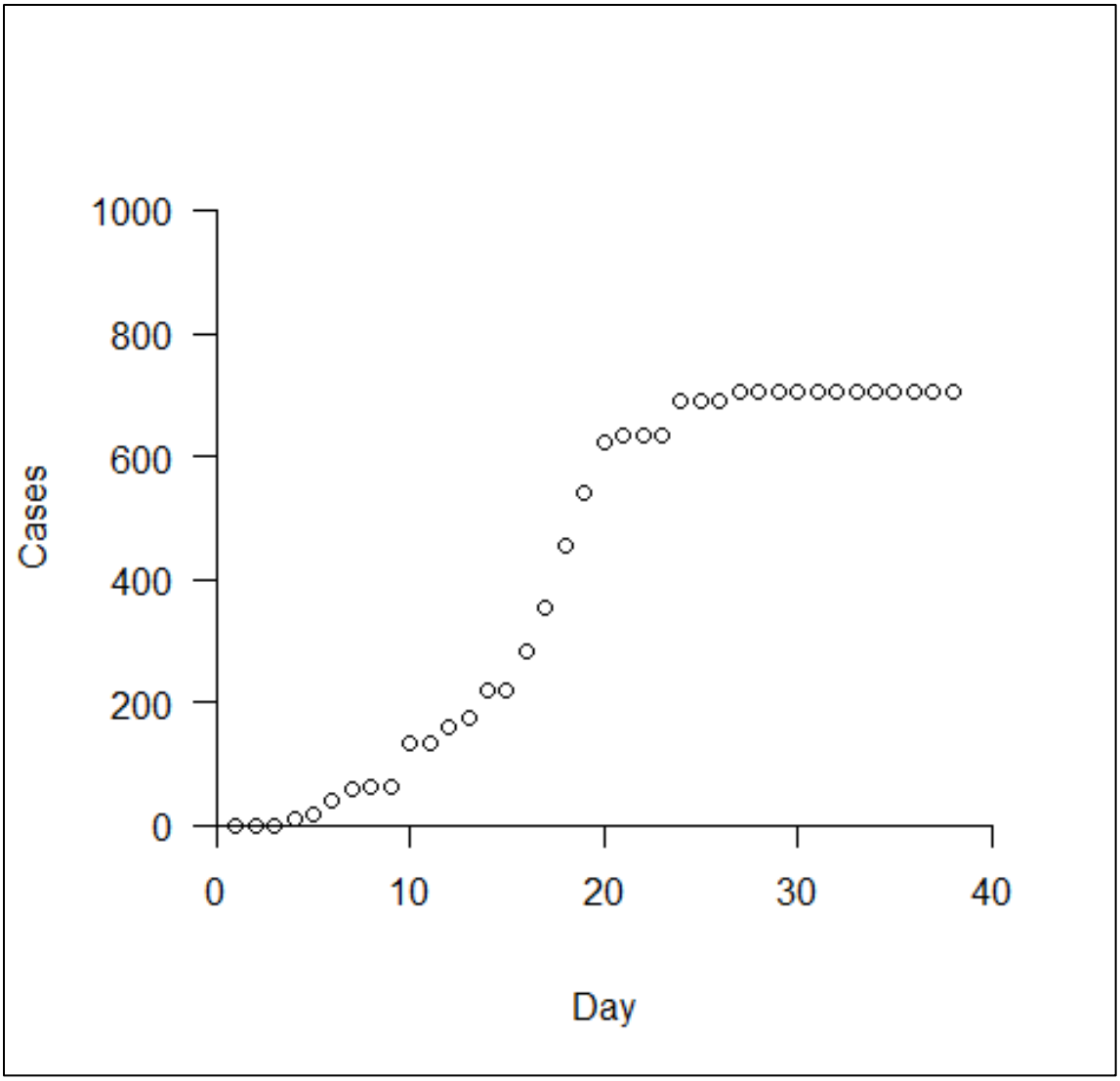

Figure 2. Daily reports through day 38.

Data were counts of infective cases originating aboard the cruise ship Diamond Princess between 2 February and 8 March 2020. Beginning on day 27, no new cases were observed for the remainder of the reporting period, despite daily sampling. For this reason, an assumption was made that the effective population had been fully exposed by day 27. Note that there were several sequences of subsequent days earlier in the series, for example, days 8 and 9, for which no change in the number of cases was reported. This is because no new samples were collected on those days, even though a daily report was issued. These days contribute no new information and reduce model likelihood; therefore, they were not included in the analysis.

Only Days 0 through 27, with reported cases from days without new sampling removed, were included in the analysis. This resulted in a relatively small (n=21) dataset (Figure 3), which was used to calculate maximum-likelihood estimates.

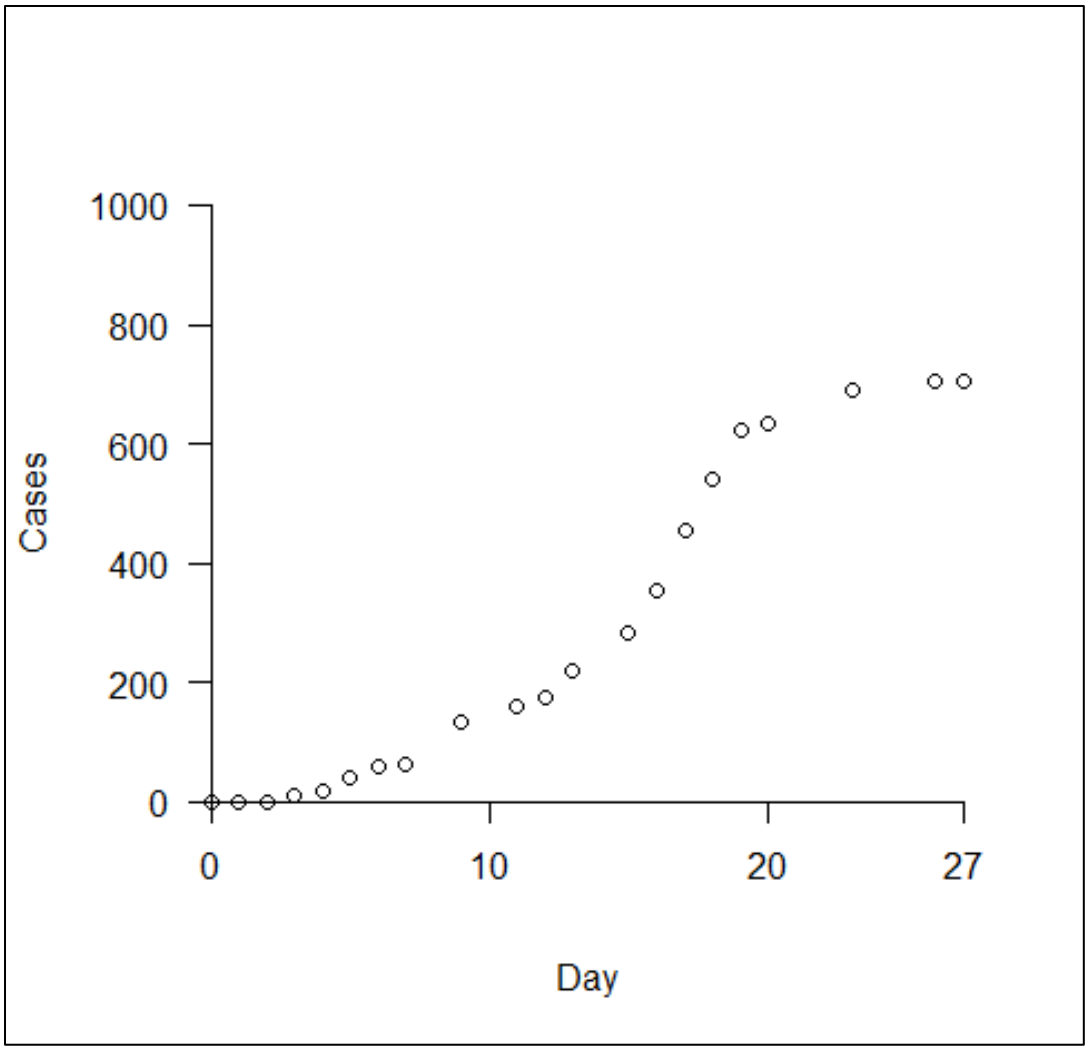

Figure 3. New samples through Day 27.

2.2 SEIR Models and the Well-Mixed Assumption

A series of SEIR models that conformed to the well-mixed assumption were produced (Figure 4).

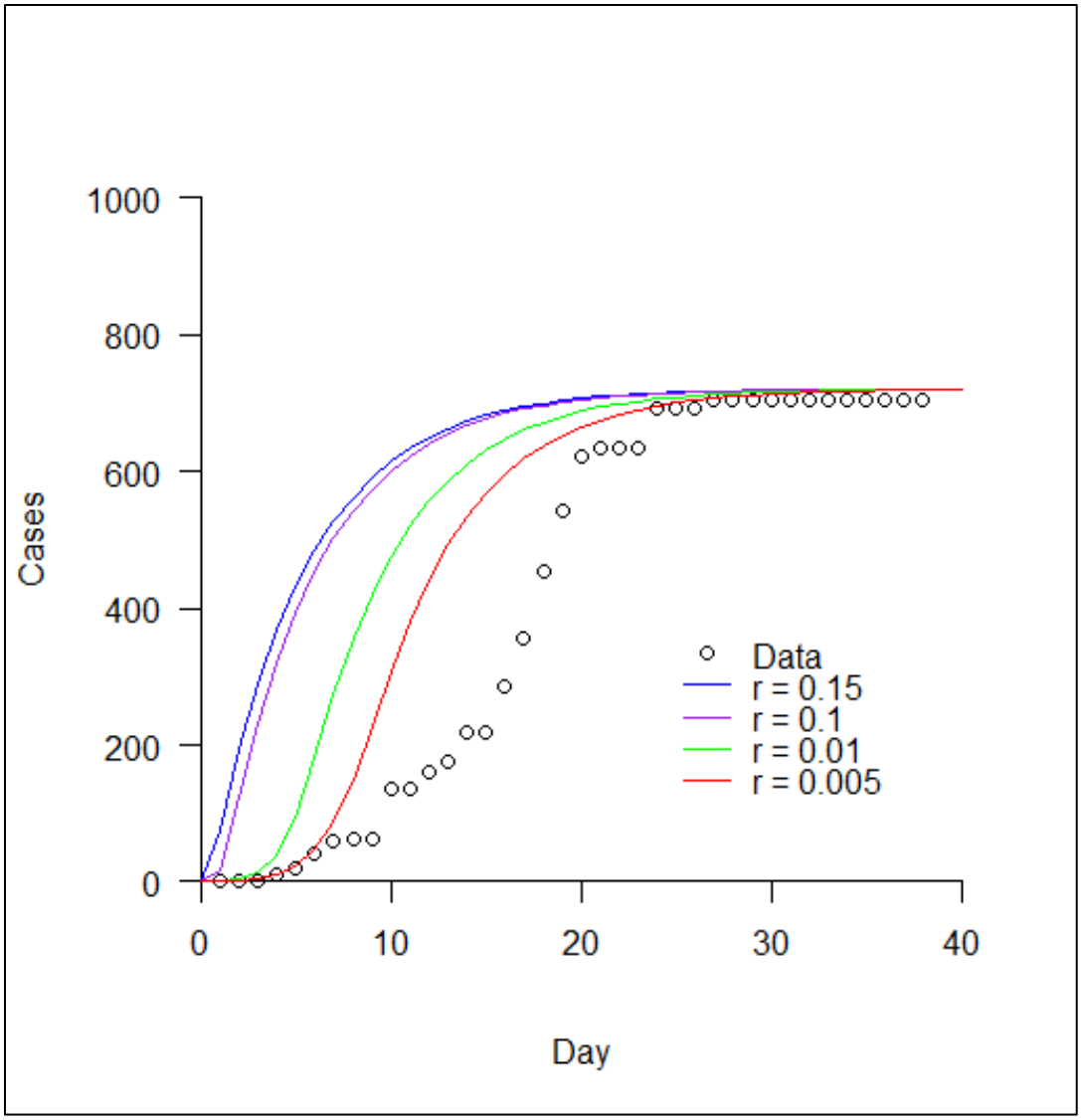

Figure 4. The failure of the well-mixed assumption.

The traces in Figure 4 represent predicted cases from well-mixed SEIR models with growth rate r. The likelihoods produced when these models confronted the data were found to be inestimably small. Curves with values that matched or exceeded the data values at the beginning and end of the series could not be made close enough to intermediate days' values to produce probabilities high enough to be estimated with available computing resources. This is a recurring problem with maximum-likelihood estimation. However, models with inestimably small likelihoods are inadequate for describing the modeled process using maximum-likelihood methods. Alternatively, it will be demonstrated in Section 2.3 that network models with estimably high likelihoods could be produced from these data.

2.3 The Open-Population Network Model

To relax the well-mixed assumption, an open-population network model was developed (Figure 5).

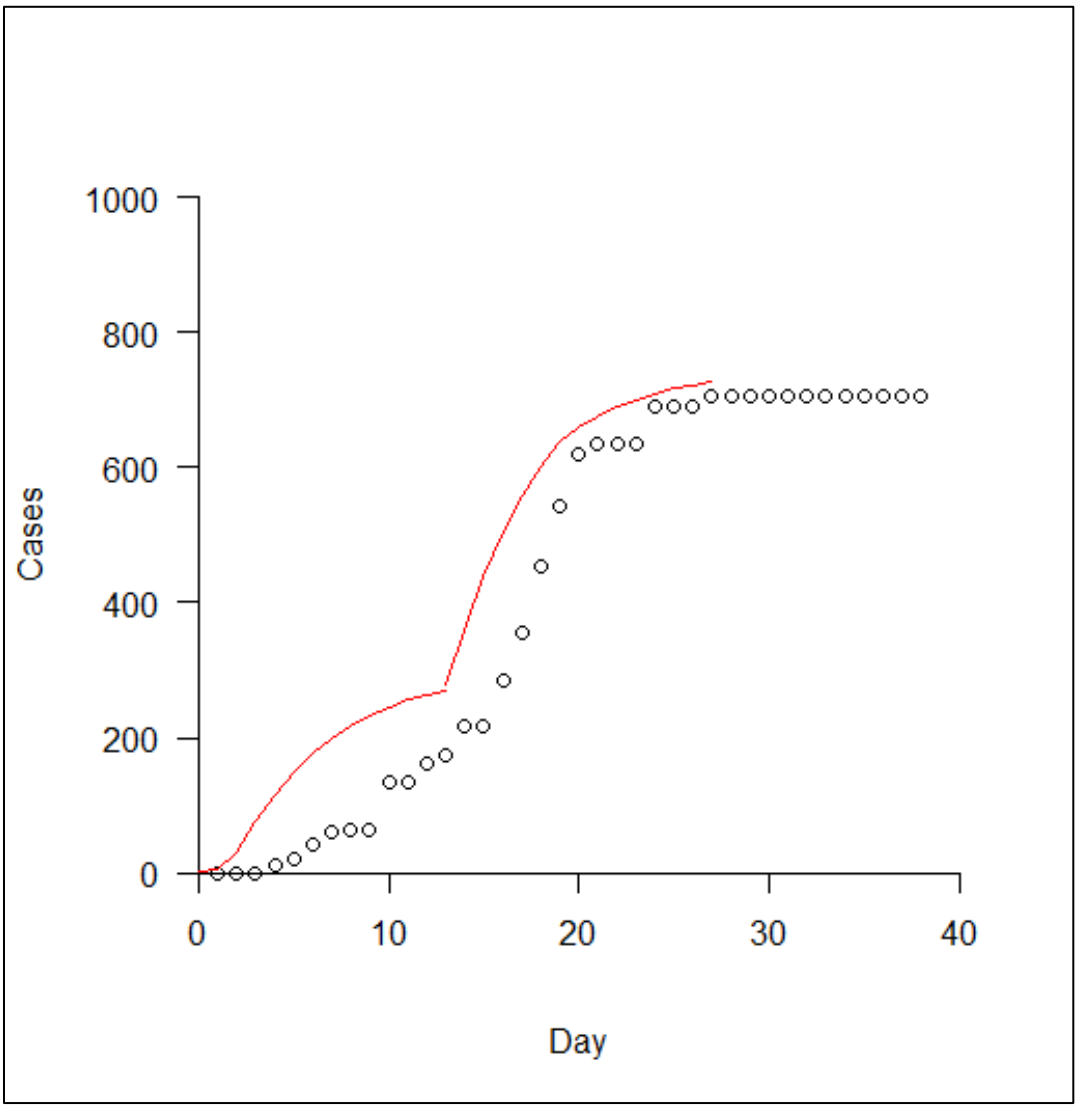

Figure 5. A network model with underreporting (red trace).

Spread between incompletely isolated sub-populations (nodes) was simulated by adjusting the effective population size, which created shoulders in the curves such as those seen prior to day 15. These changes in effective population were programmed by changing the size of the susceptible population (S) term in the differential equation solver.

2.4 The Detection Models

The probability of model output was compared with each datum using Newton's theorem to give a likelihood for its parameter values, including likelihood for values of the detection probability. The value of detection probability producing the highest probability for that day's case count, given the other model parameters, was its point-wise maximum-likelihood estimate (Figure 6).

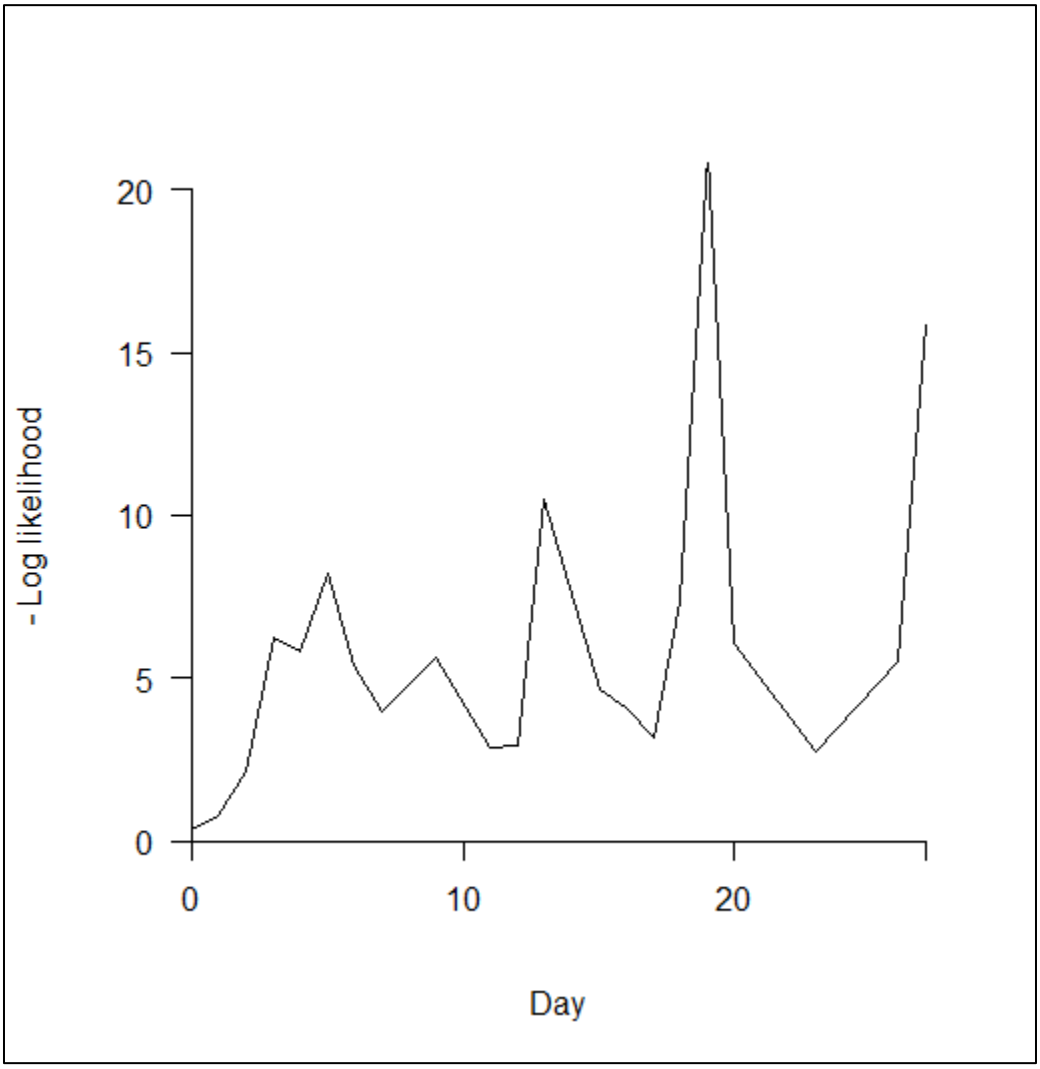

Figure 6. The point-wise negative-log likelihoods for detection values.

Using the network model rendered all detection probabilities estimable (Figure 6). The negative log-likelihoods are shown, meaning that peaks show less fit to data. The absolute value of the negative-log likelihood is meaningless, other than its being high enough to be estimated through computation. But the relative values can be used to compare estimates of model terms, thereby enabling researchers to select those terms that produce the highest likelihoods (lowest negative log-likelihoods).

Because all values were estimable, maximum-likelihood methods were used to produce a maximum-likelihood estimate of daily detection probability. These values were then used to produce a logistic model for detection probabilities across the entire data series (Figure 7).

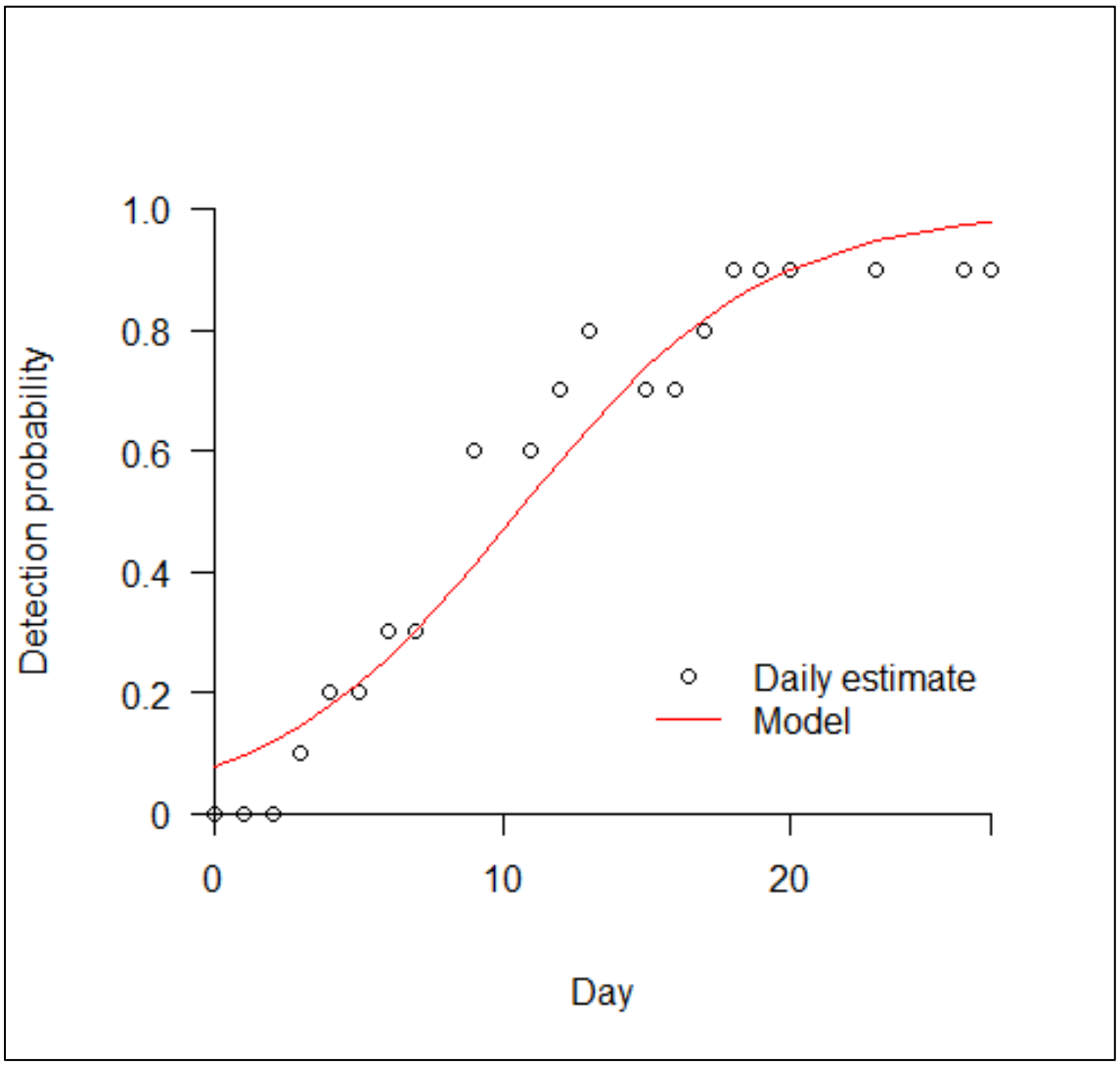

Figure 7. Logistic model for detection probability.

2.5 Adjusting r

The detection model itself had an estimable likelihood, which allowed the maximum-likelihood value for the growth rate r to be estimated. Adjusting the growth rate had the effect of changing the limit of the model slope at any given point (Figure 8).

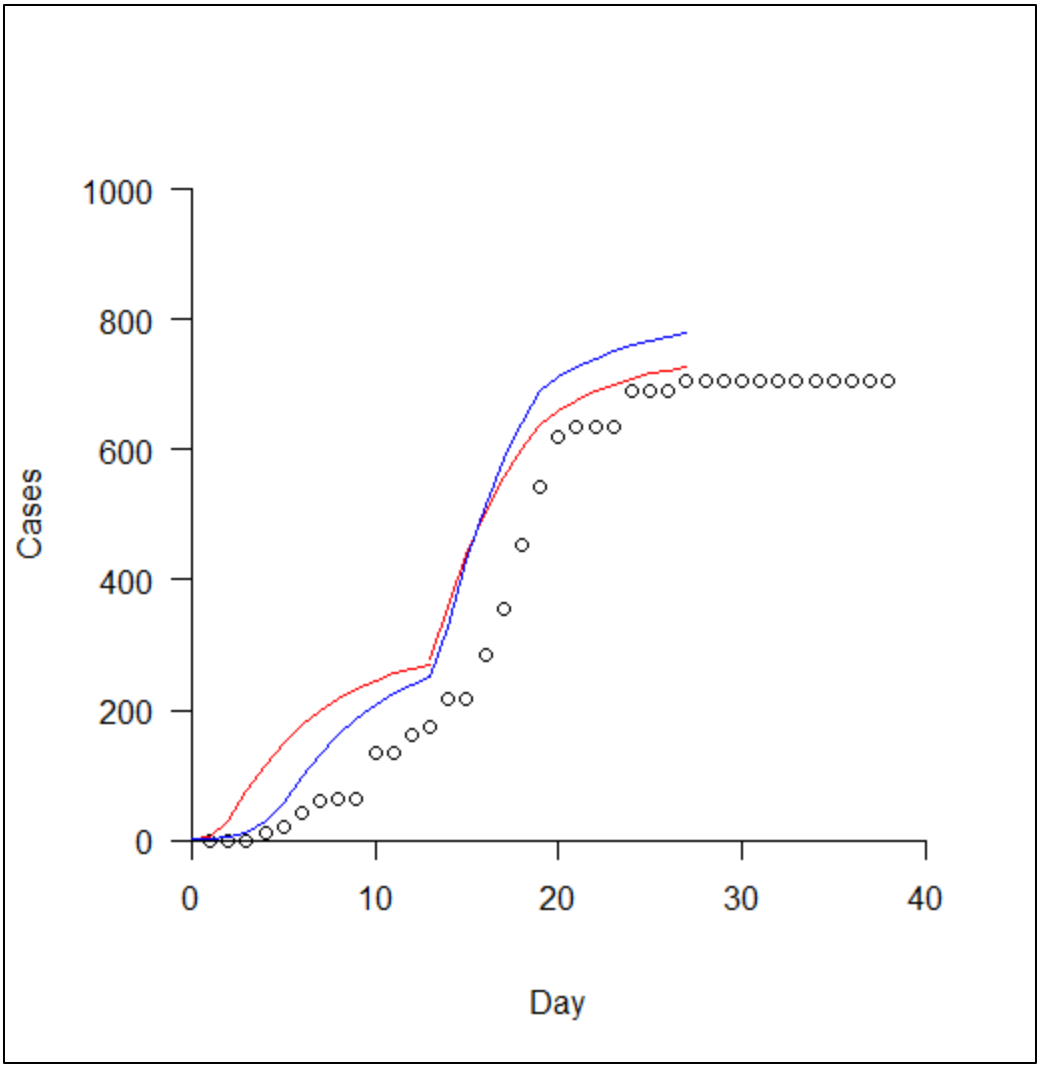

Figure 8. Adjusting r.

2.6 Adjusting the Node Size

Adjusting the node size had the effect of moving the shoulders between alternative models (red versus blue traces) in the network model (Figure 9).

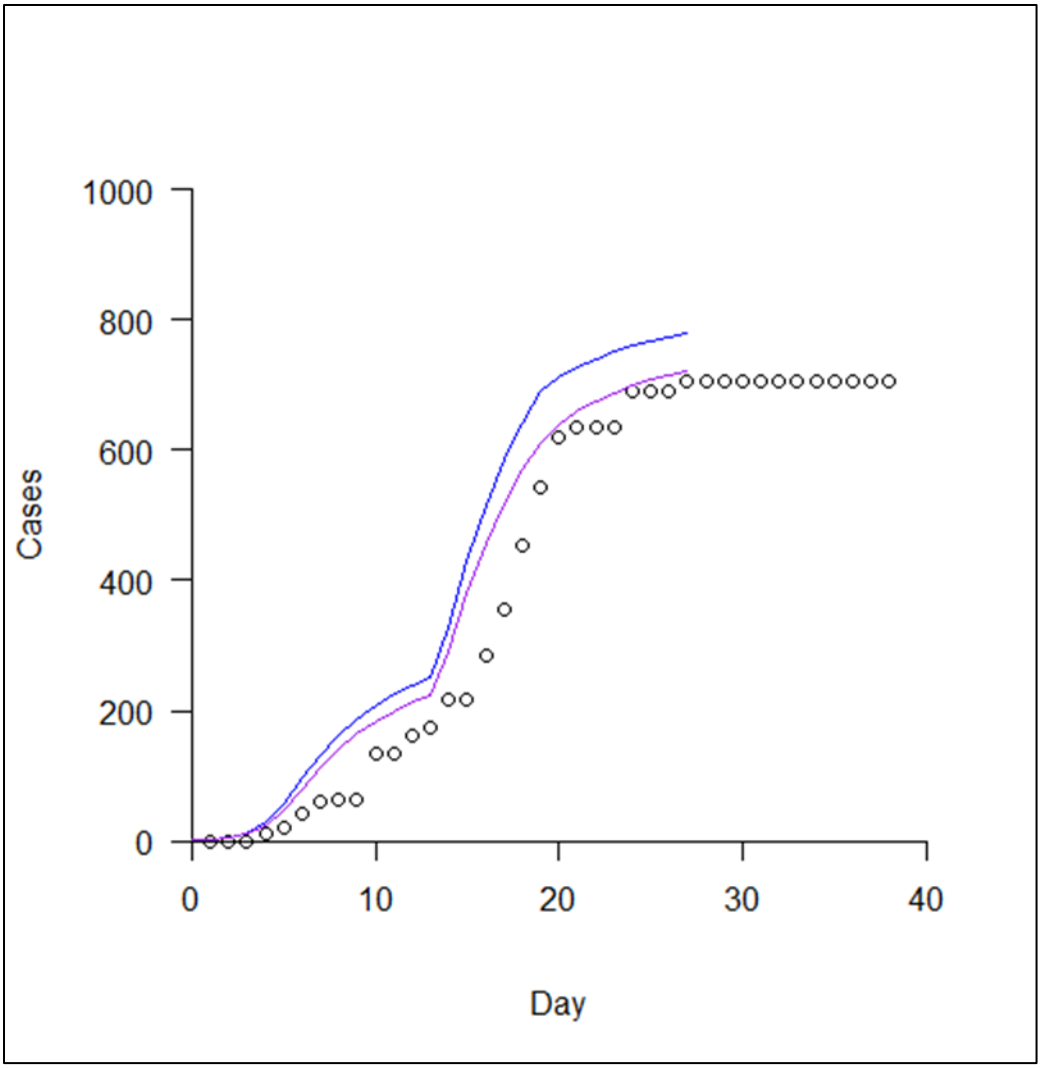

Figure 9. Adjusting the size of the nodes.

The days where links were encountered, and thus node size changed, were limited to three for this proof-of-concept. Likelihoods were then compared between alternative models (blue versus purple traces).

3. RESULTS

3.1 Maximum-Likelihood Estimate for R_0

After optimizing node size and detection probabilities, likelihoods among models with alternative growth rates were compared to get a maximum-likelihood estimate for R_0 (Figure 10).

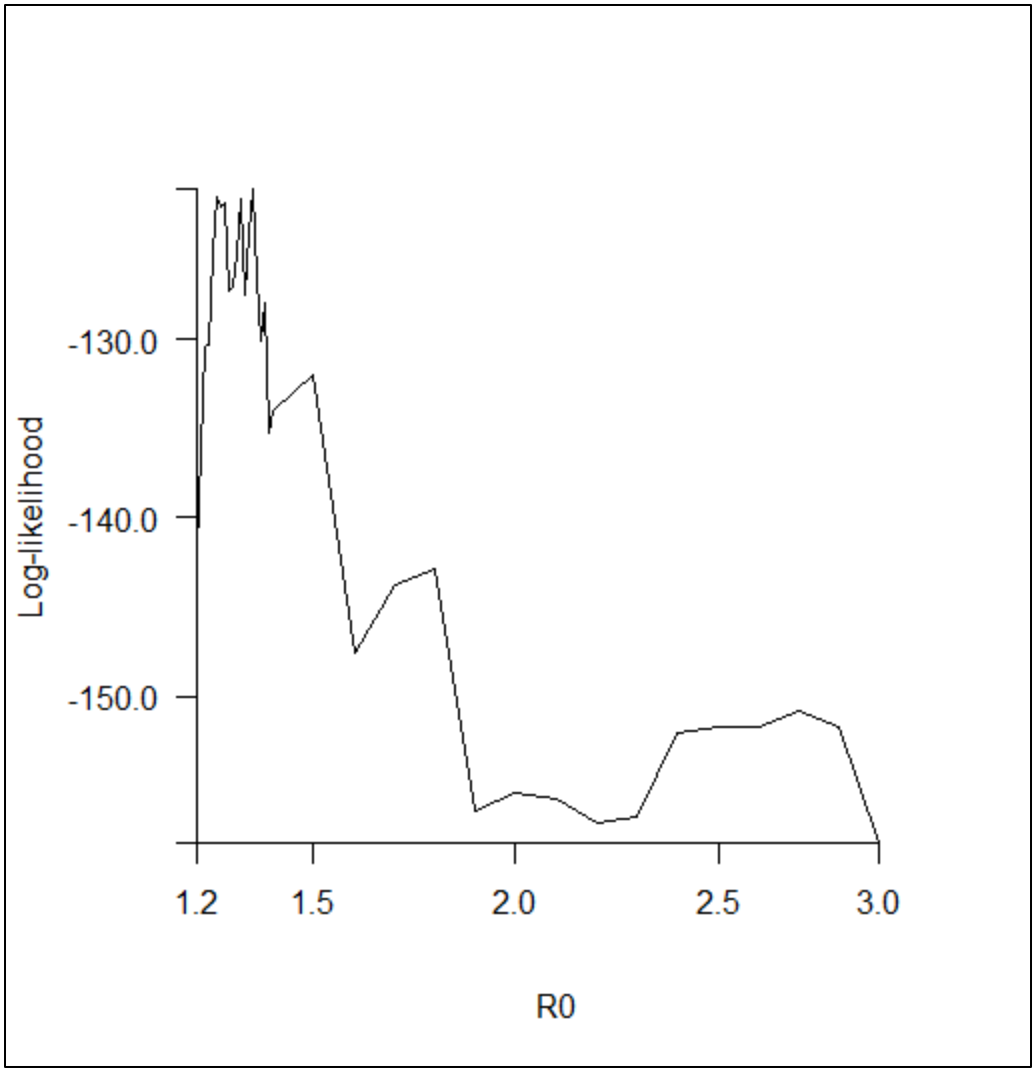

Figure 10. Maximum-likelihood estimate for R_0 (1.26).

Figure 10 exhibits substantial multi-modality, reflecting the complexity of the relation within the sparseness of the data. Estimates of R_0 assumed an infective period of 10 days (14).

3.2 Predictive Models for Small Populations

A predictive model for a different population was assembled based on R_0 and network terms estimated from the Diamond Princess data (Figure 11).

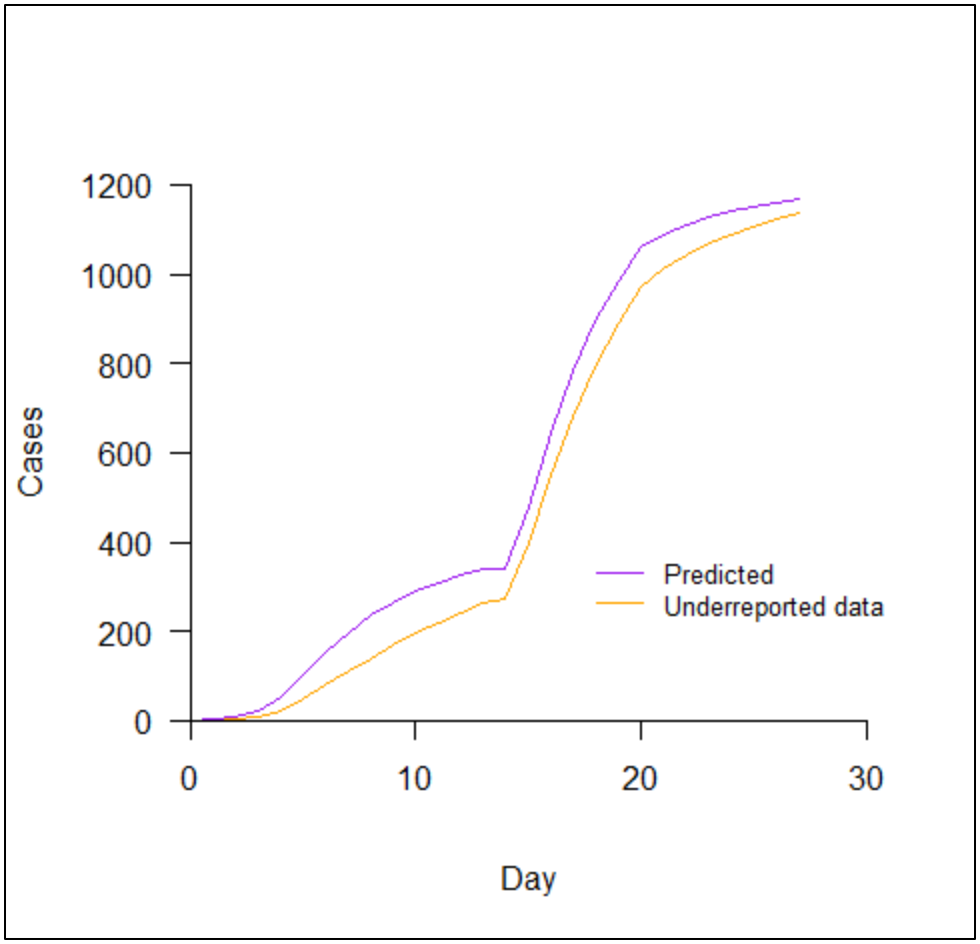

Figure 11. Predictive model for a hypothetical ship.

The model shown in Figure 11 predicts dynamics for a shipboard effective population of 1400, with the estimated true count of infected hosts in purple, and the expected observed counts with underreporting in orange. As with the Diamond Princess data, a link occurs at Day 14, and emigration occurs at Day 19.

Another predictive model for a hypothetical community was assembled based on R_0 from the Diamond Princess data and varying the size of the nodes. Varying the size of the nodes had the effect of changing the network structure and the model trajectory in a way that resembles the trajectory observed early in outbreaks in China and Italy (9). The model was compared to a well-mixed model with a delay in starting time (Figure 12).

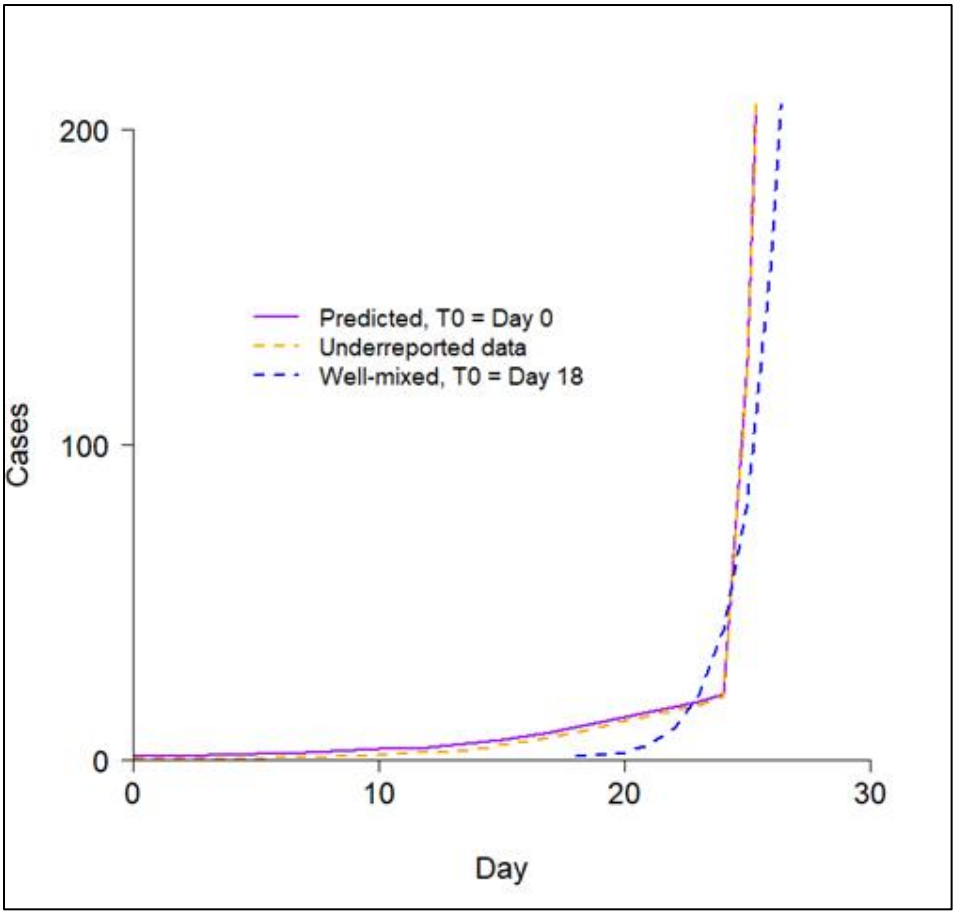

Figure 12. Predictive network model for a hypothetical village. Network model (purple trace), compared to a well-mixed model with an 18-day delay (blue trace).

4. DISCUSSION

4.1 Success of the Demonstration

EDDIS successfully demonstrated that a tool for estimating infective disease-dynamics parameters when contact rates are uneven, surveillance data are not systematically sampled, and cases are underreported could be developed. R_0, node size, the timing of links and emigration, and a subsequent predictive model (Figure 11) were all successfully estimated. This report presented the proof-of-concept. That said, several tasks remain that will complete the tool as an effective aid to inform decision making. These tasks include further validation, development of plausible confidence intervals, generalization through improved modeling of network structure, and especially, further automation through migration to a machine-learning environment. Increased automation will facilitate completion of the other tasks.

4.2 Validation

Although EDDIS successfully demonstrated that a good fit and plausible explanation could be achieved with the Diamond Princess data, validation is far from complete. EDDIS should be further validated by comparison with other surveillance data (14). Ultimately, validation should include extensive testing with simulated data where the underlying generating parameters are known.

4.3 The Detection Models

EDDIS currently uses a logistic model for detection, which creates a monotonic increasing relation between detection and day across the study period. This reflects our prior knowledge about the sampling schedule and devices used on the Diamond Princess, that is, increasing sampling intensity with redundant sampling by the period's end. The logistic model is simply a logit-transformed linear model; the logit transformation is applied to bind the results between 0 and 1 so that the result is a probability. Despite the simplicity of the logistic model, a surprisingly good fit was attained (Figure 7). However, it is easy to imagine why detection might not follow a linear relationship in other cases. When COVID-19 first emerged in the United States, the public health system struggled to deliver testing at a pace matching that of epidemic growth. In this case, detection could not be expected to be monotonic increasing with time. A non-linear model such as a generalized additive mixed-model (GAMM; 15) could provide a better approximation than the logistic model in these other cases. The GAMM has arbitrary curvature, so it can approximate drops and surges in detection, especially when detection covariates such as sampling intensity are recorded and included in modeling. That said, the EDDIS detection model proved to be surprisingly adequate in its simplicity for the Diamond Princess case and allows the data from early in the series, when sampling was irregular and incomplete, to contribute to the estimation of other model terms such as R_0. This has strong implications for other emerging diseases that produce limited and imperfect data early in the outbreak.

The detection estimate itself is useful in that it could be used to inform decisions to deploy additional surveillance resources. Low levels of detection early in the COVID-19 pandemic may have contributed to the virus's spread. Low levels of early detection in a relatively lethal disease could be used to trigger alarms in the public health system.

4.4 The Open-Population Network Model

The network model began with limits applied by expert judgement with consideration for knowledge of events and their timing aboard Diamond Princess. For example, unimpeded spread occurred between passengers prior to Day 5 when a quarantine was declared. Those exposed before the quarantine began are represented by the first node. A shoulder occurring prior to Day 14 suggests that a link occurs near this date and that the first node has a size of approximately 230. The link and node size were then fine-tuned by comparing alternative values using maximum likelihood, giving a node size of 270, which included those individuals who never became infected. Fitting to the steepness of the curve following Day 14 using maximum likelihood gives the size of the second node: 800. Emigration began at Day 12 but

expanded dramatically on Day 17 when all Americans were evacuated, resulting in a third, smaller node, the size of which was estimated using maximum likelihood at 750, along with a link expressed at Day 19. Rather than testing all possible node sizes and links, which would require full automation and a machine-learning environment, a selection of possible values surrounding the suspected dates and sizes was tested. The resulting approximation could certainly be improved by programming additional automation into the algorithm, which was beyond the means of the current proof-of-concept. That said, the network model provides a far better data fit than any of the well-mixed models (Figures 4 and 5).

According to the model, COVID-19 began aboard the Diamond Princess in a subpopulation of 270 individuals. Because of its relatively low infectivity (R_0 of 1.26) and partial isolation from the rest of those onboard, the disease was able to persist undetected for an unknown number of days and spread throughout the node until a link with a larger node triggered a steep period of spread starting around Day 14. This spread never reached its maximum because the effects of emigration began to take over on Day 19, creating a smaller node and leveling the curve. According to the data, no new cases occurred after Day 27.

Network models are a broad class of models that relax the well-mixed assumption. This might be restated as models that reflect heterogeneous isolation of parts of the population (nodes). Transmission between nodes occurs during a link. Currently, many network models treat the nodes as small spatially compartmented subpopulations that are connected spatially through restrictive links. The result is an often complex, web-like structure with many small components. The author suggests that these basic concepts of networks might be redefined in functional terms, so that a node is a subpopulation that exhibits dynamic behavior as if it is well mixed, a link is a temporal event that occurs when new nodes exhibit exposure to the pathogen, and its magnitude is the size of the exposed class. The result is a simplified open-population model whose metrics can be estimated using maximum likelihood, as shown in Figure 9.

For example, hospital staff might be viewed as a node when protection is inadequate and transmission occurs freely. They are not necessarily spatially isolated from other staff who are not exposed, but connectivity is heterogeneous because of behavior. A portion of those exposed provide the opportunity for another link when the disease is transmitted to their children and enters a second node, a population at schools. This may occur at several schools, so the node does not have spatial continuity but behaves as a well-mixed population because of coincident exposure, resembling an open population. The link occurs when this node begins to exhibit signs of infectivity, and its magnitude is the number of individuals infected by hospital staff. Such a system may be characterized through examination and analysis of the surveillance data rather than reliance on prior knowledge of its spatial structure.

4.5 The SEIR Model

EDDIS development began as an effort to use imperfect data to estimate R_0 so that data assimilation could be used to augment SEIR. In these respects, EDDIS development has been fully successful. Beginning with a SEIR predictive model for a closely-related pathogen, which was used for the 2003 SARS outbreak (6), SEIR was used with a network model and

COVID-19 surveillance data to populate a model for detection. The combination of process (SEIR in network) with observation (detection) modeling is called a hierarchical model (1). Data assimilation was through Newton's theorem, where model output and data were combined to estimate both the detection parameter and the likelihood. Estimates of R_0 obtained in this manner were predictive of observed dynamics (Figure 5) and useful in predicting the properties of simulated populations (Figures 11 and 12). This is because within-node populations approximately comply with the well-mixed assumption while retaining incomplete isolation from other nodes.

4.6 The Predictive Models

Predictive models used R_0 and network parameters extracted from the data-assimilated models, which were then used to predict the trajectories of simulated populations. We simulated two populations: (1) a simulated shipboard population that differed from Diamond Princess only in initial population size, and (2) a simulated village population with node size selected to mimic small populations in China and Italy (9) that were affected early in the pandemic.

The shipboard model (Figure 11; purple trace) behaved exactly as expected; it was simply a larger version of the Diamond Princess model (Figure 9; purple trace). Infection started slowly, eventually saturating the primary node, then grew exponentially after encountering a link with a larger node at Day 14. Exponential growth then changed to logistic growth following emigration on Day 19. The expected value of underreported data was also estimated (Figure 11, orange trace). Recall that model values can be expected to equal or exceed observed values. This demonstrates that the model can begin with prior knowledge of a similar system and then be adjusted for known differences in conditions to make predictions.

The village model (Figure 12, purple trace) represented an outbreak that began at Day 0 in a small node where it persisted at low, largely undetected levels until encountering a link, whereupon exponential growth was initiated. A well-mixed model is offered for comparison (blue trace). However, note that the well-mixed model had a lower R_0 (1.1 vs 1.25) and must be delayed 18 days to provide similarity to the network model. This demonstrates that the networked disease can persist undetected at low levels, then suddenly erupt into exponential growth. The well-mixed model cannot be made to resemble this pattern of rapid growth following low-level persistence. In fact, it is believed that the COVID-19 outbreak began as early as early October 2019 in Italy and China (16) and persisted undetected at low levels until its discovery in January 2020, which was coincident with eruption into exponential growth. The author has seen no class of models other than network models that adequately explain this pattern.

4.7 Recommendations

Further development should begin with increased automation. Automation provides the gateway to validation, confidence boundaries, analysis of larger data sets, and divestment of some of the need for expert operation.

Development should include incorporation of more complex detection models, particularly GAMMs. Estimation of node parameters such as link timing, link magnitude, and the size of nodes could be enhanced through further automation. Increased automation is fundamental to generalization so that EDDIS can be applied to a variety of emerging diseases and surveillance conditions. Addition of a graphic data visualization toolbox could begin with the R codes for figures included in the Appendix. For use in large datasets, addition of a Bayesian methods toolbox may be helpful in reducing computational errors generated by inestimable likelihoods.

The author would consider EDDIS in its current state to be suitable for use by experts and developers. In particular, EDDIS can be used with small imperfect surveillance data sets to estimate R_0 and network parameters when these values can be selected from a relatively small set of values, that is, when values are well-bounded. Expert judgement and wrapper programming would still be required, but the necessary functions are provided (Appendix) to produce the analysis and graphics given here. Application of EDDIS to other data sets (14) will provide important validation. These analyses could be used for prediction and then compared with trajectories from later in the pandemic.

Blank

LITERATURE CITED

1. Royle, J.A.; Dorazio, R.M. *Hierarchical Modeling and Inference in Ecology*; Elsevier: London. 2008.

2. Ingersoll, T.E. Reducing False Positives via Capture/Recapture Modeling for Early-Warning. In *FY17 Proceedings of the Edgewood Chemical Biological Center*. U.S. Army Edgewood Chemical Biological Center: Aberdeen Proving Ground, MD, 2017, pp 140–147; UNCLASSIFIED Publication.

3. Forbes, E.G. Gauss and the Discovery of Ceres. *Journal for the History of Astronomy*. **1971,** *2* (3), 195–199.

4. Kermack, W.O.; McKendrick, A.G. A Contribution to the Mathematical Theory of Epidemics. *Proc. R. Soc. Lond.* **1927,** *A115,* 700–721.

5. Anderson, R.; May, R. *Infectious Diseases of Humans, Dynamics and Control*; Oxford University Press: New York, 1991.

6. Riley, S.; Fraser, C.; Donnelly, C.A.; Ghani, A.C.; Abu-Raddad, I.J.; Hedley, A.J.; Leung, G.M.; Ho, L.M.; Lam, T.H.; Thatch, T.Q.; Chau, P.; Chan, K.P.; Leung, P.Y.; Tsang, T.; Ho, W.; Lee, K.H.; Lau, E.M.C.; Ferguson, N.M.; Anderson, R.M. Transmission Dynamics of the Etiological Agent of SARS in Hong Kong: Impact of Public Health Interventions. *Science* **2003,** *300,* 1961–1966.

7. Kline, M. *History of Mathematical Thought*; Oxford University Press: New York, 1972.

8. Keeling, M.J.; Eames, K.T.D. Networks and Epidemic Models. *J. R. Soc. Interface* **2005,** *2,* 295–307.

9. Center for Systems Science and Engineering (CSSE), Johns Hopkins University (JHU). *COVID-19 Dashboard*. https://coronavirus.jhu.edu/map.html (accessed 5 November 2021).

10. Moriarty, L.F.; Plucinski, M.M.; Marston, B.J.; Kurbatova, E.V.; Knust, B.; Murray, E.L.; Pesik, N.; Rose, D.; Fitter, D.; Kobayashi, M.; et al. Public Health Responses to COVID-19 Outbreaks on Cruise Ships–Worldwide, February–March 2020. *MMWR Morb. Mortal. Wkly Rep.* **2020,** *69* (12), 347–352.

11. Centers for Disease Control and Prevention. *Real-Time RT-PCR Diagnostic Panel. Fact Sheet for Healthcare Providers.* December 1, 2020. https://www.cdc.gov/coronavirus/2019-ncov/downloads/Factsheet-for-Healthcare-Providers-2019-nCoV.pdf (accessed 5 November 2021).

12. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 2019. https://www.R-project.org/ (accessed 5 November 2021).

13. Soetaert, K.; Petzoldt, T.; Setzer, R.W. Solving Differential Equations in R: Package deSolve. *J. Stat. Soft.* **2010,** *33* (9), 1–25. DOI: 10.18637/jss.v033.i09.

14. New York Department of Health. *New York State Statewide COVID-19 Testi*ng. https://health.data.ny.gov/ (accessed 5 November 2021).

15. Wood, S.N. *Generalized Additive Models, An Introduction with R*, 2nd ed.; Chapman and Hall: Boca Raton, FL, 2017.

16. van Dorp, L.; Acman, M.; Richard, D.; Shaw, L.P.; Ford, C.E.; Ormond, L.; Owen, C.J.; Pang, J.; Tan, C.C.S.; Boshier, F.T.A.; et al. Emergence of Genomic Diversity and Recurrent Mutations in SARS-CoV-2. *Infect. Genet. Evol.* **2020,** *83*, 104351. DOI: 10.1016/j.meegid.2020.104351.

ACRONYMS AND ABBREVIATIONS

COVID-19	coronavirus disease 2019
E	the exposed class
EDDIS	estimators for disease dynamics with imperfect surveillance
GAMM	generalized additive mixed-model
I	the infectious class
PCR	polymerase chain reaction
r	rate of new infections
R	recovered class
R_e	effective reproductive number
R_0	basic reproductive number
S	susceptible class
SARS	severe acute respiratory syndrome
SEIR	susceptible, exposed, infectious, and recovered

Blank

APPENDIX

COMPUTER CODES

```
## Network SEIR, a script in R
## Written by Tom Ingersoll, 2/28/2021
## analyses data from the Diamond Princess CoViD19 outbreak
## Extract the data from where it is stored
## (D:\\Data on this computer)
DATA1<-read.table("D:\\Data\\DP_No_na.txt", header=T)
## DATA1<-as.matrix(DATA1)
attach(DATA1)
## display the data
DATA1
##    Day Count
## 1    0     0
## 2    1     0
## 3    2     0
## 4    3    10
## 5    4    20
## 6    5    41
## 7    6    61
## 8    7    64
## 9    9   135
## 10  11   161
## 11  12   175
## 12  13   218
## 13  15   285
## 14  16   355
## 15  17   454
## 16  18   542
## 17  19   621
## 18  20   634
## 19  23   691
## 20  26   705
## 21  27   705
## load the library
library(deSolve)
## Exposed, a function written by Tom Ingersoll, January 2021
## Runs an SEIR compartmental model in library deSolve
Exposed <- function(t, state, parameters) {
with(as.list(c(state, parameters)), {
dS <- v - (Beta*S*I) - (u*S)
dE <- (Beta*S*I) - (o*E) - (u*E)
dI <- (o*E) - (g*I) - (u*I)
dR <- (g*I) - (u*R)
Beta<<-Beta
list(c(dS, dE, dI, dR))
})
}
## Counts, a function that tabulates output from function Exposed
## written by Tom Ingersoll, January 2021
Counts<-function(p){
ps<-seq(p,1, ((1/26)*(1-p))   )
COUNTS<-matrix(NA,length(times),6)
for(i in 1: length(times)){
```

```
COUNTS[i,1]<-out[i,1]
COUNTS[i,2]<-ceiling(out[i,3])
COUNTS[i,3]<-(ceiling(out[i,4])) + (ceiling(out[i,5]))
COUNTS[i,4]<-ps[i]
COUNTS[i,5]<-as.double(DATA1[(t1+i),2])
COUNTS[i,6]<-dbinom(as.double(DATA1[i,2]),(ceiling(out[i,4]))+ (ceiling(out[i,5])),  ps[i], log = FALSE)
COUNTS<<-COUNTS
}
OUT2<-matrix(NA, 1, 7)
OUT2[1,1]<-parameters[1]
OUT2[1,2]<-p
OUT2[1,3]<-t1
OUT2[1,4]<-t2
OUT2[1,5]<-ceiling(out[i,3])
OUT2[1,6]<-(ceiling(out[i,4])) + (ceiling(out[i,5]))
OUT2[1,7]<-prod(COUNTS[,6])
OUT2<<-OUT2
        write.table(OUT2, file = "D:\\Data\\Bugs.txt", append = TRUE, quote = TRUE, sep = " ",
      eol = "\n", na = "NA", row.names = FALSE,
      col.names = FALSE)
}
## load a vector of counts from Diamond Princess
DATAALL<-c(0  ,
0  ,
0  ,
10 ,
20 ,
41 ,
61 ,
64 ,
64 ,
135,
135,
161,
175,
218,
218,
285,
355,
454,
542,
621,
634,
634,
634,
691,
691,
691,
705,
705,
705,
705,
705,
705,
705,
705,
705,
705,
```

```
705,
705)
################################################
### Plot the daily case reports
A<- DATAALL[1:length(DATAALL)]
B<-seq(1,length(DATAALL),1)
plot(A, B, type="n", ylim=c(0,1000),xlim=c(0,40),
xlab="Day",
ylab="Cases",
axes=F)
lines(B,A, type="p", col="black")
axis(side=1, at = c(0,10,20,30,40), labels = c("0","10","20","30","40"), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000), labels = c("0","200","400","600","800","1000"),
tick = TRUE, line = NA, las=1, pos = 0)
## Plot the data
A<- Count
B<-Day
plot(A, B, type="n", ylim=c(0,1000),xlim=c(0,27),
xlab="Day",
ylab="Cases",
axes=F,
## ann=F,
)
lines(B,A, type="p", col="black")
axis(side=1, at = c(0,10,20,27), labels = c("0","10","20","27"), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000), labels = c("0","200","400","600","800","1000"),
tick = TRUE, line = NA, las=1, pos = 0)
################################################
### The failure of the well-mixed assumption
A<- DATAALL[1:length(DATAALL)]
B<-seq(1,length(DATAALL),1)
### plot with the data
plot(A, B, type="n", ylim=c(0,1000),xlim=c(0,40),
xlab="Day",
ylab="Cases",
axes=F,
## ann=F,
)
axis(side=1, at = c(0,10,20,30,40), labels = c("0","10","20","30","40"), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000), labels = c("0","200","400","600","800","1000"),
tick = TRUE, line = NA, las=1, pos = 0)
lines(B,A, type="p", col="black")
parameters <- c(Beta = 0.15,v = 0, u = 0, o = 0.2, g = 0)
state <- c(S = 620, E = 100, I = 1, R= 0)
times <- seq(0, 40, by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines(out[,1],out[,4], col="blue")
parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0)
state <- c(S = 720, E = 0, I = 1, R= 0)
times <- seq(0, 40, by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines(out[,1],out[,4], col="purple")
parameters <- c(Beta = 0.01,v = 0, u = 0, o = 0.2, g = 0)
state <- c(S = 720, E = 0, I = 1, R= 0)
times <- seq(0, 40, by = 1)
```

```
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines(out[,1],out[,4], col="green")
parameters <- c(Beta = 0.005,v = 0, u = 0, o = 0.2, g = 0)
state <- c(S = 720, E = 0, I = 1, R= 0)
times <- seq(0, 40, by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines(out[,1],out[,4], col="red")
legend(25,380, legend = c("Data","r = 0.15", "r = 0.1","r = 0.01","r = 0.005"),
pch=c(1,NA,NA,NA,NA),
lty=c(NA,"solid","solid","solid","solid"),
col=c("black","blue","purple","green","red"),
bty = "n")
########################################################################
### The metapopulation network models
## Adjusting r
A<- DATAALL[1:length(DATAALL)]
B<-seq(1,length(DATAALL),1)
plot(A, B, type="n", ylim=c(0,1000),xlim=c(0,40),
xlab="Day",
ylab="Cases",
axes=F)
axis(side=1, at = c(0,10,20,30,40), labels = c("0","10","20","30","40"), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000), labels = c("0","200","400","600","800","1000"),
tick = TRUE, line = NA, las=1, pos = 0)
lines(B,A, type="p", col="black")
t1=0
t2=13
parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 300, E = 0, I = 1, R = 0)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="red")
Counts(0.5)
t1=13
t2=19
parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 800-(25+62+215), E = 25, I = 62, R = 215)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="red")
Counts(0.5)
t1=19
t2=27
parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 750-(163+209+429), E = 163, I = 209, R = 429)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="red")
Counts(0.5)
t1=0
t2=13
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 300, E = 0, I = 1, R = 0)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="blue")
NS=as.double(((out[t2+1,3])+(out[t2+1,4])+(out[t2+1,5])))
```

```
e=as.double(out[t2+1,3])
i=as.double(out[t2+1,4])
r=as.double(out[t2+1,5])
t1=13
t2=19
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = (900-NS), E = e, I = i, R = r)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="blue")
NS=as.double(((out[t2-t1+1,3])+(out[t2-t1+1,4])+(out[t2-t1+1,5])))
e=as.double(out[t2-t1+1,3])
i=as.double(out[t2-t1+1,4])
r=as.double(out[t2-t1+1,5])
t1=19
t2=27
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(800-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="blue")
##############################################################################################
## Adjusting the nodes
A<- DATAALL[1:length(DATAALL)]
B<-seq(1,length(DATAALL),1)
plot(A, B, type="n", ylim=c(0,1000),xlim=c(0,40),
xlab="Day",
ylab="Cases",
axes=F)
axis(side=1, at = c(0,10,20,30,40), labels = c("0","10","20","30","40"), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000), labels = c("0","200","400","600","800","1000"),
tick = TRUE, line = NA, las=1, pos = 0)
lines(B,A, type="p", col="black")
t1=0
t2=13
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 300, E = 0, I = 1, R = 0)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="blue")
NS=as.double(((out[t2+1,3])+(out[t2+1,4])+(out[t2+1,5])))
## (CUI)
e=as.double(out[t2+1,3])
i=as.double(out[t2+1,4])
r=as.double(out[t2+1,5])
t1=13
t2=19
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = (900-NS), E = e, I = i, R = r)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="blue")
NS=as.double(((out[t2-t1+1,3])+(out[t2-t1+1,4])+(out[t2-t1+1,5])))
e=as.double(out[t2-t1+1,3])
i=as.double(out[t2-t1+1,4])
r=as.double(out[t2-t1+1,5])
t1=19
```

```
t2=27
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(800-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="blue")
t1=0
t2=13
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 270, E = 0, I = 1, R = 0)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="purple")
NS=as.double(((out[t2+1,3])+(out[t2+1,4])+(out[t2+1,5])))
e=as.double(out[t2+1,3])
i=as.double(out[t2+1,4])
r=as.double(out[t2+1,5])
ta=t1
tb=t2
t1=13
t2=19
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(800-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="purple")
NS=as.double(((out[t2-tb+1,3])+(out[t2-tb+1,4])+(out[t2-tb+1,5])))
e=as.double(out[t2-tb+1,3])
i=as.double(out[t2-tb+1,4])
r=as.double(out[t2-tb+1,5])
ta=t1
tb=t2
t1=19
t2=27
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(750-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines((out[,1]+t1),out[,4]+out[,5], col="purple")
##############################################################################################
### Maximum likelihood estimation of r and R0
`expit` <-
function(x){
exp(x)/(1+exp(x))
}
##Ts<-c(0,13,19,27)
##NEs<-c(900,270,800,750)
##INIT<-c(300, 0, 1, 0)
## parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0.2)
Ts<-c(0,13,19,27)
NEs<-c(900,300,900,800)
INIT<-c(300, 0, 1, 0)
parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0.2)
## Count
## a function that tabulates output from the network model
## written by Tom Ingersoll January 2021
Count<-function(Ts,NEs,INIT,parameters,DATA1){
COUNT<-matrix(NA,length(DATA1[,1]),7)
```

```
for(m in 1:(length(DATA1[,1]))){
COUNT[m,4]<-DATA1[m,2]
COUNT[m,1]<-DATA1[m,1]
}
t1=Ts[1]
t2=Ts[2]
state <- c(S = INIT[1], E = INIT[2], I = INIT[3], R = INIT[4])
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
for(m in 1:length(Day[Day<=t2])){
COUNT[m,2]<-1
COUNT[m,3]<-(out[out[,1]==Day[m],4]+out[out[,1]==Day[m],5])
START<<-m
}
for(k in 2:(length(Ts)-1)){
t1=Ts[k]
t2=Ts[k+1]
## parameters <- c(Beta = 0.1,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(NEs[k+1]-(out[length(out[,1]),3]+out[length(out[,1]),4]+out[length(out[,1]),5])),
E = as.double(out[length(out[,1]),3]), I = as.double(out[length(out[,1]),4]), R =
as.double(out[length(out[,1]),5]))
times <- seq(0, t2-t1, by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
for(m in 2:length(Day[Day>=t1&Day<=t2])){
COUNT[START+m-1,2]<-k
COUNT[START+m-1,3]<-(out[out[,1]==Day[m],4]+out[out[,1]==Day[m],5])
}
START<<-START+(length(Day[Day>=t1&Day<=t2])-1)
}
#####################################################
## find the by-row ML detection value
Ps<-c(0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)
for (s in 1:length(COUNT[,1])){
        BDs<-matrix(NA,length(Ps),2)
        for (t in 1:length(Ps)){
                BDs[t,1]<-dbinom(COUNT[s,4], ceiling(COUNT[s,3]), Ps[t], log = FALSE)
                BDs[t,2]<-Ps[t]
                BD<-max(BDs[,1], na.rm=TRUE)
                }
        COUNT[s,5]<-BDs[BDs[,1]==BD,2]
        }
## produce a generalized linear model for detction across the series
DET<-COUNT[,5]
DAY<-COUNT[,1]
GLM<-glm(DET~DAY, binomial(link = "logit"))
new <- data.frame(DAY = seq(0, 27, 1))
PREDDET<-predict(GLM, new)
DAYDET<-matrix(NA, length(PREDDET), 2)
D <- seq(0, 27, 1)
for(u in 1:length(PREDDET)){
        DAYDET[u,1]<-D[u]
        DAYDET[u,2]<-expit(PREDDET[u])
        }
DAYDET<<-DAYDET
for (s in 1:length(COUNT[,1])){
        COUNT[s,6]<-DAYDET[DAYDET[,1]==COUNT[s,1],2]
        }
for (s in 1:length(COUNT[,1])){
```

```
            COUNT[s,7]<-dbinom(COUNT[s,4], ceiling(COUNT[s,3]), COUNT[s,6], log = FALSE)
            }
OUT<-matrix(NA,1,2)
OUT[1]<-Beta
OUT[2]<-prod(COUNT[,7])
OUT<<-OUT
COUNT<<-COUNT
write.table(OUT, file = "D:\\Data\\Bugs2.txt",
append = TRUE, quote = FALSE, sep =  ,
eol = "\n", na = "NA", row.names = FALSE,
col.names = FALSE)
}
# for the maximum likelihood value
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
Count(Ts,NEs,INIT,parameters,DATA1)
## a warning will display, whic can be ignored
## Display the table
COUNT
##      [,1] [,2]      [,3] [,4]  [,5]     [,6]        [,7]
##  [1,]    0    1   1.000000    0 0.001 0.2780713 7.219287e-01
##  [2,]    1    1   1.733885    0 0.001 0.3129746 4.720039e-01
##  [3,]    2    1   4.522241    0 0.001 0.3501341 1.159094e-01
##  [4,]    3    1  11.862783   10 0.800 0.3892062 1.963952e-03
##  [5,]    4    1  28.523373   20 0.700 0.4297551 2.948657e-03
##  [6,]    5    1  58.068564   41 0.700 0.4712689 2.713326e-04
##  [7,]    6    1  95.563870   61 0.600 0.5131841 4.582095e-03
##  [8,]    7    1 131.618144   64 0.500 0.5549146 1.884885e-02
##  [9,]    9    1 187.342322  135 0.700 0.6355606 3.649677e-03
## [10,]   11    1 224.812312  161 0.700 0.7092504 5.759823e-02
## [11,]   12    1 238.622765  175 0.700 0.7426037 5.443041e-02
## [12,]   13    1 249.929803  218 0.900 0.7733536 2.826205e-05
## [13,]   15    2 328.322553  285 0.900 0.8267755 9.517904e-03
## [14,]   16    2 430.359575  355 0.800 0.8495076 1.715645e-02
## [15,]   17    2 515.478060  454 0.900 0.8697266 4.249600e-02
## [16,]   18    2 585.180025  542 0.900 0.8875886 6.459281e-04
## [17,]   19    2 642.247191  621 0.900 0.9032739 6.575564e-10
## [18,]   20    3 711.686331  634 0.900 0.9169753 2.345789e-03
## [19,]   23    3 727.700408  691 0.900 0.9481098 6.660086e-02
## [20,]   26    3 740.806124  705 0.900 0.9679766 3.973759e-03
## [21,]   27    3 751.536145  705 0.900 0.9727888 1.312838e-07
## Relationship between R0 and r (Beta)
## for demonstration and boundary values
## D=10
## R0<-c(1.0, 1.1, 1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9)
## r = (R0-1)/D
## r
## [1] 0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
## we use the CDC value for inectious period
## Clinical Questions about COVID-19: Questions and Answers
## Updated Feb. 22, 2021
## https://www.cdc.gov/coronavirus/2019-ncov/hcp/faq.html#Transmission
## note this may err on the long side for caution
D=10
## Anderson & May 1991, Infectious Diseases of Humans, Dynamics and Control, pg 19:
Beta = (R0-1)/D
## implies
R0=(Beta*D)+1
### estimate a maximum likelihood value for Beta
```

```
Ts<-c(0,13,19,27)
NEs<-c(900,300,900,800)
INIT<-c(300, 0, 1, 0)
BETAS<-c(0.021, 0.022, 0.023, 0.024, 0.025, 0.026, 0.027, 0.028, 0.029, 0.030,
0.031, 0.032, 0.033, 0.034, 0.035, 0.036, 0.037, 0.038, 0.039, 0.040,
0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11,
0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19)
for (w in 1: length(BETAS)){
parameters <- c(Beta = BETAS[w],v = 0, u = 0, o = 0.2, g = 0.2)
Count(Ts,NEs,INIT,parameters,DATA1)
}
DATA2<-read.table("D:\\Data\\Bugs2.txt", header=T)
## DATA1<-as.matrix(DATA1)
attach(DATA2)
DATA2
names(DATA2)
## [1] "BETA" "LIK"
D=10
A<- (BETA*D)+1
B<-log(LIK)
plot(A, B, type="n", ylim=c(min(B),max(B)),xlim=c(1.20,3.0),
xlab="R0",
ylab="Log-likelihood",
axes=F,
## ann=F,
)
axis(side=1, at = c(1.21,1.5,2.0,2.5,2.9), labels = c("1.2","1.5","2.0","2.5","3.0"), tick = TRUE, line = NA,
pos = -158.2919)
axis(side=2, at = c(-158.2919,-150,-140,-130,-121.626), labels = c("","-150.0","-140.0","-130.0",""),
tick = TRUE, line = NA, las=1, pos = 1.21)
lines(A,B, type="l", col="black")
##############################################################################################
####
## the maximum likelihood detection values and detection model
A<- COUNT[,5]
B<-COUNT[,1]
C<-COUNT[,6]
plot(A, B, type="n", ylim=c(0,1),xlim=c(0,27),
xlab="Day",
ylab="Detection probability",
axes=F)
axis(side=1, at = c(0,10,20,27), labels = c("0","10","20",""), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,0.2,0.4,0.6,0.8,1.0), labels = c("0","0.2","0.4","0.6","0.8","1.0"),
tick = TRUE, line = NA, las=1, pos = 0)
lines(B,A, type="p", col="black")
lines(B,C, type="l", col="red")
legend(15,0.3, legend = c("Daily estimate","Model"),
pch=c(1,NA),
lty=c(NA,"solid"),
col=c("black","red"),
bty = "n")
## the pointwise model likelihoods
A<- COUNT[,7]
B<-COUNT[,1]
plot(-(log(A)), B, type="n", ylim=c(0,20),xlim=c(0,27),
xlab="Day",
ylab="- Log likelihood",
```

```
axes=F)
axis(side=1, at = c(0,10,20,27), labels = c("0","10","20",""), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,5,10,15,20), labels = c("0","5","10","15","20"),
tick = TRUE, line = NA, las=1, pos = 0)
lines(B,-(log(A)), type="l", col="black")
#####################################################
## perform a simulation  using network values estimated
## in the purple model above for the Diamond Princess node and link
## for a hypothetical shipboard population
t1=0
t2=13
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 400, E = 0, I = 1, R = 0)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
## lines((out[,1]+t1),out[,4]+out[,5], col="purple")
NS=as.double(((out[t2+1,3])+(out[t2+1,4])+(out[t2+1,5])))
e=as.double(out[t2+1,3])
i=as.double(out[t2+1,4])
r=as.double(out[t2+1,5])
ta=t1
tb=t2
SEIR<-matrix(NA,28,8)
for(z in 1:14){
SEIR[z,1]<-DAYDET[z,1]
SEIR[z,2]<-out[z,2]
SEIR[z,3]<-out[z,3]
SEIR[z,4]<-out[z,4]
SEIR[z,5]<-out[z,5]
SEIR[z,6]<-(out[z,4]+out[z,5])
SEIR[z,7]<-DAYDET[z,2]
SEIR[z,8]<-DAYDET[z,2]*(out[z,4]+out[z,5])
}
t1=13
t2=19
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(1400-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
## lines((out[,1]+t1),out[,4]+out[,5], col="purple")
NS=as.double(((out[t2-tb+1,3])+(out[t2-tb+1,4])+(out[t2-tb+1,5])))
e=as.double(out[t2-tb+1,3])
i=as.double(out[t2-tb+1,4])
r=as.double(out[t2-tb+1,5])
ta=t1
tb=t2
for(z in 1:(t2-t1)){
SEIR[z+14,1]<-DAYDET[z+14,1]
SEIR[z+14,2]<-out[z,2]
SEIR[z+14,3]<-out[z,3]
SEIR[z+14,4]<-out[z,4]
SEIR[z+14,5]<-out[z,5]
SEIR[z+14,6]<-(out[z,4]+out[z,5])
SEIR[z+14,7]<-DAYDET[z+14,2]
SEIR[z+14,8]<-DAYDET[z+14,2]*(out[z,4]+out[z,5])
}
t1=19
```

```
t2=27
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(1200-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
## lines((out[,1]+t1),out[,4]+out[,5], col="purple")
for(z in 1:(t2-t1)){
SEIR[z+20,1]<-DAYDET[z+20,1]
SEIR[z+20,2]<-out[z,2]
SEIR[z+20,3]<-out[z,3]
SEIR[z+20,4]<-out[z,4]
SEIR[z+20,5]<-out[z,5]
SEIR[z+20,6]<-(out[z,4]+out[z,5])
SEIR[z+20,7]<-DAYDET[z+20,2]
SEIR[z+20,8]<-DAYDET[z+20,2]*(out[z,4]+out[z,5])
}
A<- SEIR[1:length(SEIR[,1]),1]
B<-SEIR[1:length(SEIR[,1]),6]
C<-SEIR[1:length(SEIR[,1]),8]
plot(B, A, type="n", ylim=c(0,1200),xlim=c(0,30),
xlab="Day",
ylab="Cases",
axes=F)
lines(SEIR[,1],SEIR[,6], col="purple")
lines(SEIR[,1],SEIR[,8], col="orange")
axis(side=1, at = c(0,10,20,30), labels = c("0","10","20","30"), tick = TRUE, line = NA,
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000,1200),
labels = c("0","200","400","600","800","1000", "1200"),
tick = TRUE, line = NA, las=1, pos = 0)
legend(17,400, legend = c("Predicted","Underreported data"),
cex = 0.8,
lty=c("solid","solid"),
col=c("purple","orange"),
bty = "n")
#####################################################
## perform a simulation  using R0 estimated above
## and a hypothetical village population
## and a larger population
t1=0
t2=13
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = 40, E = 0, I = 1, R = 0)
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
## lines((out[,1]+t1),out[,4]+out[,5], col="purple")
NS=as.double(((out[t2+1,3])+(out[t2+1,4])+(out[t2+1,5])))
e=as.double(out[t2+1,3])
i=as.double(out[t2+1,4])
r=as.double(out[t2+1,5])
ta=t1
tb=t2
SEIR<-matrix(NA,28,8)
for(z in 1:14){
SEIR[z,1]<-DAYDET[z,1]
SEIR[z,2]<-out[z,2]
SEIR[z,3]<-out[z,3]
SEIR[z,4]<-out[z,4]
```

```
SEIR[z,5]<-out[z,5]
SEIR[z,6]<-(out[z,4]+out[z,5])
SEIR[z,7]<-DAYDET[z,2]
SEIR[z,8]<-DAYDET[z,2]*(out[z,4]+out[z,5])
}
t1=13
t2=19
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(400-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
## lines((out[,1]+t1),out[,4]+out[,5], col="purple")
NS=as.double(((out[t2-tb+1,3])+(out[t2-tb+1,4])+(out[t2-tb+1,5])))
e=as.double(out[t2-tb+1,3])
i=as.double(out[t2-tb+1,4])
r=as.double(out[t2-tb+1,5])
ta=t1
tb=t2
for(z in 1:(t2-t1)){
SEIR[z+14,1]<-DAYDET[z+14,1]
SEIR[z+14,2]<-out[z,2]
SEIR[z+14,3]<-out[z,3]
SEIR[z+14,4]<-out[z,4]
SEIR[z+14,5]<-out[z,5]
SEIR[z+14,6]<-(out[z,4]+out[z,5])
SEIR[z+14,7]<-DAYDET[z+14,2]
SEIR[z+14,8]<-DAYDET[z+14,2]*(out[z,4]+out[z,5])
}
t1=19
## (CUI)
t2=27
parameters <- c(Beta = 0.025,v = 0, u = 0, o = 0.2, g = 0.2)
state <- c(S = as.double(3000-NS), E = as.double(e), I = as.double(i), R = as.double(r))
times <- seq(0, (t2-t1), by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
## lines((out[,1]+t1),out[,4]+out[,5], col="purple")
for(z in 1:(t2-t1)){
SEIR[z+20,1]<-DAYDET[z+20,1]
SEIR[z+20,2]<-out[z,2]
SEIR[z+20,3]<-out[z,3]
SEIR[z+20,4]<-out[z,4]
SEIR[z+20,5]<-out[z,5]
SEIR[z+20,6]<-(out[z,4]+out[z,5])
SEIR[z+20,7]<-DAYDET[z+20,2]
SEIR[z+20,8]<-DAYDET[z+20,2]*(out[z,4]+out[z,5])
}
A<- SEIR[1:length(SEIR[,1]),1]
B<-SEIR[1:length(SEIR[,1]),6]
C<-SEIR[1:length(SEIR[,1]),8]
plot(B, A, type="n", ylim=c(0,1200),xlim=c(0,30),
xlab="Day",
ylab="Cases",
axes=F,
## ann=F,
)
lines(SEIR[,1],SEIR[,6], col="purple")
lines(SEIR[,1],SEIR[,8], col="orange")
axis(side=1, at = c(0,10,20,30), labels = c("0","10","20","30"), tick = TRUE, line = NA,
```

```
pos = 0)
axis(side=2, at = c(0,200,400,600,800,1000,1200),
labels = c("0","200","400","600","800","1000", "1200"),
tick = TRUE, line = NA, las=1, pos = 0)
### compare to a well-mixed population
A<- DATAALL[1:length(DATAALL)]
B<-seq(1,length(DATAALL),1)
parameters <- c(Beta = 0.0011,v = 0, u = 0, o = 0.2, g = 0)
state <- c(S = 3000, E = 0, I = 1, R= 0)
times <- seq(10, 40, by = 1)
out <- ode(y = state, times = times, func = Exposed, parms = parameters)
lines(out[,1],out[,4], col="blue")
## R0 for this model is
(0.0011*10)+1
## [1] 1.011
legend(5,1000, legend = c("Predicted","Underreported data", "Delayed well-mixed"),
cex = 0.8,
lty=c("solid","solid","solid"),
col=c("purple","orange","blue"),
bty = "n")
```

DISTRIBUTION LIST

The following individuals and organizations were provided with one Adobe portable document format (pdf) electronic version of this report:

U.S. Combat Capabilities Development
Command Chemical Biological Center
(DEVCOM CBC)
FCDD-CBR-IM
ATTN: Ingersoll, T.
Kierzewski, M.

Defense Threat Reduction Agency
DTRA-RD-IAR
ATTN: Batni, S.

DEVCOM CBC Technical Library
FCDD-CBR-L
ATTN: Foppiano, S.
Stein, J.

Defense Technical Information Center
ATTN: DTIC OA


U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND
CHEMICAL BIOLOGICAL CENTER