Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company’s public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
A stochastic process based modular tool-box for simulating COVID-19 infection spread

S.S. Manathunga\textsuperscript{a}, I.A. Abeyagunawardena\textsuperscript{b,c}, S.D. Dharmaratne\textsuperscript{c}

\textsuperscript{a} National Hospital of Sri Lanka, Sri Lanka
\textsuperscript{b} National Hospital of Sri Lanka, Home Address- 155, George E De Silva Mawatha, Kandy, Sri Lanka
\textsuperscript{c} Department of Community Medicine, Faculty of Medicine, University of Peradeniya, Sri Lanka, Institute for Health Metrics and Evaluation, Department of Health Metrics Sciences, School of Medicine, University of Washington, United States

\textbf{ARTICLE INFO}

\textbf{Keywords:}
Coronavirus
COVID-19
Disease modeling
COVID-19 simulation

\textbf{ABSTRACT}

\textbf{Background:} The novel coronavirus disease (COVID-19) culminated in a pandemic with many countries affected in varying stages. We aimed to develop a simulation environment for COVID-19 spread, taking environmental and social factors into account.

\textbf{Methods:} The program was written in R language. A stochastic point process simulation model for simulating epidemics, a maximum-likelihood estimation model, an exponential growth rate model for calculating the basic reproduction number (R\textsubscript{0}), and functions for generating graphical representations of the simulations were utilized.

Geographical area definition, population size, the number of initial infected individuals, period of simulation, parameters accounting for the radius of spread like masks usage, mobility level, intrinsic viral virulence, average infectious period, fraction of population vaccinated, time of vaccination, the efficacy of the vaccine, presence or absence of quarantine centers, time of establishment of quarantine centers, the efficacy of case detection and average time to quarantine from the detection of the infection were considered.

\textbf{Results:} When the defined parameters were input, the model performed successfully producing the epidemic curve, R\textsubscript{0} and an animation of infection spread. It was found that when parameters of known epidemics such as COVID-19 in California, Texas and, Florida were input, the epidemic curve generated was comparable to the epidemic curve in reality.

\textbf{Conclusion:} This model can be utilized by many countries to visualize the effects of various mitigation strategies applied in their stage of disease and for policy makers to make informed decisions. It is applicable to many infectious diseases and hence can be used for research and educational purposes.

1. Introduction

The novel coronavirus SARS-CoV-2, first detected in Wuhan, China in November 2019 spread rapidly globally culminating in a pandemic. The disease caused by this virus named as coronavirus disease-2019 (COVID-19) swept the world to unprecedented turmoil with the loss of countless lives. Multiple strategies were implemented by each nation to attempt to curb the spread of disease such as implementing health education on wearing masks and hand hygiene, lockdowns, strict curfews and, vaccination\textsuperscript{1}. The spread of COVID-19 varied from population to population due to the complex interplay of multiple factors.

We aimed to develop a simulation environment for COVID-19 spread, taking the complex interactions between environmental and social factors into account. The program was written in R language, and the source code is available at github\textsuperscript{2}. The main components of this tool-box are a stochastic point process simulation model for simulating epidemics, a maximum-likelihood estimation model, an exponential growth rate model for calculating the basic reproduction number, and functions for generating graphical representations of the simulations.

Conventional deterministic epidemiological models such as compartmentalized susceptible, infected, recovered (SIR) models have underlying assumptions of constant environmental factors namely homogeneous mixing of infected and susceptible populations\textsuperscript{3}. However, as we have witnessed during COVID-19 pandemic, these

\textsuperscript{*} Corresponding author.

\textit{E-mail addresses:} ss123ssm@gmail.com (S.S. Manathunga), ishanya1993@gmail.com (I.A. Abeyagunawardena), Samath.Dharmaratne@mbbs.md (S.D. Dharmaratne).

\url{https://doi.org/10.1016/j.imu.2022.100899}

Received 14 January 2022; Received in revised form 25 February 2022; Accepted 25 February 2022

Available online 5 March 2022

2352-9148/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
assumptions do not hold, due to various mitigation strategies implemented by governments like travel bans, partial and full lockdowns, quarantines, social distancing and, other non-pharmaceutical interventions[4].

Stochastic models have been used for simulations of the COVID-19 in previous papers. Xie proposed a stochastic point process based model which takes the average infection rate as a user defined input parameter [5]. Triambak et al. proposed a random-walk Monte Carlo simulation model, in which, the control interventions such as lockdown measures and mobility restrictions are incorporated through a single parameter namely, the size of each step in the random walk process[6]. Our proposed model is different from previous stochastic models due to its ability to account for various control interventions, environmental factors and, the ability to incorporate dynamic parameters which change throughout the simulation. Specifically, the ability to individually control the parameters i.e representing the fraction of the population vaccinated, the timing of the vaccination, efficacy of the vaccine, time of the establishment of quarantine centers, contact tracing efficacy, the average time to quarantine from the detection of the infection and the public mobility level is the most advantageous feature of the proposed model, which was not seen in previous stochastic models.

The function for our base model takes parameters for geographical

---

Fig. 1. Variation of reproduction number (R) with time in the predetermined population.

Fig. 2. The variation of the total number of cases with time in the predetermined population.
area definition, population size, the number of initially infected individuals, period of simulation, parameters accounting for the radius of spread like masks usage, mobility level, intrinsic viral virulence, average infectious period, the fraction of population vaccinated, time of vaccination, the efficacy of the vaccine, presence or absence of quarantine centers, time of establishment of quarantine centers, the efficacy of case detection and average time to quarantine from the detection of the infection. Other parameters are for the basic reproduction number (R0) calculation and graphical output generation.

This model is also unique due to its modular nature. Users can plug-in pre-defined modules like ‘vaccinate’, ‘quarantine’ and ‘restrict mobility’ to the model workflow at different points of time and also can define new modules by following the model convention. This functionality is a new feature seen in this model which may be of utmost use for policy making and to answer many difficult questions such as when to impose travel restrictions and whether going for a full lockdown will help in suppressing the infection at a certain time. Furthermore, it helps to determine which fraction of the population to be immediately vaccinated with a certain level of spread already present and whether increasing contact tracing and quarantine efficacy or increasing the fraction of vaccination is more effective in containing disease spread etc.

This program also calculates the basic reproduction number for each simulation and generates graphical representations of the simulated spread, including animations of the evolution of the infection allowing the user to visualize the implications of the conditions set in an attractive interface.

Fig. 3. The variation of total number of individuals in ‘infected’, ‘susceptible’ and ‘removed’ compartments with time in the predetermined population.

Fig. 4. The basic reproduction number (R0) of the predetermined population calculated using exponential growth rate method (EG) and maximum likelihood estimation method (ML).
2. Materials and methods

The toolbox consists of three main components; a stochastic process model for simulation, a maximum likelihood estimation model and an exponential growth rate model running parallel for R0 calculation, and a final graphics component. Parameters for all the components are taken into a single function called final(). The tunable parameters of the simulation model can be calibrated for a given epidemic curve of interest.

The geographical area of interest is represented as a two-dimensional ‘space’, which is divided into ‘cells’. Each cell represents a block of population. The model is initiated with random placement of the population across the space and the dimension of the space and the size of the population can be tuned by the user. By default, one infective block will be placed randomly at the time of model initiation, however, the number of infective blocks can also be customized. Each susceptible block is assigned to the compartment ‘S’ and infective blocks to the compartment ‘I’.

The number of people represented by a population block and the scale of the time-step of the iterations have to be decided on the context, considering the population density, area and, the duration of the simulation. Once the space is defined, a few other static model parameters need to be set, which we are termed ‘states’. These states are constant throughout the simulation period.

The ‘spread’ parameter determines the probability of transmitting the disease from one infected block, to a susceptible block, a unit block-length away, at a fixed time from the contraction of the infection by the infector. There are several factors that affect the spread parameter like the intrinsic virulence of the virus and protective measures such as wearing masks and hand washing.
The "infectious-period" (ip) is another parameter, which determines the total time duration an infected block can remain infectious. The model by default assumes a negative linear relationship between the time and infectivity for each infected cell, but a custom function for the infectivity over time can explicitly be defined by the user according to the following Equation (01). The default behaviour is to set

\[ \text{infectivity} = 1 - \frac{\text{day}}{\text{ip}} \]  

where 'day' is the number of time-steps elapsed since contraction of the infection. A conditional check for \( \text{infectivity} \geq 0 \) has been utilized to avoid negative values for the infectivity.

The probability of a cell being sampled as infected, is directly proportional to the values of the parameters 'spread' and 'infectivity' as further discussed later.

The 'radius' parameter (R) defines the maximum distance from an infected cell, from which the infection can spread. The model assumes that the probability for a susceptible block to contract the disease has a negative linear relationship with the distance from the nearest infected block and reaches zero at the distance of the radius parameter. Users can explicitly define a function for the probability of causing the disease against the radius parameter, such as a hyperbolic function assuming an inverse proportionality and use it instead of the default negative linear function.

Next the dynamic parameters are set with the help of a 'time-line' feature. These are special type of parameters, which can be changed...
during the simulation. There are three default modules for dynamic parameter classes; ‘vaccination’, ‘quarantine’ and, ‘mobility’. Each module is a function that takes several parameters, which are called attributes. The user can define the initial values for the attributes in each module and how they should change over time.

The vaccination module has three attributes; vaccination fraction, efficacy and time. It is possible to define at which point of the time-line the vaccination takes place and the efficacy of the particular vaccine. Upon invocation, the function randomly subsamples the fraction determined by the ‘vaccination fraction’ parameter from the susceptible population blocks. Only the percentage defined by the ‘vaccination efficacy’ will be assigned to the compartment ‘vaccinated’, which will be immune to the infection.

The quarantine module also has three attributes; the proportion of infected individuals that are quarantined, the average time to quarantine from the detection of the infection and, the time at which the quarantine centers were established after the detection of the first case. The function first estimates the number of time steps (st) which have to elapse for an infected cell block before transferring to the quarantine center. This time lag represents the delay in real-world case detection and contact tracing. However, the parameter that the model takes from the user is the infectivity (Inf) of an infector to trigger the quarantine function, which is a value ranging from zero to one. Assuming the infectivity of an infector decreases linearly with time during the infectious period, the threshold infectivity (Ti) is, recalculated within the function according to Equation (02) as follows.

Fig. 9. The variation of the total number of cases with time in the California population.

Fig. 10. The variation of total number of individuals in ‘infected’, ‘susceptible’ and ‘removed’ compartments with time in the California population.
The function then creates an array of infected individuals whose infectivity parameter is equal to \( Ti \), and randomly samples the fraction defined by the quarantine efficacy parameter by calling the function \( \text{sample}(\text{which}(\text{tab}\$\text{infectivity} = = \text{Ti}), (\text{efficacy} \times \text{length}(\text{which}(\text{tab}\$\text{infectivity} = = \text{Ti})))) \).

The position coordinates of the sampled array are immediately changed into the predefined coordinates of the quarantine center, which by default is the left bottom cell block. This process simulates transferring the detected infected individuals to a quarantine center within one time-step duration. It should be noted that this is likely to create a high-risk zone around the quarantine center from where a new wave of infection can arise. To avoid such circumstances, it is possible to transpose the coordinates of the quarantine center into coordinates out of the space boundaries if needed.

The mobility threshold attribute of the mobility module determines the probability of a block of population to move to one of neighboring cells and this stochastic process mimics public mobility during the pandemic. The parameter can take values from zero to one and can be tuned to simulate full lockdowns, partial lockdowns and high mobility states. The user also can define at which point each mobility restriction is to be imposed through time-line.

Once all the three key elements; the space, the static parameters and, the time-line of dynamic modules are defined, the model is ready for simulation. For prediction purposes, the initial model parameters to simulate an observed epidemic curve can be estimated using the ‘calibrate’ function. The function takes a data-frame containing different values of the tuning parameters and estimates optimal parameter values.
which minimize the Root of the Mean Squared Error (RMSE) for the cumulative caseload. Assuming the loess model predictions for cumulative caseload to be \( p_1, p_2, \ldots, p_n \) and observed caseload to be \( o_1, o_2, \ldots, o_n \), RMSE can be expressed as follows in Equation (03),

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - o_i)^2}{n}}
\]  

(3)

At the beginning of the simulation, a unique ID is assigned to each populated cell block. A data-frame is constructed which contains the details of each block including their compartment state (susceptible or infected), coordinates in the defined space, day of infection if infected and, the infectivity. Four functions are applied sequentially to the data-frame at each iteration, namely ‘Mobilize’, ‘Alter’, ‘Time-line’ and, ‘Snapshot’.

The ‘Mobilize’ function takes the data-frame as the argument and changes the coordinates of each populated cell block depending on the value of the mobility parameter. This is done by the construction of a two-dimensional spatial representation of the coordinate data. The function samples a random fraction of populated cells to mobilize determined by the mobility parameter, and for each candidate cell, randomly selects the direction of movement through x and y axes. The random sampling is based on the default Pseudo-random Number Generator Engine of R with the Mersenne twister method[7]. If that new position is an empty cell, the population block will be moved to that new coordinate. Edge handling and collision handling are also done within the function. Ultimately, the ‘Mobilize’ function returns the data-frame to the ‘Alter’ function.

The ‘Alter’ function iterates through each block and updates the cell-specific parameters. It calls several helper functions to determine the new compartment-state of each cell block to determine which new cells have become infected. For each infected cell block, the function stores the positions and states of the neighboring cells into a list, depending on the radius parameter (R). Then, the probability of each stored cell being infected is calculated, assuming the Euclidean distance from the respective cell to the infector to have a negative linear relationship with the probability of contracting the infection (\( P \))[8]. Assuming \( p \) and \( q \) being the coordinates of the infector and the cell of interest in the Euclidean n-space, the distance, \( d(p,q) \) can be calculated as in Equation (04),

\[
d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}
\]  

(4)

and the probability of contracting the disease (\( P \)) can be expressed as follows (Equation (05)),

\[
P = 1 - \frac{d(p,q)}{R} \text{ where } 0 \leq d(p,q) \leq R
\]  

(5)

The final compartment state of each stored cell is determined by calling sample(c(TRUE, FALSE), 1, prob = c(infectivity * spread * P, 1 - (infectivity * spread + P)))

For each cell, by using a vectorized wrapper function that returns a logical vector of the compartment states. The function then updates the infectivity and day of illness parameters of the data-frame and hands it over to the ‘Time-line’ function.

The ‘Time-line’ function keeps track of the elapsed time and checks if any time-dependent parameter change is necessary, and attaches the relevant modules to the workflow. For example, it checks if any quarantine center is to be opened, any mobility restriction to be imposed or, any vaccination campaign to be carried out according to the initial time-line setup defined by the user. Therefore, depending on the initial time-line settings, the function may call additional helper functions like ‘vaccinate’, ‘change-mobility’ or, ‘quarantine’ at this stage and will return the data-frame to the next step.

Ultimately, the ‘Snapshot’ function takes the data-frame and creates a graphical representation of the space. It also appends the data-frame’s spatial and dynamic data into a master-table, which can be used for tracking the disease spread throughout the period of simulation.

At the end of the simulation, the snapshots are combined to visualize the evolution of the infection.

Since the scale of the time-steps and the size of the population blocks are not fixed, the expanded epidemic curve has to be computed using the master table. A local regression (loess) model with user-defined span parameter is constructed to predict the caseload, using the time-step as the predictor [9]. The distance weights (w) of the model are calculated using the tri-cubic function as shown in Equation (06)

\[
w(x) = \begin{cases} 
(1 - |x|^3)^3 & |x| < 1 \\
0 & |x| \geq 1 
\end{cases}
\]  

(6)

The R0 for the simulated epidemic curve is then computed with the maximum likelihood estimation method, assuming that the number of secondary cases caused by an index case is Poisson distributed with expected value R0[10,11]. Given that the incidence cases generated by the loess model are \( N_0, N_1, \ldots, N_t \), and the generation time distribution \( g \), R0 is estimated by maximizing the log-likelihood in Equation (07).

\[
LL(R0) = \sum_{i=1}^{T} \log \left( \frac{e^{-\mu_i} \mu_i^{N_i}}{N_i!} \right) \text{ where } \mu_i = R0 \sum_{i=1}^{T} N_{i, g},
\]  

(7)

The R0 yielded is further corroborated by an exponential growth rate model, as \( R0 = \frac{1}{\mu - \alpha} \) where M is the moment generation function of the discretized generation time distribution and \( \alpha \) is the exponential growth rate of the epidemic curve[10]. Bootstrap resampling is used for confidence interval calculation.

The graphics component generates graphs of the epidemic curve including the dynamics of the susceptible and infected compartments.
over time, error bars of R0 and also saves an animation of the spread of the infection in the space in the form of a graphics interchange format (GIF) file. The frame rate and the number of frames of the animation can directly be fed into the main function by the user at the time of model initiation through ‘fps’ and ‘frames’ parameters.

3. Results

Firstly, an imaginary population of 4 million is selected. A block size of 10,000, a space of 26 × 26 cells and 65 time-steps, each of one-week length are set. One quarantine center will be opened by the 20th time-step with a case detection efficacy of 60%. Detected infective blocks will be quarantined when their infectivity reaches 70% from the maximum infectivity. 30% of the initial population is vaccinated with a vaccine having an efficacy of 80%. The mobility level is set at 0.25 for this simulation.

The snapshot of the events can be visualized in Animation 01 in the form of a GIF. The red dots represent infected population blocks, green is susceptible and, blue is vaccinated. The opacity intensity of each block in the animations is proportional to its infectivity.

Animation 01 (Right) depicts the number of individuals in each compartment, blue is susceptible, green is total infected and, red is daily infected cases with time. The variation of R, total infected cases, the epidemic curve with time and the basic reproduction numbers are depicted in Fig. 1, Fig. 2, Fig. 3 and, Fig. 4 respectively.

Taking this example scenario as a baseline, predictions of the spread of the infection in different conditions were attempted. For the second scenario, the vaccination coverage was increased up to 50% and the quarantine center was established at an early stage, at 10 weeks. The simulation is depicted in Animation 02 and the variation of R, total infected cases, the epidemic curve with time and the basic reproduction numbers are depicted in Fig. 5, Fig. 6, Fig. 7 and, Fig. 8 respectively.

The spread of the infection in the above two scenarios can be visualized and contrasted in Animation 03.

California reported the highest number of COVID-19 cases in the United States reported as 4.9 million by the end of October 2021. Nearly four million cases had been reported during the period starting from March 2020, to June 2021. The population of California by 2020 was estimated to be almost 40 million and more than 60% of the population had received at least one dose of COVID 19 vaccination by the end of June 2021[12].

The population of California was represented by 4000 blocks (block size of 10,000), in a 90 × 90 space and the simulation was carried out as depicted in Animation 04 and variation of the total number of cases with time and the epidemic curve is represented in Figs. 9 and 10 respectively.

Texas reported to have the second-highest number of cases, which reached 4.2 million by November 2021[13]. The space-size and population block sizes are calibrated to represent Texas. The model run with the default parameters is illustrated in Animation 05 and variation of the total number of cases with time and the epidemic curve is represented in Figs. 11 and 12 respectively.

3.6 million cases of COVID 19 were reported from Florida, which comes at third in the list. The population of Florida is estimated to be 21 million[14]. The simulation is depicted in Animation 06 after calibrating for space and population density and variation of the total number of cases with time and the epidemic curve is represented in Figs. 13 and 14 respectively.

4. Discussion

The model proposed has the ability to predict the epidemic curve of infectious diseases which spread like COVID-19, reflecting the outcome of diverse mitigation strategies. Furthermore, past epidemics could be analyzed and learnt from, providing a great scope for learning. It differs from other stochastic models published due to its ability to simulate the impact of the spread of disease with measures such as the implementation of lockdowns at different points in time, the impact of vaccination and the effect of vaccinating different proportions, the efficacy of quarantine centers and countless other scenarios accurately, which would be instrumental in decision making whilst battling COVID-19. It can be seen that countries worldwide are in different stages of the COVID-19 pandemic and therefore this model could be an essential tool for policy-makers worldwide to make informed decisions. The final model is composed of linearly stacked individually functioning modules, and each module has its own set of functions. The source code of the program is published on github. With new knowledge about the dynamics of the disease, the underlying model functions can be modified to keep the model updated for more accurate simulations. A limitation seen in stochastic process-based models namely being computationally heavy and hence requiring a higher processing power to achieve stable predictions with repeated simulations is also seen in this model. In future, more robust cross-validation techniques could be implemented as well as more efficient model parameter estimation methods using stochastic gradient descent. In addition, this model will be combined with a time series prediction model in the future, enabling forecasting of the epidemic curve beyond the simulation window.

5. Conclusion

This model has the ability to account for various control interventions, environmental factors and, the ability to incorporate dynamic parameters which change throughout the simulation while predicting the outcome by calculating the basic reproduction number, generating the epidemic curve and animation which depicts the spread of disease.

Therefore, it can be utilized as a tool for policy-makers to make informed decisions in the COVID-19 battle and provide a source to analyze past epidemics for research and educational purposes.

Ethics approval and consent to participate

Not applicable.

Consent to publication

Not applicable.

Availability of data and material

The source code utilized has been made available at github.

Funding

There was no funding source for this study.

Author contributions

SDD perceived the necessity of a model such as the one proposed during a pandemic such COVID-19. SSM extracted the data, wrote the code for this model, implemented the idea and contributed to the writing of the full paper. IAA contributed to the writing of the manuscript. All authors provided intellectual input, contributed to the final manuscript and approved the final version for publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
Acknowledgements

Not applicable.

List of abbreviations

COVID-19 Coronavirus disease-2019
SIR Susceptible, Infected, Recovered
R₀ Basic Reproduction Number
Ip Infectious period
R Radius
St Time steps
Inf Infectivity
Ti Threshold infectivity
RMSE Root of Mean Squared Error
P Probability
M Moment generating function
r exponential growth rate
GIF Graphics interchange format

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.imu.2022.100899.

References

[1] World Health Organization. Advice for the public: coronavirus disease (COVID-19). World Health Organization, 2021 https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public (accessed Nov 10, 2021).
[2] Manathunga SS, contagion. 2021 https://github.com/ssm123sm/contagion (accessed Nov 7, 2021).
[3] Cooper I, Mondal A, Antonopoulos CG. A SIR model assumption for the spread of COVID-19 in different communities. Chaos, Solit Fractals 2020;139:110057.
[4] Mukerjee S, Chow CM, Li M. Mitigation strategies and compliance in the COVID-19 fight; how much compliance is enough? PLoS One 2021;16:e0239352.
[5] Xin G. A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time. Sci Rep 2020;10:1–9.
[6] Triambak S, Mahapatra DP. A random walk Monte Carlo simulation study of COVID-19-like infection spread. Phys Met 2021;574:126014.
[7] Matsumoto M, Nishimura T. Marsenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans Model Comput Simulat 1998;8:3–30.
[8] Gower JC. Properties of Euclidean and non-Euclidean distance matrices. Linear Algebra Its Appl 1985;67:81–97.
[9] Cleveland WS, Devlin SJ. Locally weighted regression: an approach to regression analysis by local fitting. J Am Stat Assoc 1988;83:596–610.
[10] Dharmaratne S, Sudaraka S, Abeyagunawardena I, Manchanayake K, Kothalawala M, Gunathunga W. Estimation of the basic reproduction number (R₀) for the novel coronavirus disease in Sri Lanka. Virol J 2020;17:144.
[11] Obadia T, Haneef R, Boile P-Y. The R₀ package: a toolbox to estimate reproduction numbers for epidemic outbreaks. BMC Med Inf Decis Making 2012;12:147.
[12] California S of. Tracking COVID-19 in California./ (accessed Nov 7, 2021).
[13] Texas COVID-19 data. (accessed Nov 7, 2021).
[14] Home - Florida department of health COVID-19 outbreak. (accessed Nov 7, 2021).