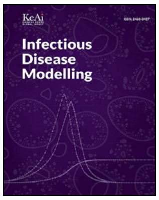
Infectious Disease Modelling 9 (2024) 411–436

Contents lists available at ScienceDirect

Infectious Disease Modelling

journal homepage: www.keaipublishing.com/idm



SubEpiPredict: A tutorial-based primer and toolbox for fitting and forecasting growth trajectories using the ensemble *n*-sub-epidemic modeling framework

Gerardo Chowell [a], [b], [*], Sushma Dahal [a], Amanda Bleichrodt [a], Amna Tariq [c], James M. Hyman [d], Ruiyan Luo [a]

[a] *Department of Population Health Sciences, School of Public Health, Georgia State University, Atlanta, GA, USA*
[b] *Department of International Epidemiology and Population Studies, Fogarty International Center, National Institutes of Health, Bethesda, MD, USA*
[c] *Department of Pediatrics, School of Medicine, Stanford University, Palo Alto, CA, USA*
[d] *Department of Mathematics, Center for Computational Science, Tulane University, New Orleans, LA, USA*

ARTICLE INFO

Article history:
Received 13 October 2023
Received in revised form 1 February 2024
Accepted 1 February 2024
Available online 9 February 2024
Handling Editor: Dr Yijun Lou

Keywords:
n-Sub-epidemic model
Forecasting
MATLAB
Sub-epidemics
Phenomenological models
Performance metrics
Model evaluation

ABSTRACT

An ensemble *n*-sub-epidemic modeling framework that integrates sub-epidemics to capture complex temporal dynamics has demonstrated powerful forecasting capability in previous works. This modeling framework can characterize complex epidemic patterns, including plateaus, epidemic resurgences, and epidemic waves characterized by multiple peaks of different sizes. In this tutorial paper, we introduce and illustrate *SubEpiPredict*, a user-friendly MATLAB toolbox for fitting and forecasting time series data using an ensemble *n*-sub-epidemic modeling framework. The toolbox can be used for model fitting, forecasting, and evaluation of model performance of the calibration and forecasting periods using metrics such as the weighted interval score (WIS). We also provide a detailed description of these methods including the concept of the *n*-sub-epidemic model, constructing ensemble forecasts from the top-ranking models, etc. For the illustration of the toolbox, we utilize publicly available daily COVID-19 death data at the national level for the United States. The MATLAB toolbox introduced in this paper can be very useful for a wider group of audiences, including policymakers, and can be easily utilized by those without extensive coding and modeling backgrounds.

© 2024 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Reliable short-term forecasts of diverse growth processes unfolding in nature and society from predicting the weather to the spread of epidemics are crucial in decision making. In the context of infectious diseases, short-term predictions can guide prevention and mitigation interventions, as well as resource allocation. While simple statistical time-series forecasting approaches such as auto-regressive integrated moving average models (ARIMA) are widely used (Dimri et al., 2020; Hyndman &

[*] Corresponding author. Department of Population Health Sciences, School of Public Health, Georgia State University, Atlanta, GA, USA.
E-mail address: gchowell@gsu.edu (G. Chowell).
Peer review under responsibility of KeAi Communications Co., Ltd.

https://doi.org/10.1016/j.idm.2024.02.001

2468-0427/© 2024 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Athanasopoulos, 2018; Mondal et al., 2014; Shamsnia et al., 2011; Tektaş, 2010), forecasts from dynamical models that aim to infer a mixture of underlying growth components may outperform classic time-series forecasting methods and provide insights into the nature of the dynamics governing short-term growth or decline (Bleichrodt et al., 2023; Chowell et al., 2019, 2022). In particular, the ensemble *n*-sub-epidemic framework included in this tutorial is a recently developed ensemble modeling approach that integrates sub-epidemics to capture complex temporal dynamics and has yielded competitive forecasts of the trajectory of the coronavirus disease 2019 (COVID-19) and monkeypox (mpox) epidemics (Bleichrodt et al., 2023; Chowell et al., 2022).

This mathematical framework characterizes time-series by aggregating multiple asynchronous growth processes and has previously outperformed simpler growth models (Chowell et al., 2019; Tariq et al., 2022). Different growth curves can start at different time points and may follow different growth rates, scaling of growth, and final sizes. Hence, this ensemble modeling framework can characterize plateaus, resurgences, and waves characterized by multiple peaks of different sizes. To facilitate their application by non-specialists, there is an urgent need for an easy-to-use and flexible toolbox that requires minimal understanding of the mathematics behind the model and coding skills.

In this tutorial-based primer, we introduce a user-friendly MATLAB (TheMathWorks Inc, 2022) toolbox to fit and forecast the growth trajectories of infectious disease epidemics and pandemics using the ensemble *n*-sub-epidemic modeling framework (Chowell et al., 2022). The toolbox is not only useful for studying the dynamics of infectious diseases but can also be applied to a wide range of other natural and social science studies. Here, we provide a detailed step-by-step guide to demonstrate how to prepare the data for analysis, use different options in the toolbox to generate fitting and forecasting results, and interpret the findings. The toolbox can be useful to audiences such as policy makers and researchers interested in fitting models to time series data, estimation of parameters with quantified uncertainty, generating short term forecasts derived from the top-ranked and ensemble models, and evaluating performance of the models during the calibration and forecast period. In addition, the users can fit the models using different parameter estimation approaches, under different error structure assumptions, and select the underlying function for the sub-epidemic building block such as the generalized-logistic growth model (GLM), Richards model, and the generalized Richards model (GRM). They can also choose whether the sub-epidemics start synchronously at time 0 or asynchronously. We illustrate these functions in the toolbox using daily COVID-19 deaths data from the United State of America (USA), and a tutorial video that demonstrates the different functions is available at: https://www.youtube.com/channel/UC6IzIu-pPcMLlLYAho43loQ.

2. Methods

In this section, we provide an overview of the toolbox functions and describe the methods implemented in this toolbox.

2.1. Installing the toolbox

- Download the MATLAB code located in folder ‘ensemble *n*-subepidemic code v1.0’ from the GitHub repository: https://github.com/gchowell/ensemble_n-subepidemic_framework.
- Create an ‘input’ folder in your working directory where your input data will be stored.
- Create an ‘output’ folder in your working directory where the output files will be stored.
- Open a MATLAB session.

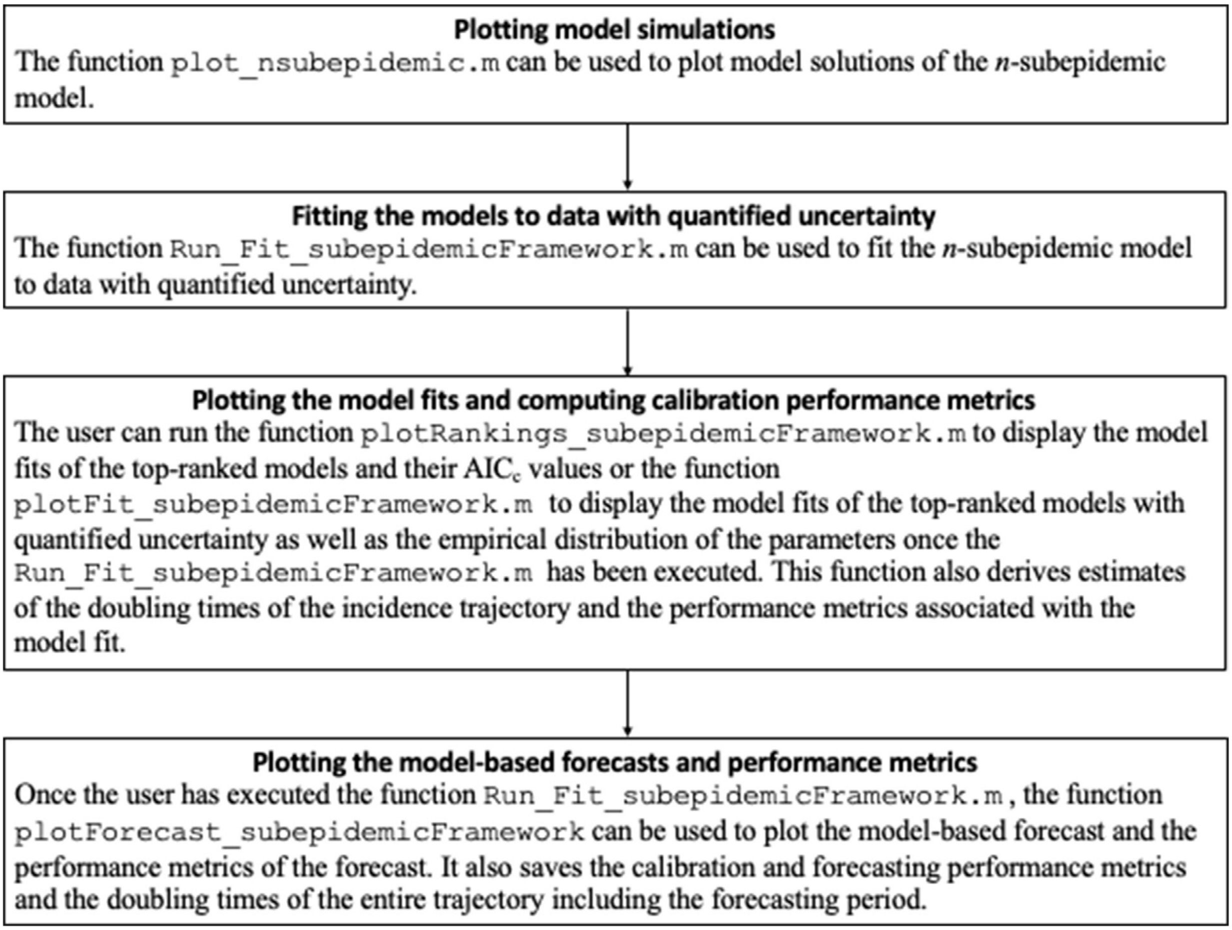
2.2. Overview of the toolbox functions

The workflow described in this tutorial, summarized in Fig. 1, is composed of 4 main sections: 1) plotting model simulations, 2) fitting the models to data with quantified uncertainty, 3) plotting the resulting model fits and calibration performance metrics, and 4) plotting model-based forecasts and the associated forecasting performance metrics. Table 1 lists the names of user functions associated with the toolbox along with a brief description of their roles. The internal functions associated with the toolbox are given in Appendix A (Table A.1). In the `options_fit.m` and `options_forecast.m` functions, the user can specify the parameters related to model fitting and forecasting as described in the toolbox tutorial (Section 3).

2.3. The n-sub-epidemic model

2.3.1. The building block growth model

We model epidemic trajectories consisting of one or more overlapping and asynchronous sub-epidemics. That is, the sub-epidemics are used as building blocks to characterize more complex epidemic trajectories. The mathematical equation for the



Plotting model simulations

The function `plot_nsubepidemic.m` can be used to plot model solutions of the *n*-subepidemic model.

Fitting the models to data with quantified uncertainty

The function `Run_Fit_subepidemicFramework.m` can be used to fit the *n*-subepidemic model to data with quantified uncertainty.

Plotting the model fits and computing calibration performance metrics

The user can run the function `plotRankings_subepidemicFramework.m` to display the model fits of the top-ranked models and their AIC$_c$ values or the function `plotFit_subepidemicFramework.m` to display the model fits of the top-ranked models with quantified uncertainty as well as the empirical distribution of the parameters once the `Run_Fit_subepidemicFramework.m` has been executed. This function also derives estimates of the doubling times of the incidence trajectory and the performance metrics associated with the model fit.

Plotting the model-based forecasts and performance metrics

Once the user has executed the function `Run_Fit_subepidemicFramework.m`, the function `plotForecast_subepidemicFramework` can be used to plot the model-based forecast and the performance metrics of the forecast. It also saves the calibration and forecasting performance metrics and the doubling times of the entire trajectory including the forecasting period.

Fig. 1. Overview of the workflow for fitting and forecasting time series trajectories using the *n*-sub-epidemic model.

Table 1
Description of user functions available in the toolbox.

Function	Role
`options.m`	Specifies the parameters related to model fitting including the characteristics of the time series data, the sub-epidemic model, parameter estimation method, error structure, smoothing, and calibration period.
`options_forecast.m`	Specifies the parameters related to the forecast including the forecasting period, the type of ensemble weight for the ensemble models, and whether the forecasts will be evaluated.
`plot_nsubepidemic.m`	Plots simulations of the *n*-sub-epidemic model.
`Run_Fit_subepidemicFramework.m`	Derives the top-ranking sub-epidemic models to data with quantified uncertainty.
`plotRankings_subepidemicFramework.m`	Plots the mean model fits of the top-ranking models including their sub-epidemic profiles and the associated quality of model fit metrics including the Akaike Information Criterion (AIC_c), the relative likelihood, and the evidence ratio.
`plotFit_subepidemicFramework.m`	Displays the model fit and 95% prediction interval as well as the empirical distribution of the parameters. It also saves output .csv files in the output folder with the model fit, the parameter estimates including 95% confidence intervals, and the calibration performance metrics.
`plotForecast_subepidemicFramework.m`	Displays the model-based forecast and the performance metrics of the forecast. Moreover, the data associated with the forecasts, the parameter estimates, as well as the calibration and forecasting performance metrics, are saved as .csv files in the output folder.

sub-epidemic building block is the 3-parameter generalized-logistic growth model (GLM), which is specified by setting the parameter `<flag1>=1` in the `options.m` file. This growth model has performed well in short-term forecasts of single outbreak trajectories for different infectious diseases, including COVID-19 (Chowell et al., 2016; Chowell et al., 2024; Pell et al., 2018; Shanafelt et al., 2018; Yan & Chowell, 2019). The GLM is given by the following differential equation:

$$\frac{dC(t)}{dt} = C'(t) = rC^p(t)\left(1 - \frac{C(t)}{K_0}\right), \tag{1}$$

where $C(t)$ denotes the cumulative curve, $\frac{dC(t)}{dt}$ describes the curve of daily incidences (here deaths for our COVID-19 data) over time t. The positive parameter r denotes the growth rate per unit of time, K_0 is the final outbreak size, and $p \in [0,1]$ is the "scaling of growth" parameter which allows the model to capture early sub-exponential and exponential growth patterns. If $p = 0$, this equation describes a constant number of new deaths over time, while $p = 1$ indicates that the early growth phase is exponential. Intermediate values of p $(0 < p < 1)$ describe early sub-exponential (e.g., polynomial) growth dynamics. Alternative growth equations to model the sub-epidemic building block include the 3-parameter Richards model (`<flag1>=4`) and the 2-parameter logistic growth model (`<flag1>=2`).

2.3.2. The number of sub-epidemics

An n-sub-epidemic trajectory comprises n overlapping sub-epidemics and is given by the following system of coupled differential equations:

$$\frac{dC_i(t)}{dt} = C_i{}'(t) = A_i(t) r_i C_i^{p_i}(t)\left(1 - \frac{C_i(t)}{K_{0i}}\right), \tag{2}$$

where $C_i(t)$ tracks the cumulative number of deaths for sub-epidemic i, and the parameters that characterize the shape of the i_{th} sub-epidemic are given by (r_i, p_i, K_{0i}), for $i = 1, \ldots, n$. Thus, the 1-sub-epidemic model is equivalent to the 3-parameter GLM described above (1). When $n > 1$, we model the onset timing of the $(i+1)_{th}$ sub-epidemic, where $(i+1) \leq n$, by employing an indicator variable given by $A_i(t)$ so that the $(i+1)_{th}$ sub-epidemic is triggered when the cumulative curve of the i $_{\mathrm{th}}$ sub-epidemic exceeds a threshold value denoted as C_{thr}. We assume that the $(i+1)_{th}$ sub-epidemic is only triggered when $C_{thr} \leq K_{0i}$. Hence, we have:

$$A_i(t) = \begin{cases} 1, C_{i-1}(t) > C_{thr} \\ 0, \text{Otherwise} \end{cases} \textit{for } i = 2, \ldots n, \tag{3}$$

where $A_1(t) = 1$ for the first sub-epidemic. The total number of parameters that are needed to model an n-sub-epidemic trajectory scales linearly with the number of sub-epidemics and is given by $3n + 1$. The initial number of deaths is given by $C_1(0) = I_0$, where I_0 is the initial number of deaths in the observed data. The cumulative curve of the n-sub-epidemic trajectory is given by:

$$C_{tot}(t) = \sum_{i=1}^{n} C_i(t). \tag{4}$$

The n-sub-epidemic wave model can characterize diverse epidemic patterns, including epidemic plateaus where the epidemic stabilizes at a high level for an extended period, epidemic resurgences where incidence increases again after a low incidence period, and epidemic waves characterized by multiple peaks. The maximum number of sub-epidemics considered in the epidemic trajectory is specified using parameter `<npatches_fixed>` in the `options.m` file. Here, we set `<npatches_fixed>=2`.

2.3.3. Fixed sub-epidemic onset

We can also consider n-sub-epidemic models with onset fixed at time 0. In this case, all sub-epidemics start at time 0, and the threshold parameter C_{thr} drops from the model. We use parameter `<onset_fixed>` in the `options.m` file to specify whether the onset timing of the sub-epidemics is fixed at time 0 (`<onset_fixed>=1`) or not (`<onset_fixed>=0`).

2.3.4. Top-ranked sub-epidemic models

To select the top-ranked sub-epidemic models, we calculate the AIC_c values of the set of best fit sub-epidemic models with different values of C_{thr}. The AIC_c is given by (Hurvich & Tsai, 1989; Sugiura, 1978):

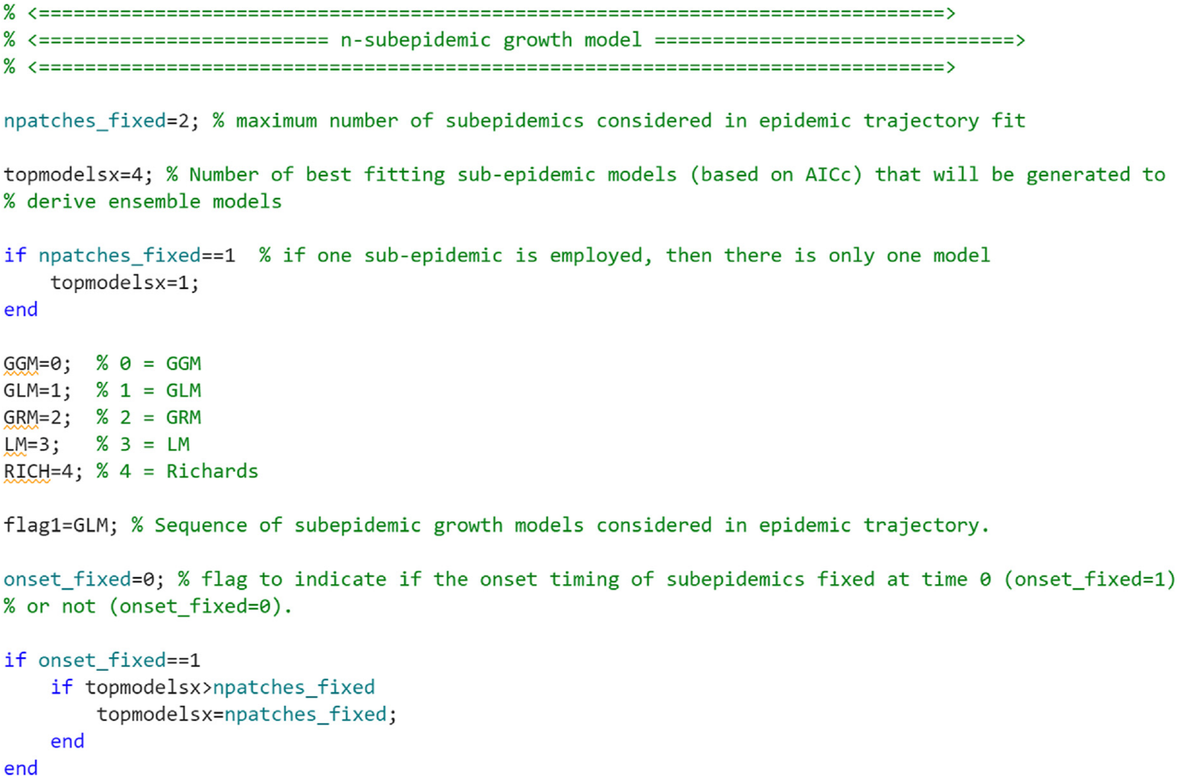
$$AIC_c = -2\ log(likelihood) + 2m + \frac{2m(m+1)}{n_d - m - 1}, \tag{5}$$

where m is the number of model parameters, including parameter C_{thr} if $n > 1$ and `<onset_fixed>=0`, and n_d is the number of data points. Specifically for normal distribution, the AIC_c is

$$AIC_c = n_d\ log(SSE) + 2m + \frac{2m(m+1)}{n_d - m - 1}, \tag{6}$$

where $SSE = \sum_{j=1}^{n_d}(f(t_j, \widehat{\Theta}) - y_{t_j})^2$. Parameter `<topmodelsx>` in the `options.m` file is used to specify the number of top-ranked models that will be generated and used to derive ensemble models. If the sub-epidemics' onset is fixed at time 0, then the maximum number of models that can be selected (e.g., top-ranked models) cannot exceed the maximum number of sub-epidemics considered in the epidemic's trajectory (`<npatches_fixed>`) .

To illustrate the methodology, we set `<onset_fixed>=0` and analyzed four top-ranking sub-epidemic models (`<topmodelsx>=4`). We used them to construct three ensemble sub-epidemic models, which we refer to as: Ensemble(2), Ensemble(3), and Ensemble(4), representing the ensembles of 2, 3, and 4 sub-epidemics, respectively. In the `options.m` file, the values of the parameters related to the n-sub-epidemic model and the number of top-ranked sub-epidemic models follow the code shown below:



```
% <============================================================================>
% <========================= n-subepidemic growth model ==============================>
% <============================================================================>

npatches_fixed=2; % maximum number of subepidemics considered in epidemic trajectory fit

topmodelsx=4; % Number of best fitting sub-epidemic models (based on AICc) that will be generated to
% derive ensemble models

if npatches_fixed==1  % if one sub-epidemic is employed, then there is only one model
    topmodelsx=1;
end

GGM=0;  % 0 = GGM
GLM=1;  % 1 = GLM
GRM=2;  % 2 = GRM
LM=3;   % 3 = LM
RICH=4; % 4 = Richards

flag1=GLM; % Sequence of subepidemic growth models considered in epidemic trajectory.

onset_fixed=0; % flag to indicate if the onset timing of subepidemics fixed at time 0 (onset_fixed=1)
% or not (onset_fixed=0).

if onset_fixed==1
    if topmodelsx>npatches_fixed
        topmodelsx=npatches_fixed;
    end
end
```

2.4. *Plotting simulations of the n-sub-epidemic model*

Before fitting the sub-epidemic model to the data, it is useful to check that the selected model yields simulations broadly consistent with the range of the time series data by generating model simulations with different parameter values. The function `plot_nsubepidemic.m` can be used to plot model solutions where the user provides the type of growth model to model each sub-epidemic by passing the parameter `<flag1>` (generalized-logistic growth model, Richards, Gompertz, etc.), the vectors containing model parameter values for each sub-epidemic, the number of sub-epidemics, n , the C_{thr} value, the initial condition for the first sub-epidemic, $C(0)$, and the simulation duration, as passing input parameters to the function in the following order: `flag1` , $r, p, a, K, n, C_{thr}, C(0)$, and finally the duration of the simulation. For example, the following call plots a simulation of the n-sub-epidemic model using the generalized logistic growth model (`<flag1>=1`) as the building block and the following model parameter values for three sub-epidemics: $r = [0.18\ 0.18\ 0.18], p = [0.98\ 0.98\ 0.98], K = [10000\ 5000\ 1000], n = 3, C_{thr} = 100$. The initial condition $C(0) = 1$, and the total duration of the simulation is set at 220.

```
>>plot_nsubepidemic(1,[0.18 0.18 0.18],[0.98 0.98 0.98],[],[10000 5000 1000],3,100,220)
```

Of note, in the above call, the value of parameter a is passed empty ([]) since the generalized logistic growth model does not use this parameter. This function will generate a figure (Fig. 2A) that shows the corresponding model solution $dC(t)/\ dt$. Additional representative simulations with other values of the C_{thr} are shown in Fig. 2.

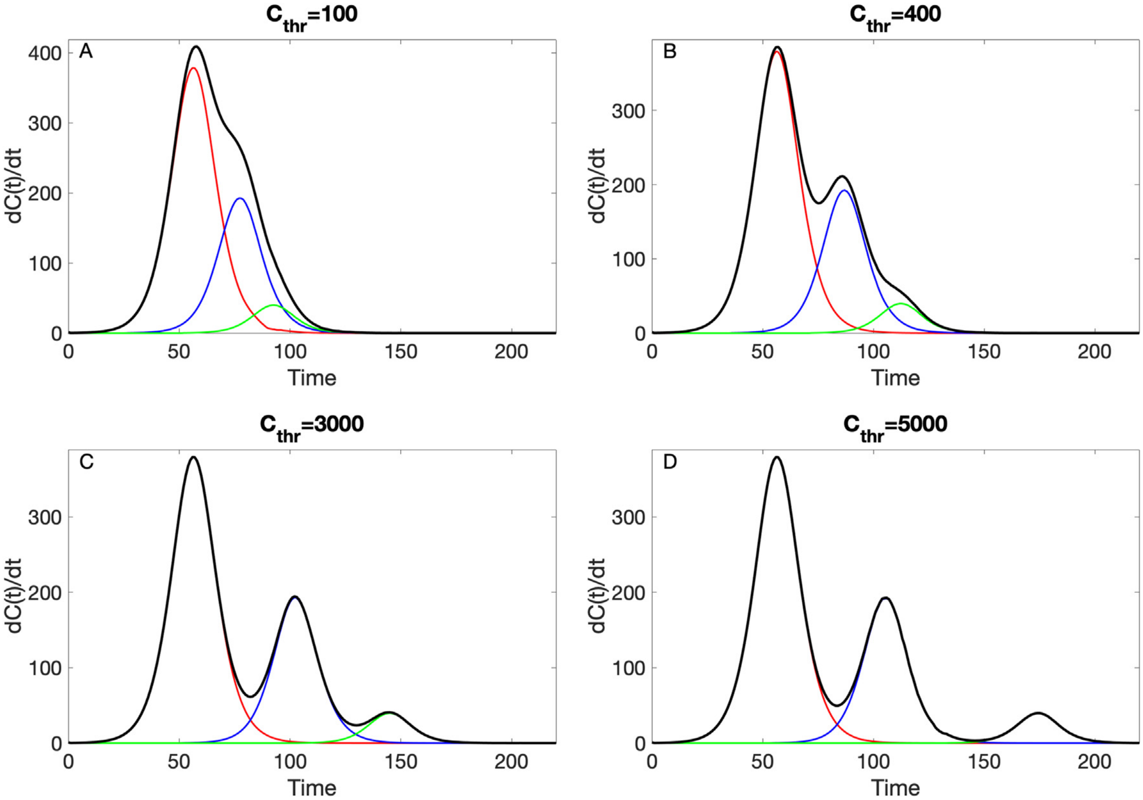


Fig. 2. Four representative profiles of the n-sub-epidemic model generated using the function `plot_nsubepidemic.m` where the sub-epidemic building block is modeled using the generalized logistic growth model and characterized by the following parameters: $r = [0.18\ 0.18\ 0.18]$, $p = [0.98\ 0.98\ 0.98]$, $K = [10000\ 5000\ 1000]$ for three sub-epidemics ($n = 3$), the initial condition $C(0) = 1$, the C_{thr} value is varied with values: A) 100, B) 400, C) 3000, D) 5000, and the total duration of the simulation is set at 220. The solid black line corresponds to the overall aggregated curve whereas the individual sub-epidemics are shown in different colors (red, blue, and green).

2.5. Parameter estimation method

Let $f(t,\Theta)$ denote the expected curve of the epidemic's trajectory. We can estimate the set of model parameters Θ by fitting the model solution to the observed data via nonlinear least squares (Banks et al., 2014) or maximum likelihood method (MLE) (Roosa et al., 2019). We can choose the nonlinear least squares method by setting the parameter `<method1>` to 0 in the `options.m` file. This is achieved by searching for the set of parameters $\widehat{\Theta}$ that minimizes the sum of squared differences between the observed data $\{y_{t_1}, y_{t_2}, \ldots, y_{t_{n_d}}\}$ and the model mean which corresponds to $f(t,\Theta)$. We will use smoothed data (as discussed in section 3.2) to estimate a total of $3n+1$ model parameters, namely $\Theta = (C_{thr}, r_1, p_1, K_{01}, \ldots, r_n, p_n, K_{0n})$. We estimate parameter C_{thr} through simple discretization of its range of plausible values. Our estimation procedure consists of two steps. First, for each C_{thr}, we search for the set of parameters $(r_1, p_1, K_{01}, \ldots, r_n, p_n, K_{0n})$ to minimize the sum of squared errors. Then we choose the C_{thr} and the corresponding estimates of other parameters leading to minimum SSE as the best fit.

Nonlinear least squares estimation method weights each of the data points equally and does not require a specific distributional assumption for y_t, except for the first moment $E[y_t] = f(t_i;\Theta)$. That is, the mean of the observed data at time t is equivalent to the expected count denoted by $f(t,\Theta)$ at time t (Myung, 2003). This method yields asymptotically unbiased point estimates regardless of any misspecification of the variance-covariance error structure. Hence, the estimated model mean $f(t_i,\widehat{\Theta})$ yields the best fit to observed data y_{t_i} in terms of squared L2 norm. In MATLAB, we can use the *fmincon* (TheMathWorks Inc, 2006) function to set the optimization problem. We also employ MATLAB's *MultiStart* (TheMathWorks Inc, 2010) feature to specify the number of random initial guesses of the model parameters using the parameter `<numstartpoints>` in the `options.m` file in order to search thoroughly for the best-fit parameter estimates.

We can also estimate parameters via MLE and assume different error structures in the data such as Poisson, negative binomial, and normal probability distributions.

a) Poisson

For a Poisson error structure, the full log-likelihood is given by:

$$\sum_{j=1}^{n}\left\{y_i\, ln(\mu_i) - ln\left(y_{t_j}!\right) - \mu_j\right\}, \tag{7}$$

where $\mu_j = f(t_j, \Theta)$ denotes the mean of y_{t_j}. The Poisson error structure can be specified by setting `<method1>=1` in the `options.m` file.

b) Negative binomial

Let $r > 0$ denote the number of failures until the experiment is stopped, $p \in [0, 1]$ denote the success probability in each experiment. The number of successes y before the r-th failure occurs has a negative binomial distribution:

$$f(r,p) = \left(\frac{r+y-1}{y}\right)p^y\,(1-p)^r = \frac{1}{y!}\prod_{j=0}^{y-1}(j+r)\,.p^y(1-p)^r \tag{8}$$

With mean $\mu = \frac{rp}{(1-p)}$ and variance $\sigma^2 = \frac{rp}{(1-p)^2} > \mu$. For n observations $y_1, \ldots, y_n$, the full log-likelihood is:

$$l(r,p) = \sum_{i=1}^{n}\left\{\left\{\sum_{j=0}^{y_i-1} ln(j+r)\right\} + y_i\, ln(p_i) + r ln(1-p_i) - ln(y_i!)\right\} \tag{9}$$

If we want to express the distribution with μ and σ^2, we can plug-in $p = 1 - \frac{\mu}{\sigma^2}$ and $r = \frac{\mu^2}{\sigma^2-\mu}$.

There are different types of variances commonly used in a negative binomial distribution. If the variance scales linearly with the mean,i.e., $\sigma^2 = \mu + \alpha\mu$, (`<method1>=2` in `options_fit.m`), then $p = \frac{\alpha}{1+\alpha}$ and $r = \mu/\alpha$. The full log-likelihood (9) can be expressed as follows:

$$l(\theta,\alpha) = \sum_{i=1}^{n}\left\{\left\{\sum_{j=0}^{y_i-1} ln\left(j+\alpha^{-1}f(t_i,\theta)\right)\right\} + y_i\, ln(\alpha) - \left(y_i + \alpha^{-1}f(t_i,\theta)\right)ln(1+\alpha) - ln(y_i!)\right\}. \tag{10}$$

If the variance scales quadratically with the mean, i.e., $\sigma^2 = \mu + \alpha\mu^2$ (`<method1>=4` in `options_fit.m` or `options_forecast.m`), then $p = \frac{\alpha\mu}{1+\alpha\mu}$ and $r = 1/\alpha$. The full log-likelihood (9) can be expressed as follows:

$$l(\theta,\alpha) = \sum_{i=1}^{n}\left\{\left\{\sum_{j=0}^{y_i-1} ln\left(j+\alpha^{-1}\right)\right\} + y_i\, ln(\alpha f(t_i,\theta)) - \left(y_i + \alpha^{-1}\right)ln(1+\alpha f(t_i,\theta)) - ln(y_i!)\right\}. \tag{11}$$

The more general form of variance is $\sigma^2 = \mu + \alpha\mu^d$ (`<method1>=5` in `options_fit.m`) with any $-\infty < d < \infty$. Then the full log-likelihood (9) can be expressed as follows:

$$l(\theta,\alpha) = \sum_{i=1}^{n}\left[\left\{\sum_{j=0}^{y_i-1} ln\left(j+\alpha^{-1}{\mu_i}^{2-d}\right)\right\} + y_i\, ln\left(\alpha{\mu_i}^{d-1}\right) - \left(y_i+\alpha^{-1}{\mu_i}^{2-d}\right)ln\left(1+\alpha{\mu_i}^{d-1}\right) - ln(y_i!)\right], \tag{12}$$

where $\mu_i = f\,(t_i, \theta)$. The number of parameters is 1 plus the number of parameters in the dynamical model based on ordinary differential equations (ODE) for (10)–(11), and 2 plus the number of parameters in the dynamical model for (12) if d is also estimated via MLE.

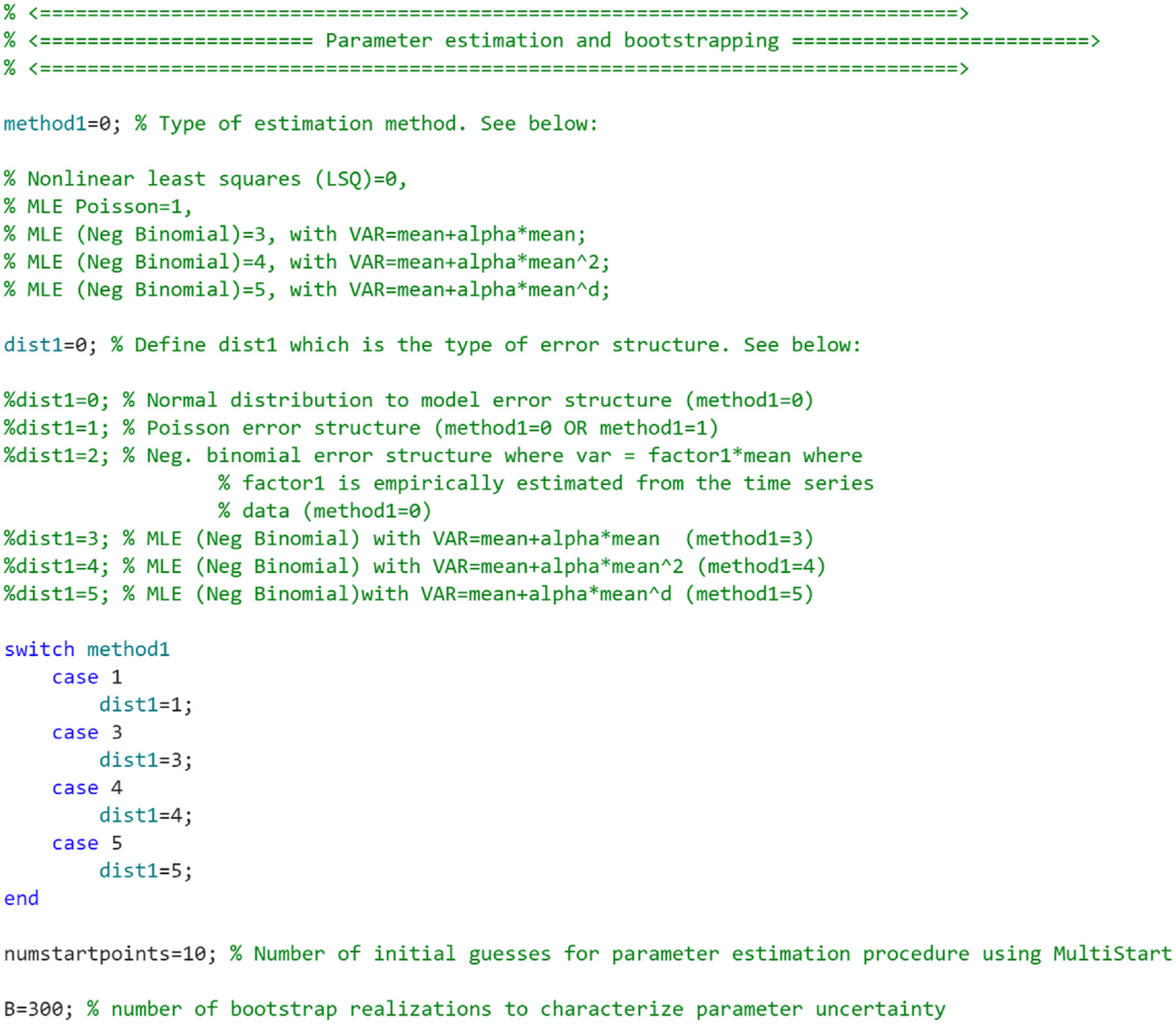
2.5.1. Parametric bootstrapping

To quantify parameter estimate uncertainty, we follow a parametric bootstrapping approach which allows the computation of standard errors and related statistics in the absence of closed-form formulas (Hastie et al., 2001). We generate B bootstrap samples from the best-fit model $f(t, \widehat{\Theta})$, with an assumed error structure specified using parameter `<dist1>` in the `options.m` file to quantify the uncertainty of the parameter estimates and construct confidence intervals. Typically, the error structure in the data is modeled using a probability model such as the Poisson or negative binomial distribution. Using nonlinear least squares (`<method1>=0`), besides a normally distributed error structure (`<dist1>=0`), we can also assume a

Poisson (`<dist1>=1`) and a negative binomial distribution (`<dist1>=2`) whereby the variance-to-mean ratio is empirically estimated from the time series. To estimate this constant ratio, we group a fixed number of observations (e.g., 7 observations for daily data into a bin across time), calculate the mean and variance for each bin, and then estimate a constant variance-to-mean ratio by calculating the average of the variance-to mean ratios over these bins. Using maximum likelihood estimation, we can estimate parameter uncertainty for Poisson and negative binomial error structures in the data (`<method1>=1` & `<dist1>=1` for Poisson and `<method1>=3` & `<dist1>=3`, `<method1>=4` & `<dist1>=4`, and `<method1>=5` & `<dist1>=5` for the different negative binomial error structures described above).

Using the best-fit model $f(t,\widehat{\Theta})$, we generate B-times replicated simulated datasets of size n_d, where the observation at time t_j is sampled from the corresponding distribution specified by `<dist1>`. Next, we refit the model to each of the B simulated datasets to re-estimate the parameters. The new parameter estimates for each realization are denoted by $\widehat{\Theta}_b$, where $b = 1, 2, \ldots, B$. Using the sets of re-estimated parameters $(\widehat{\Theta}_b)$, it is possible to characterize the empirical distribution of each estimate, calculate the variance, and construct confidence intervals for each parameter. The resulting uncertainty around the model fit can similarly be obtained from $(t,\widehat{\Theta}_1)$, $f(t,\widehat{\Theta}_2),\ldots,f(t,\widehat{\Theta}_B)$. We characterize the uncertainty using 300 bootstrap realizations (i.e., parameter `B=300` in the `options.m` file).

For the COVID-19 death data employed for illustration purposes, we fit the models through nonlinear least squares fitting and a normal error structure (i.e., `<method1>=0` and `<dist1>=0`). In the `options.m` file, the values of the parameters related to the parameter estimation method and parametric bootstrapping follow:



```
% <==============================================================================>
% <======================= Parameter estimation and bootstrapping =========================>
% <==============================================================================>

method1=0; % Type of estimation method. See below:

% Nonlinear least squares (LSQ)=0,
% MLE Poisson=1,
% MLE (Neg Binomial)=3, with VAR=mean+alpha*mean;
% MLE (Neg Binomial)=4, with VAR=mean+alpha*mean^2;
% MLE (Neg Binomial)=5, with VAR=mean+alpha*mean^d;

dist1=0; % Define dist1 which is the type of error structure. See below:

%dist1=0; % Normal distribution to model error structure (method1=0)
%dist1=1; % Poisson error structure (method1=0 OR method1=1)
%dist1=2; % Neg. binomial error structure where var = factor1*mean where
                  % factor1 is empirically estimated from the time series
                  % data (method1=0)
%dist1=3; % MLE (Neg Binomial) with VAR=mean+alpha*mean  (method1=3)
%dist1=4; % MLE (Neg Binomial) with VAR=mean+alpha*mean^2 (method1=4)
%dist1=5; % MLE (Neg Binomial)with VAR=mean+alpha*mean^d (method1=5)

switch method1
    case 1
        dist1=1;
    case 3
        dist1=3;
    case 4
        dist1=4;
    case 5
        dist1=5;
end

numstartpoints=10; % Number of initial guesses for parameter estimation procedure using MultiStart

B=300; % number of bootstrap realizations to characterize parameter uncertainty
```

2.6. Model-based forecasts with quantified uncertainty

We use $f(t+h,\widehat{\Theta})$ as the h days ahead forecast. The uncertainty of the forecasted value can be obtained using the previously described parametric bootstrap method. Let

$$f(t+h,\widehat{\Theta}_1),f(t+h,\widehat{\Theta}_2),\ldots,f(t+h,\widehat{\Theta}_B) \tag{13}$$

denote the forecasted value of the current state of the system propagated by a horizon of h time units, where $\widehat{\Theta}_b$ denotes the estimation of parameter set Θ from the b_{th} bootstrap sample. We can use these values to calculate the bootstrap variance as the measure of the uncertainty of the forecasts and use the 2.5% and 97.5% percentiles to construct the 95% prediction intervals (PI). We can set the forecasting horizon using the parameter `<forecastingperiod1>` in the `options_forecast.m` file.

2.7. Performance metrics

To assess the performance of the models during the calibration or forecasting periods, we used four performance metrics: the mean absolute error (MAE), the mean squared error (MSE), the coverage of the 95% prediction intervals (PI), and the weighted interval score (WIS) (Gneiting & Raftery, 2007). In the `options_forecast.m` file, the parameter `<getperformance>` is a Boolean variable (0/1) to indicate whether the user wishes to compute the performance metrics of the forecasts.

The *mean absolute error* (MAE) is given by:

$$MAE=\frac{1}{N}\sum_{h=1}^{N}\left|f(t_h,\widehat{\Theta})-y_{t_h}\right|, \tag{14}$$

where t_h are the time points of the calibration or forecasting period (Kuhn & Johnson, 2013), and N is the number of time points of the period. Similarly, the *mean squared error* (MSE) is given by:

$$MSE=\frac{1}{N}\sum_{h=1}^{N}\left(f(t_h,\widehat{\Theta})-y_{t_h}\right)^2 . \tag{15}$$

The coverage of the 95% PI corresponds to the fraction of data points that fall within the 95% PI, and is calculated as:

$$95\%\ PI\ Coverage=\frac{1}{N}\sum_{h=1}^{N}1\{Y_{t_h}>L_{t_h}\cap Y_{t_h}<U_{t_h}\}, \tag{16}$$

where L_{t_h} and U_{t_h} are the lower and upper bounds of the 95% PIs, respectively, Y_t are the data and **1** is an indicator variable that equals 1 if Y_t is in the specified interval and 0 otherwise.

The *weighted interval score* (WIS) (Gneiting & Raftery, 2007; *M4Competition:Competitor's Guide: Prizes and Rules*, 2018), which is a proper score recently embraced for quantifying model forecasting performance in epidemic forecasting studies (Bracher et al., 2021; Hwang, 2022; Roosa et al., 2020; Tariq et al., 2022), provides quantiles of predictive forecast distribution by combining a set of Interval Score (IS) for probabilistic forecasts. An IS is a simple proper score that requires only a central $(1-\alpha)\times 100\%$ PI (Gneiting & Raftery, 2007) and is described as

$$IS_\alpha(F,y)=(u-l)+\frac{2}{\alpha}\times(l-y)\times 1(y<l)+\frac{2}{\alpha}\times(y-u)\times 1(y>u)\ . \tag{17}$$

In this equation, **1** refers to the indicator function, meaning that $1(y<l)=1$ if $y<l$ and 0 otherwise. The terms l and u represent the $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$ quantiles of the forecast F. The IS consists of three distinct quantities:

1. The sharpness of F, given by the width $u-l$ of the central $(1-\alpha)\times 100\%$ PI.
2. A penalty term $\frac{2}{\alpha}\times(l-y)\times 1(y<l)$ for the observations that fall below the lower end point l of the $(1-\alpha)\times 100\%$ PI. This penalty term is directly proportional to the distance between y and the lower end l of the PI. The strength of the penalty depends on the level α.
3. An analogous penalty term $\frac{2}{\alpha}\times(y-u)\times 1(y>u)$ for the observations falling above the upper limit u of the PI.

To provide more detailed and accurate information on the entire predictive distribution, we report several central PIs at different levels $(1-\alpha_1)<(1-\alpha_2)<\ldots<(1-\alpha_K)$ along with the predictive median, $\tilde{y}$, which can be seen as a central prediction interval at level $1-\alpha_0\rightarrow 0$. This is referred to as the WIS, and it can be evaluated as follows:

$$WIS_{\alpha_{0:K}}(F,y) = \frac{1}{K+\frac{1}{2}} \cdot \left(w_0 . |y-m| + \sum_{k=1}^{K} w_k . IS_{\alpha_k}(F,y) \right), \tag{18}$$

where $w_k = \frac{\alpha_k}{2}$ for $k = 1, 2, \ldots .K$ and $w_0 = \frac{1}{2}$. Hence, WIS can be interpreted as a measure of how close the entire distribution is to the observation in units on the scale of the observed data (Bracher et al., 2021; Cramer et al., 2022).

2.8. Constructing ensemble forecasts from top-ranking models

Ensemble models that combine the strength of multiple models may exhibit significantly enhanced predictive performance (e.g. (Bleichrodt et al., 2023; Chowell et al., 2020; Chowell et al., 2022; Chowell & Luo, 2021; Ray & Reich, 2018; Viboud et al., 2018)). An ensemble model derived from the top-ranking K models are denoted by Ensemble(K) and illustrated in Fig. 3. Thus, Ensemble(2) and Ensemble(3) refer to the ensemble models generated from the combination of the top-ranking 2 and 3 models, respectively. The ensemble models can be derived from the unweighted (equal weights across contributing individual models) or a weighted combination of the highest-ranking sub-epidemic models based on the quality of fit as deemed by the AIC_{c_i} for the i-th model where $AIC_{c_1} \leq \ldots \leq AIC_{c_I}$ and $i = 1, \ldots, I$. In this case, we compute the weight w_i for the i-th model, $i = 1, \ldots, I$, where $\sum w_i = 1$ as follows:

$$w_i = \frac{\frac{1}{AIC_{c_i}}}{\frac{1}{AIC_{c_1}} + \frac{1}{AIC_{c_2}} + \ldots + \frac{1}{AIC_{c_I}}} \; for\ all\ i = 1, 2, \ldots, I, \tag{19}$$

and hence $w_I \leq \ldots \leq w_1$.

The estimated mean curve of the daily COVID-19 deaths for the Ensemble(I) model is:

$$f_{ens(I)}(t) = \sum_{i=1}^{I} w_i f_i\left(t, \widehat{\Theta}^{(i)}\right), \tag{20}$$

where given the training data, $\widehat{\Theta}^{(i)}$, denotes the set of estimated parameters, and $f_i(t, \widehat{\Theta}^{(i)})$ denotes the estimated mean curve of daily COVID-19 deaths, for the i-th model. To quantify the uncertainty of forecasts, we use wild bootstrap and for each bootstrap sample, we get the forecast for each model and compute the weighted average accordingly. Then we construct the 95% CI or PI using the 2.5% and 97.5% quantiles (Chowell & Luo, 2021). Alternatively, we can set the ensemble weights based

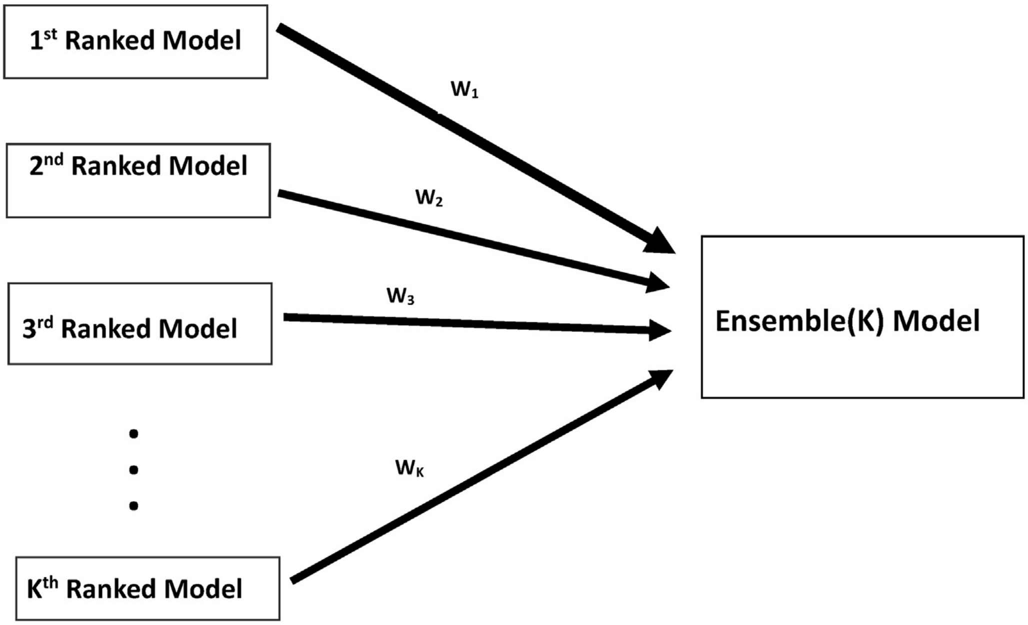


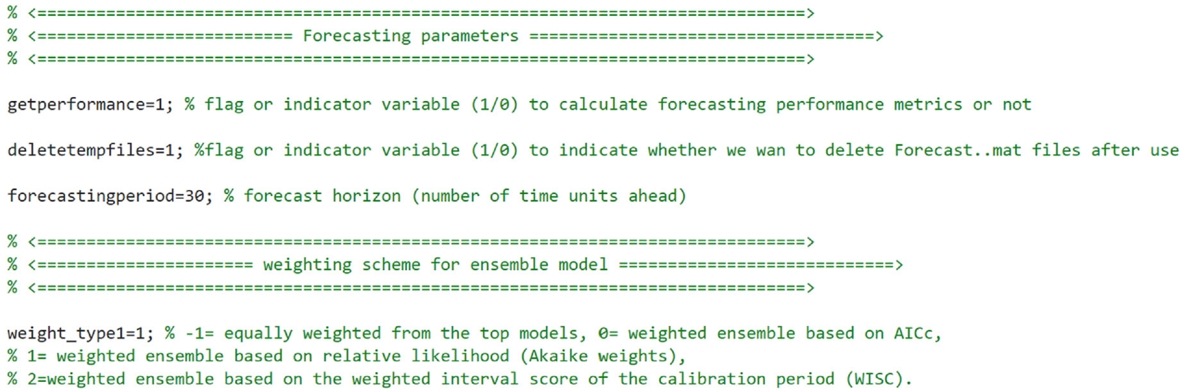
Fig. 3. Schematic diagram of the construction of the ensemble model from the weighted combination of the highest-ranking sub-epidemic models as deemed by the AIC_{c_i} for the i-th model where $AIC_{c_1} \leq \ldots \leq AIC_{c_I}$ and $i = 1, \ldots, I$. An ensemble derived from the top-ranking I models is denoted by Ensemble(I).

on different calibration performance metrics for the top-ranked models. For instance, we can make the ensemble weights proportional to the relative likelihood (l) rather than the reciprocal of the AIC_c. Let AIC_{min} denote the minimum AIC from the set of models. The relative likelihood or Akaike weight of model i is given by $l_i = e^{((AIC_{min} - AIC_i)/2)}$ (Burnham & Anderson, 2004). We compute the weight w_i for the i-th model where $\sum w_i = 1$ as follows:

$$w_i = \frac{l_i}{l_1 + l_2 + \ldots + l_I} \; for\ all\ i = 1, 2, \ldots, I, \tag{21}$$

and hence $w_I \leq \ldots \leq w_1$.

In the `options_forecast.m` file, we can specify three types of ensemble weights using the parameter `<weight_type1>`. Specifically, unweighted (`<weigth_type1>=-1`), weighted according to the AIC_c (`<weigth_type1>=0`), weighted based on the relative likelihood (`<weigth_type1>=1`), and weighted based on the WIS metric of the calibration period (`<weigth_type1>=2`). Along with choosing the type of ensemble weight, we can also specify parameters related to the forecasting horizon (`<forecastingperiod>`), and forecasting performance metrics (`<getperformance>`) in the `options_forecast.m` file as indicated below.



```
% <==============================================================================>
% <========================== Forecasting parameters ===================================>
% <==============================================================================>

getperformance=1; % flag or indicator variable (1/0) to calculate forecasting performance metrics or not

deletetempfiles=1; %flag or indicator variable (1/0) to indicate whether we wan to delete Forecast..mat files after use

forecastingperiod=30; % forecast horizon (number of time units ahead)

% <==============================================================================>
% <====================== weighting scheme for ensemble model ============================>
% <==============================================================================>

weight_type1=1; % -1= equally weighted from the top models, 0= weighted ensemble based on AICc,
% 1= weighted ensemble based on relative likelihood (Akaike weights),
% 2=weighted ensemble based on the weighted interval score of the calibration period (WISC).
```

2.9. Doubling times

Doubling times characterize the sequence of times at which the cumulative incidence doubles. Denote the times at which cumulative incidence doubles by t_{d_j}, such that $2C(t_{d_j}) = C(t_{d_{j+1}})$ where $t_{d_0} = 0, C(t_{d_0}) = C_0, j = 1, 2, 3, \ldots, n_g$ and n_g is the total number of times cumulative incidence doubles (Muniz-Rodriguez et al., 2020; Smirnova et al., 2021). The actual sequence of “doubling times” is defined as follows:

$$d_j = \Delta t_{d_j} = t_{d_j} - t_{d_{j-1}} \; where\ j = 1, 2, 3, \ldots, n_g. \tag{22}$$

For exponential growth, doubling times remain invariant and are given by $(\ln 2)/r$, whereas the doubling times increase when the growth pattern follows sub-exponential growth (Smirnova et al., 2021). We can characterize the doubling times and their uncertainty from the best-fit model $f(t, \widehat{\Theta})$ (Wallinga & Lipsitch, 2007). We can evaluate the uncertainty of the sequence of doubling times and the overall doubling time using the model parameter estimates derived from bootstrapping $(\widehat{\Theta}_b)$, where $b = 1,2,3,\ldots,B$. That is, $d_j(\widehat{\Theta}_b)$ provides a sequence of doubling times for a set of bootstrap parameter estimates, $\widehat{\Theta}_b$, where $b = 1, 2, 3, \ldots, B$. We can use these curves to derive 95% CIs for the sequence of doubling times and quantify the probability of observing a given number of doublings.

3. Toolbox tutorial

3.1. The dataset

For the purposes of this toolbox, the time series data will be stored in the input folder and needs to be a text file with the extension *.txt. The data file can contain one or more incidence time series (one per column in the file). Each column corresponds to the incidence curve over time for each epidemic corresponding to a different area/group. For instance, each column could correspond to different states in the U.S. or different countries in the world. In the `options.m` file, a specific data column in the file can be accessed for inference using the parameter `<outbreakx>`. If the time series file contains cumulative incidence count data, the name of the file containing the time series data starts with "cumulative" according to the following format:

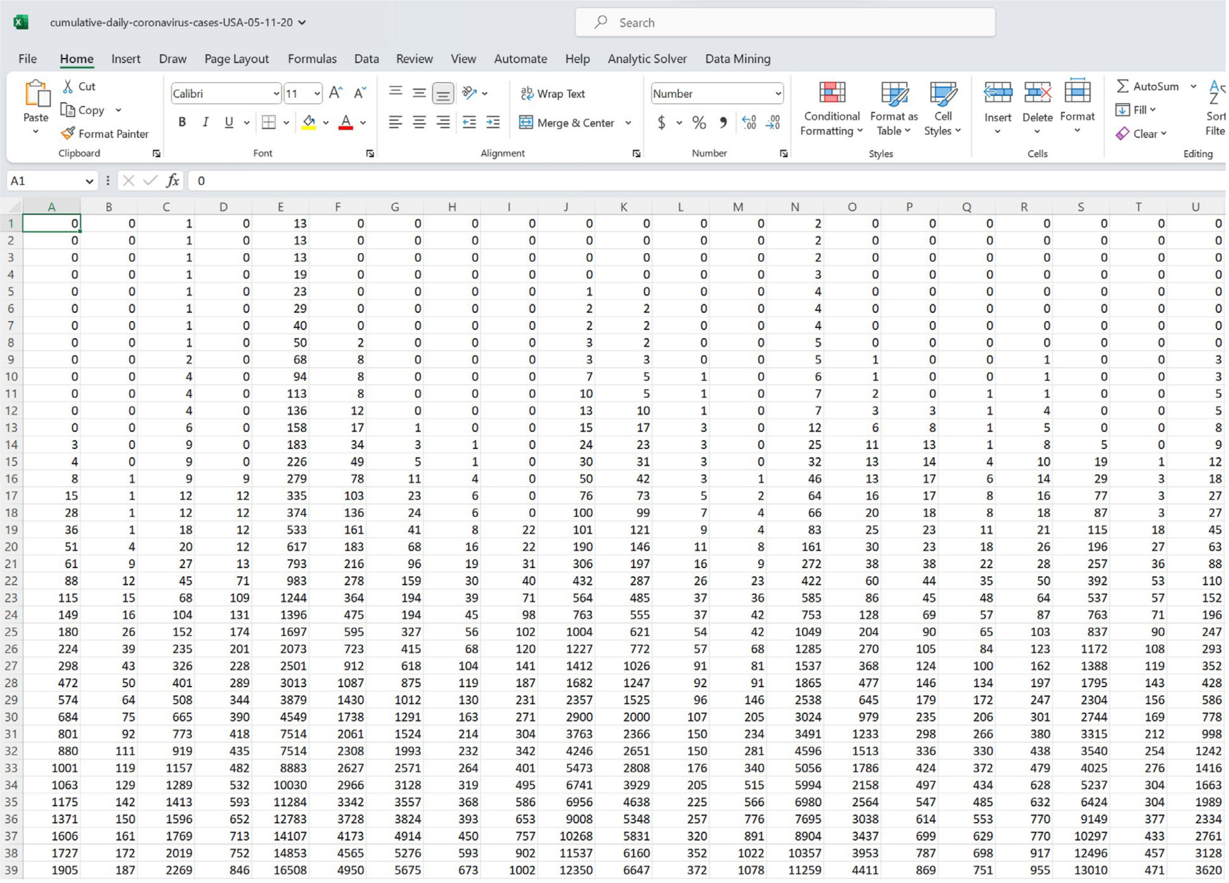
```
cumulative-<cadtemporal>-<caddisease>-<datatype>-<cadregion>-<caddate1>.txt
```

where `<cadtemporal>` is a string parameter that indicates the temporal resolution of the data (e.g., daily, weekly, yearly). Parameter `<caddisease>` is a string used to indicate the name of the disease related to the time series data. `<datatype>` is a string parameter indicating the nature of the data (e.g., cases, deaths, hospitalizations, etc.) whereas `<cadregion>` is a string parameter indicating the geographic region of the time series contained in the file (New York, USA, World, Asia, Africa, etc.). Finally, `<caddate1>` is a string to indicate the date for the most recent observation in the data file in the format: `mm-dd-yyyy`.

To illustrate the methodology presented in this tutorial paper, we used daily COVID-19 deaths reported in the USA from the publicly available data tracking system of the Johns Hopkins Center for Systems Science and Engineering (CSSE) (Dong et al., 2020). The data is also publicly available in the GitHub repository (Chowell et al., 2022). An example of a data file that we will use in this tutorial is: `cumulative-daily-coronavirus-deaths-USA-05-11-2020.txt`, which is in the input folder. This file contains daily cumulative COVID-19 deaths from 02 to 27–2020 to 05-11-2020 reported across 50 US states and Washington D.C.

(`<outbreakx>=1` through `51`) and at the national level (`<outbreakx>=52`). A partial view in Excel of the contents of the data file is shown in Fig. 4.

If the time series file contains incidence data, the name of the data file does not start with the word cumulative and follows the format:



A1 | 0

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	0	0	1	0	13	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
2	0	0	1	0	13	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
3	0	0	1	0	13	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
4	0	0	1	0	19	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
5	0	0	1	0	23	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0
6	0	0	1	0	29	0	0	0	0	2	2	0	0	4	0	0	0	0	0	0	0
7	0	0	1	0	40	0	0	0	0	2	2	0	0	4	0	0	0	0	0	0	0
8	0	0	1	0	50	2	0	0	0	3	2	0	0	5	0	0	0	0	0	0	0
9	0	0	2	0	68	8	0	0	0	3	3	0	0	5	1	0	0	1	0	0	3
10	0	0	4	0	94	8	0	0	0	7	5	1	0	6	1	0	0	1	0	0	3
11	0	0	4	0	113	8	0	0	0	10	5	1	0	7	2	0	1	1	0	0	5
12	0	0	4	0	136	12	0	0	0	13	10	1	0	7	3	3	1	4	0	0	5
13	0	0	6	0	158	17	1	0	0	15	17	3	0	12	6	8	1	5	0	0	8
14	3	0	9	0	183	34	3	1	0	24	23	3	0	25	11	13	1	8	5	0	9
15	4	0	9	0	226	49	5	1	0	30	31	3	0	32	13	14	4	10	19	1	12
16	8	1	9	9	279	78	11	4	0	50	42	3	1	46	13	17	6	14	29	3	18
17	15	1	12	12	335	103	23	6	0	76	73	5	2	64	16	17	8	16	77	3	27
18	28	1	12	12	374	136	24	6	0	100	99	7	4	66	20	18	8	18	87	3	27
19	36	1	18	12	533	161	41	8	22	101	121	9	4	83	25	23	11	21	115	18	45
20	51	4	20	12	617	183	68	16	22	190	146	11	8	161	30	23	18	26	196	27	63
21	61	9	27	13	793	216	96	19	31	306	197	16	9	272	38	38	22	28	257	36	88
22	88	12	45	71	983	278	159	30	40	432	287	26	23	422	60	44	35	50	392	53	110
23	115	15	68	109	1244	364	194	39	71	564	485	37	36	585	86	45	48	64	537	57	152
24	149	16	104	131	1396	475	194	45	98	763	555	37	42	753	128	69	57	87	763	71	196
25	180	26	152	174	1697	595	327	56	102	1004	621	54	42	1049	204	90	65	103	837	90	247
26	224	39	235	201	2073	723	415	68	120	1227	772	57	68	1285	270	105	84	123	1172	108	293
27	298	43	326	228	2501	912	618	104	141	1412	1026	91	81	1537	368	124	100	162	1388	119	352
28	472	50	401	289	3013	1087	875	119	187	1682	1247	92	91	1865	477	146	134	197	1795	143	428
29	574	64	508	344	3879	1430	1012	130	231	2357	1525	96	146	2538	645	179	172	247	2304	156	586
30	684	75	665	390	4549	1738	1291	163	271	2900	2000	107	205	3024	979	235	206	301	2744	169	778
31	801	92	773	418	7514	2061	1524	214	304	3763	2366	150	234	3491	1233	298	266	380	3315	212	998
32	880	111	919	435	7514	2308	1993	232	342	4246	2651	150	281	4596	1513	336	330	438	3540	254	1242
33	1001	119	1157	482	8883	2627	2571	264	401	5473	2808	176	340	5056	1786	424	372	479	4025	276	1416
34	1063	129	1289	532	10030	2966	3128	319	495	6741	3929	205	515	5994	2158	497	434	628	5237	304	1663
35	1175	142	1413	593	11284	3342	3557	368	586	6956	4638	225	566	6980	2564	547	485	632	6424	304	1989
36	1371	150	1596	652	12783	3728	3824	393	653	9008	5348	257	776	7695	3038	614	553	770	9149	377	2334
37	1606	161	1769	713	14107	4173	4914	450	757	10268	5831	320	891	8904	3437	699	629	770	10297	433	2761
38	1727	172	2019	752	14853	4565	5276	593	902	11537	6160	352	1022	10357	3953	787	698	917	12496	457	3128
39	1905	187	2269	846	16508	4950	5675	673	1002	12350	6647	372	1078	11259	4411	869	751	955	13010	471	3620

Fig. 4. A partial view in Excel of a data file that we will use in this tutorial (`cumulative-daily-coronavirus-deaths-USA-05-11-2020.txt`) which is in the toolbox's input folder. This file contains daily cumulative COVID-19 deaths from 02 to 27–2020 to 05-11-2020 reported across 50 US states and Washington D.C.

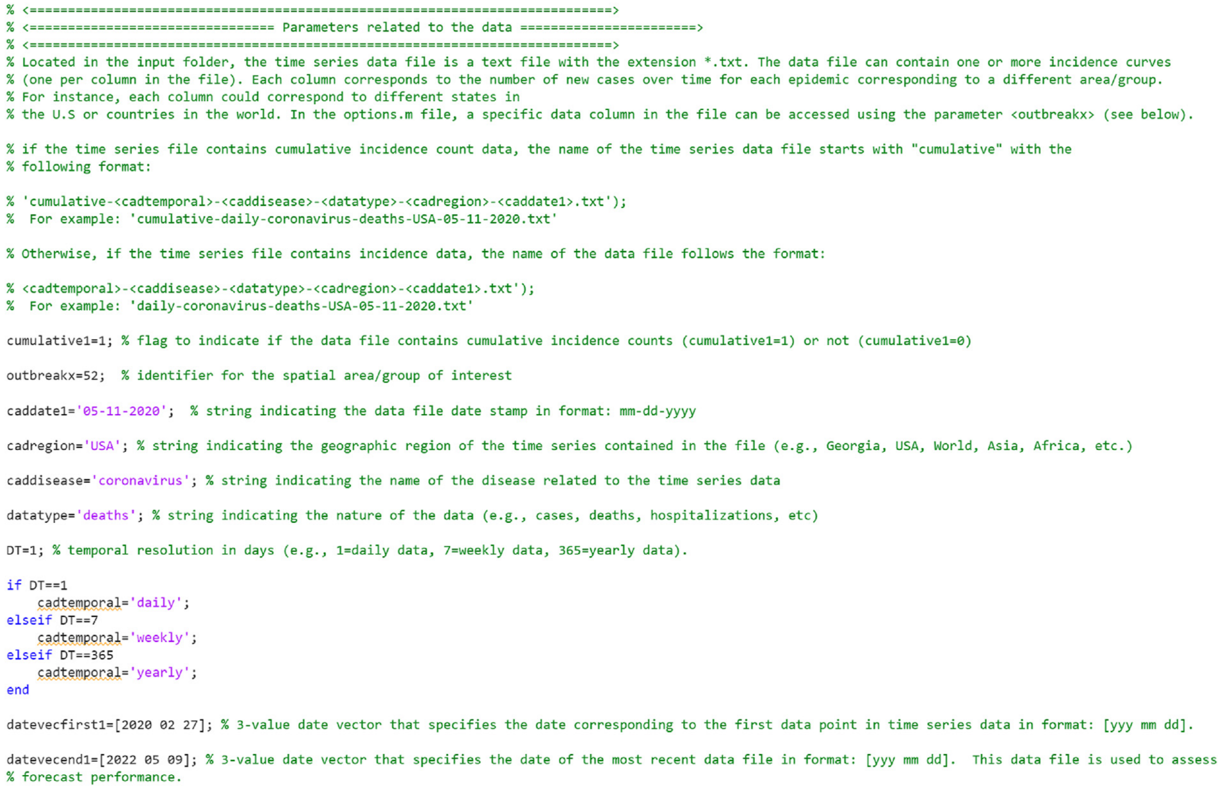
<cadtemporal>**-**<caddisease>**-**<datatype>**-**<cadregion>**-**<caddate1>**.txt**

For example: daily-coronavirus-deaths-USA-05-11-2020.txt

In the options.m file, the parameter <datevecfirst1> is a 3-value vector that specifies the date corresponding to the first data point in time series data in format: [yyyy mm dd]. If the date of the first observation corresponds to February 27th, 2020, then <datevecfirst1>= [2020 02 27]. Similarly, the parameter <datevecend1> is a 3-value vector that specifies the date of the most recent data file in format: [yyyy mm dd]. Thus, the name of the cumulative data file containing the most recent data corresponds to:

cumulative-<cadtemporal>-<caddisease>-<datatype>-<cadregion>-<**datevecend1**>.txt

which is in the input folder and contains the latest time series data to assess forecast performance. Finally, the parameter <DT> is an integer indicating the temporal resolution of the time series data (e.g., <DT>=1 for daily data; <DT>=7 for weekly data). In the options.m file, the values of the parameters related to the data follow:



```
% <============================================================================>
% <================================ Parameters related to the data ======================>
% <============================================================================>
% Located in the input folder, the time series data file is a text file with the extension *.txt. The data file can contain one or more incidence curves
% (one per column in the file). Each column corresponds to the number of new cases over time for each epidemic corresponding to a different area/group.
% For instance, each column could correspond to different states in
% the U.S or countries in the world. In the options.m file, a specific data column in the file can be accessed using the parameter <outbreakx> (see below).

% if the time series file contains cumulative incidence count data, the name of the time series data file starts with "cumulative" with the
% following format:

% 'cumulative-<cadtemporal>-<caddisease>-<datatype>-<cadregion>-<caddate1>.txt');
%  For example: 'cumulative-daily-coronavirus-deaths-USA-05-11-2020.txt'

% Otherwise, if the time series file contains incidence data, the name of the data file follows the format:

% <cadtemporal>-<caddisease>-<datatype>-<cadregion>-<caddate1>.txt');
%  For example: 'daily-coronavirus-deaths-USA-05-11-2020.txt'

cumulative1=1; % flag to indicate if the data file contains cumulative incidence counts (cumulative1=1) or not (cumulative1=0)

outbreakx=52;  % identifier for the spatial area/group of interest

caddate1='05-11-2020';  % string indicating the data file date stamp in format: mm-dd-yyyy

cadregion='USA'; % string indicating the geographic region of the time series contained in the file (e.g., Georgia, USA, World, Asia, Africa, etc.)

caddisease='coronavirus'; % string indicating the name of the disease related to the time series data

datatype='deaths'; % string indicating the nature of the data (e.g., cases, deaths, hospitalizations, etc)

DT=1; % temporal resolution in days (e.g., 1=daily data, 7=weekly data, 365=yearly data).

if DT==1
    cadtemporal='daily';
elseif DT==7
    cadtemporal='weekly';
elseif DT==365
    cadtemporal='yearly';
end

datevecfirst1=[2020 02 27]; % 3-value date vector that specifies the date corresponding to the first data point in time series data in format: [yyy mm dd].

datevecend1=[2022 05 09]; % 3-value date vector that specifies the date of the most recent data file in format: [yyy mm dd].  This data file is used to assess
% forecast performance.
```

3.2. Data adjustments

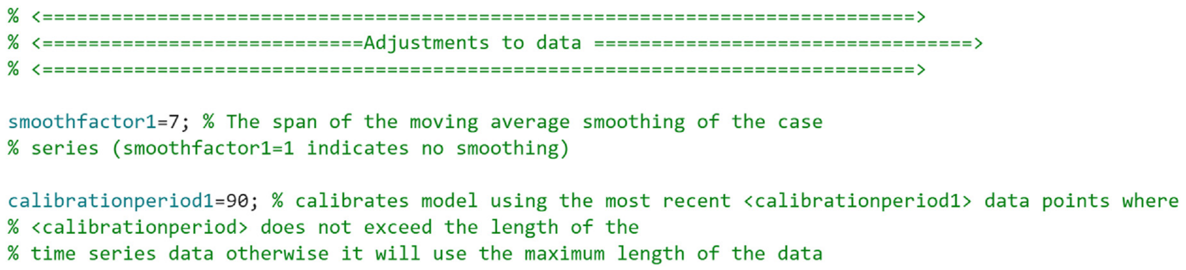
3.2.1. Data smoothing

To reduce the noise in the original data due to artificial reasons such as the weekend effects, we can smooth out the time series data using the moving average of the time series whereby <smoothfactor1> is a parameter in the options.m file that specifies the span of the moving average (e.g., <smoothfactor1>=1 implies no smoothing applied to the data). For the daily COVID-19 death data employed for illustration purposes, we set <smoothfactor1>=7 and smoothed out the daily series using a 7-day moving average to reduce the noise in the original data due to artificial reasons such as the weekend effects.

3.2.2. Calibration period

To fit the models to the most recent observations contained in a time series file, we can specify the length of the calibration period whereby <calibrationperiod1> in the options.m file indicates the number of recent data points that will be used to calibrate the models. If `<calibrationperiod1>` exceeds the length of the time series contained in the data file, the calibration period is set to the maximum length of the available data.

For illustration purposes, we used a 90-day calibration period (i.e., `<calibrationperiod1>=90`). In the `options.m` file, the values of the parameters related to smoothing and calibration period follow:



```
% <============================================================================>
% <============================Adjustments to data =================================>
% <============================================================================>

smoothfactor1=7; % The span of the moving average smoothing of the case
% series (smoothfactor1=1 indicates no smoothing)

calibrationperiod1=90; % calibrates model using the most recent <calibrationperiod1> data points where
% <calibrationperiod> does not exceed the length of the
% time series data otherwise it will use the maximum length of the data
```

3.3. *Fitting the n-sub-epidemic models to data with quantified uncertainty*

To fit the *n*-sub-epidemic models to the data with quantified uncertainty, we need to run the function `Run_Fit_subepidemicFramework.m`. This function uses the input parameters provided by the user in the `options.m` file. However, the function can also receive `<outbreakx>` and `<caddate1>` as passing input parameters while the remaining inputs are obtained from the `options.m` file.

For example, to fit the ensemble *n*-sub-epidemic models to the daily curve of COVID-19 deaths in the USA as of the week of '05-11-2020' (data file path: `input/cumulative-daily-coronavirus-deaths-USA-05-11-2020.txt`), we can run the function from MATLAB's command line window as follows:

```
>> Run_Fit_subepidemicFramework(52,'05-11-2020')
```

This function will generate several output MATLAB files in the output folder. For instance, the following output file contains the fits of the top-ranking models:

`ABC-ensem-npatchesfixed-2-onsetfixed-0-smoothing-7-daily-coronavirus-deaths-USA-state-52-05-11-2020-flag1-1-method-0-dist-0-calibrationperiod-90.mat`

Please note the names of the output files contain the values of the parameters for reference.

The following output internal files contain the uncertainty characteristics associated with each of the top-ranking models:

(a) `modifiedLogisticPatch-ensem-npatchesfixed-2-onsetfixed-0-smoothing-7-daily-coronavirus-deaths-USA-state-52-05-11-2020-flag1-1-method-0-dist-0-calibrationperiod-90-rank-1.mat`
(b) `modifiedLogisticPatch-ensem-npatchesfixed-2-onsetfixed-0-smoothing-7-daily-coronavirus-deaths-USA-state-52-05-11-2020-flag1-1-method-0-dist-0-calibrationperiod-90-rank-2.mat`
(c) `modifiedLogisticPatch-ensem-npatchesfixed-2-onsetfixed-0-smoothing-7-daily-coronavirus-deaths-USA-state-52-05-11-2020-flag1-1-method-0-dist-0-calibrationperiod-90-rank-3.mat`
(d) `modifiedLogisticPatch-ensem-npatchesfixed-2-onsetfixed-0-smoothing-7-daily-coronavirus-deaths-USA-state-52-05-11-2020-flag1-1-method-0-dist-0-calibrationperiod-90-rank-4.mat`

These output files are needed to plot model fits, derive parameter estimates, generate short-term forecasts, and quantify the calibration and forecasting performance metrics.

3.4. *Plot the mean model fits and quality of fit metrics for the top-ranked models*

After running the function `Run_Fit_subepidemicFramework.m` with the desired input parameters, we can use the function `plotRankings_subepidemicFramework.m` to plot the mean model fits of the top-ranking models including their sub-epidemic profiles and the associated quality of model fit metrics including the AIC_c, the relative likelihood, and the evidence ratio based on the inputs. However, this function can also receive `<outbreakx>` and `<caddate1>` as passing input parameters while the remaining inputs are obtained from the `options.m` file. Running this function from MATLAB's command line, we have:

```
>> plotRankings_subepidemicFramework(52,'05-11-2020')
```

Figs. 5 and 6 illustrate the outputs obtained from this function call. Fig. 5 shows the mean model fits of the top-ranked sub-epidemic models, which indicate that the top-ranked models consist of 2 sub-epidemics. The corresponding goodness of fit

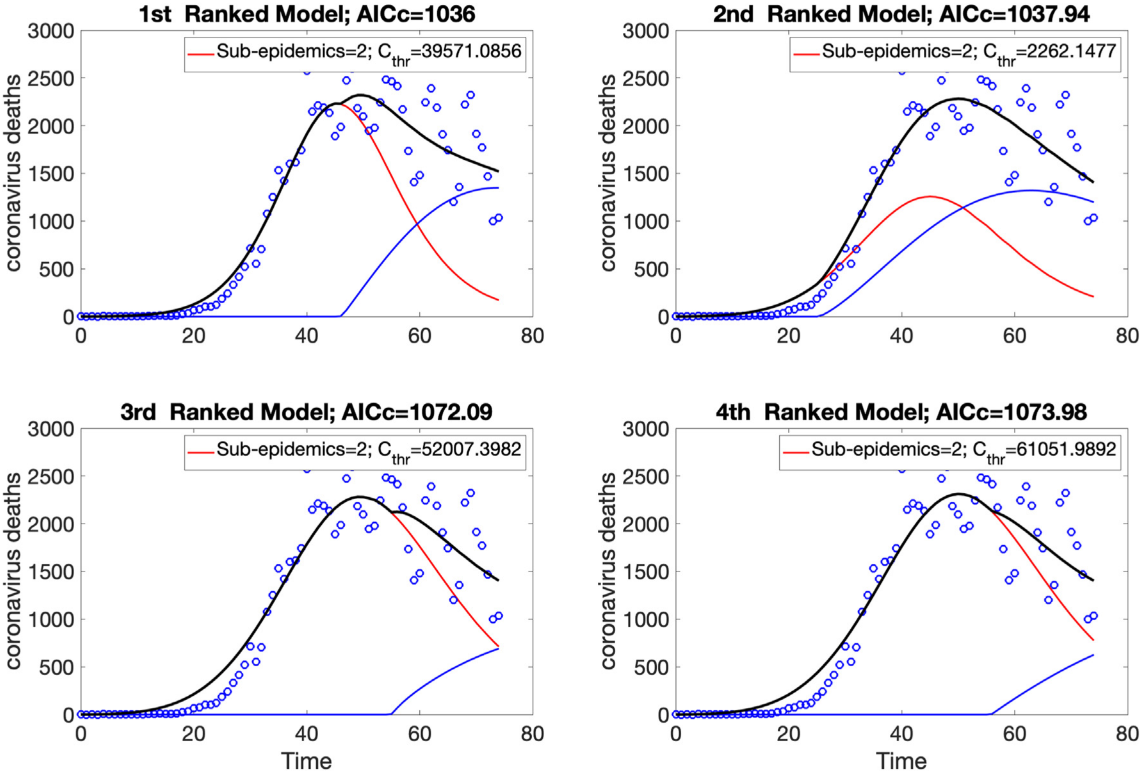


Fig. 5. Mean model fits of the top-ranked sub-epidemic models (<topmodelsx>=4 in options.m file) calibrated to the daily curve of COVID-19 deaths in the USA from 27-Feb-2020 to 11-May-2020. The blue and red solid lines correspond to the individual sub-epidemic curves. The solid black line represents the overall aggregated epidemic curve. The legend in each panel indicates the number of sub-epidemics involved in each model and the value of the C_{thr} parameter.

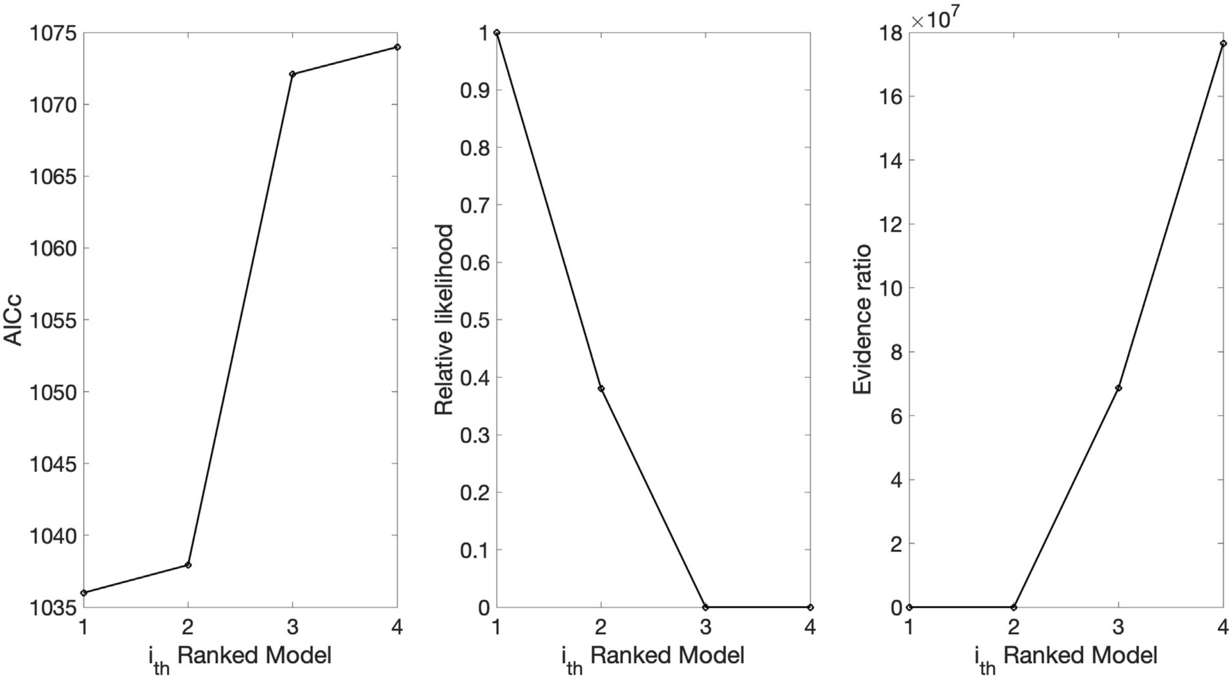


Fig. 6. Quality of model fit metrics for the top-ranked sub-epidemic models (<topmodelsx>=4 in options.m file) calibrated to the daily curve of COVID-19 deaths in the USA from 27-Feb-2020 to 11-May-2020.

statistics of the top-ranked models in terms of the AIC_c, the relative likelihood, and the evidence ratio are shown in Fig. 6. It also saves the AIC_c values of the top-ranked models in the following .csv file:

```
AICc-topRanked-onsetfixed-0-flag1-1-method-0-dist-0-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
```

For comparison, a simpler growth model consisting of a single sub-epidemic (<npatches_fixed>=1) performs substantially worse (AIC_c = 1107.13); Fig. 1 in Appendix A (Fig. A.1)).

3.5. Plot the model fits, parameter estimates, and performance metrics of the top-ranking models

Using the function `plotFit_subepidemicFramework.m`, we can plot the fits of the top-ranking models including their sub-epidemic profiles and the residual plots based on the inputs indicated in the `options.m` file. However, this function can also receive <outbreakx> and <caddate1> as passing input parameters while the remaining inputs are obtained from the `options.m` file.

In addition, this function also plots the empirical distributions of the parameters associated with each of the top-ranking models and the calibration performance metrics (MSE, MAE, 95% PI, WIS). Finally, this function also outputs a .csv file in the output folder with the calibration performance metrics associated with the top-ranking models and the estimated sequence of doubling times for each of the top-ranked models. Using the default parameter values indicated in the `options.m` file, the actual call to this function from MATLAB's command line follows:

```
>> plotFit_subepidemicFramework(52, '05-11-2020')
```

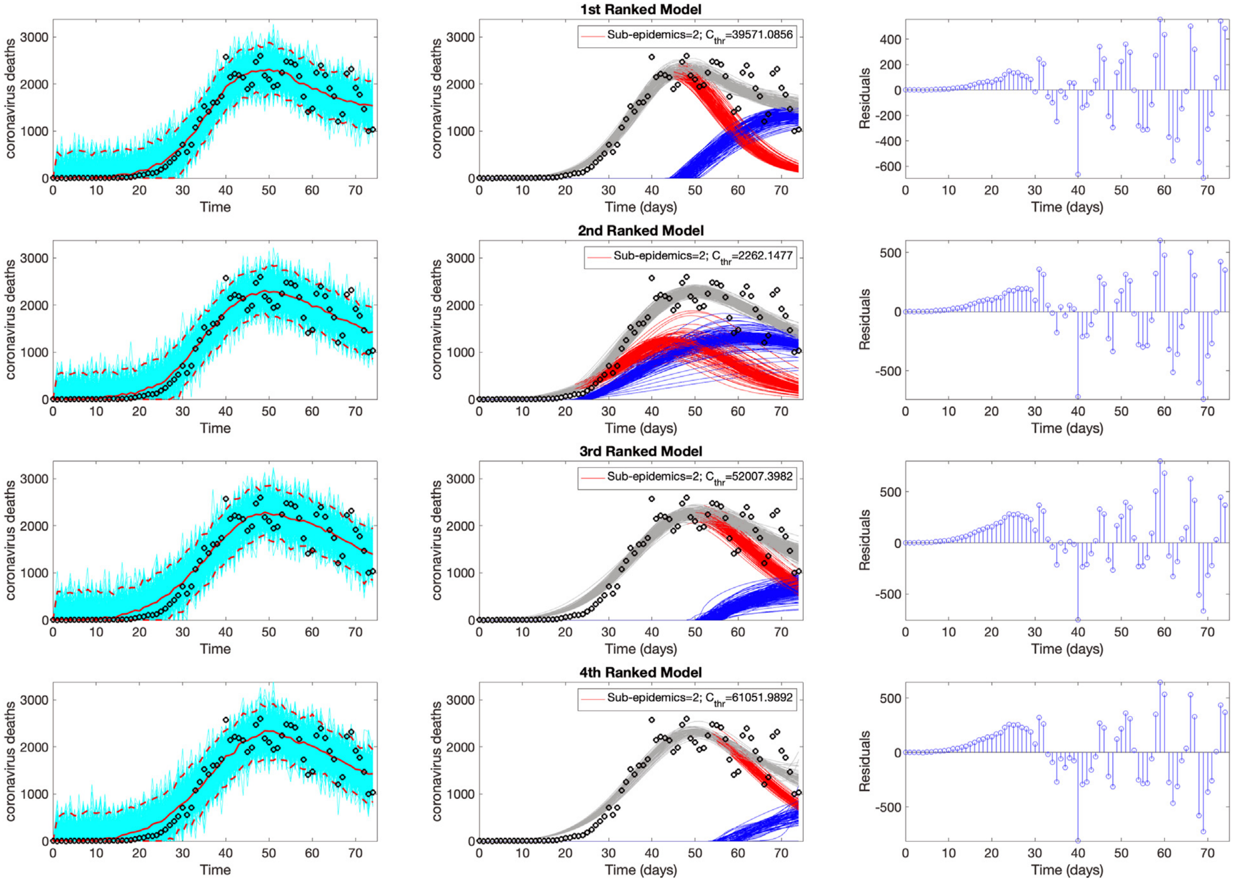


Fig. 7. Model fits of the top-ranking sub-epidemic models to the daily curve of COVID-19 deaths in the USA from 27-Feb-2020 to 11-May-2020. The sub-epidemic models capture the entire epidemic curve well, including the latter plateau dynamics, by considering models with two sub-epidemics. The best model fit (solid red line) and 95% PI (dashed red lines) are shown in the left panels. The cyan curves show simulated samples with the corresponding error structure imposed on each bootstrap model fit, which are used to derive the 95% PI. The sub-epidemic mean profiles obtained from the parametric bootstrapping with 300 bootstrap realizations are shown in the center panels, where the red and blue curves represent two sub-epidemics, and the gray curves are the aggregated epidemic trajectories. Black circles correspond to the data points. For each model fit, the residuals are also shown (right panels).

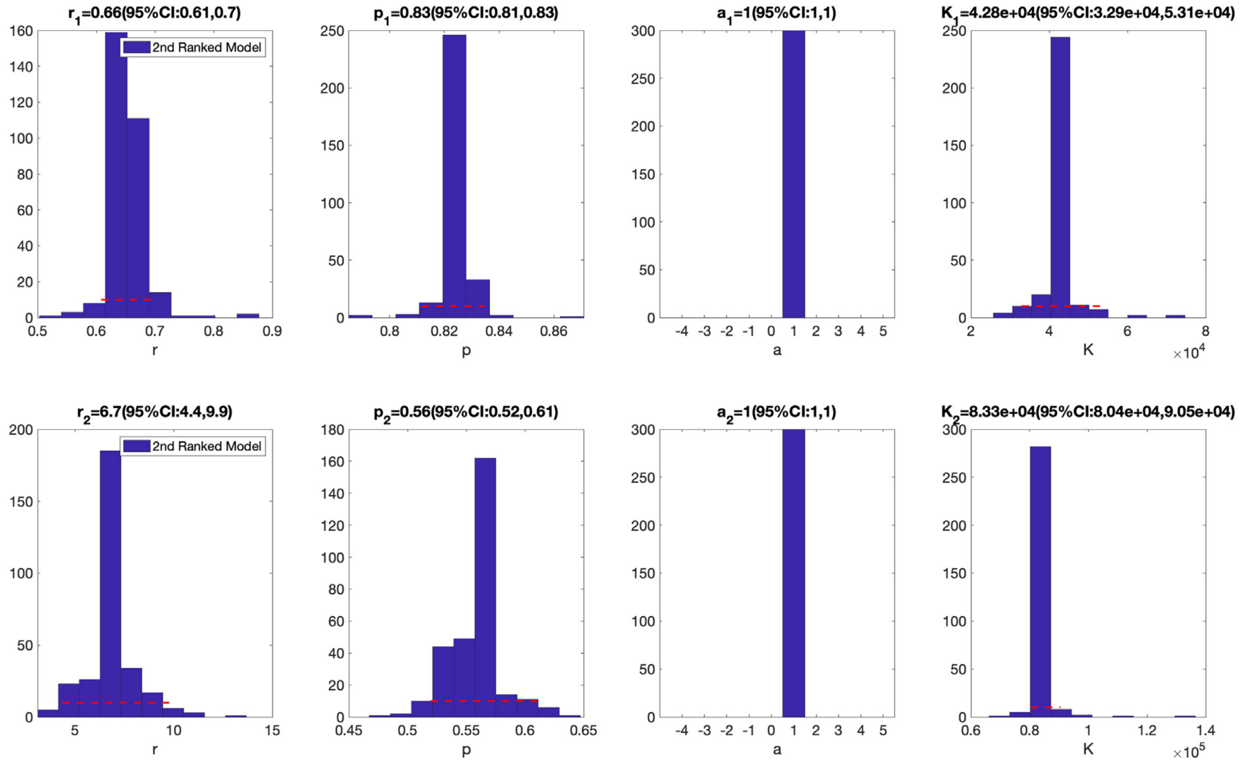


Fig. 8. Parameter estimates for the first (top panel) and the second sub-epidemics (bottom panels) are shown for the 1st-ranked sub-epidemic model after fitting the *n*-subepidemic modeling framework to the daily curve of COVID-19 deaths in the USA from 27-Feb-2020 to 11-May-2020.

Figs. 7-9 illustrate the outputs obtained from the above call to the function. The fits of the top-ranking sub-epidemic models, including the sub-epidemic profiles and residuals to the daily curve of COVID-19 deaths are shown in Fig. 7. This figure indicates that the top-ranked models yield a similarly good fit to the data. The outputs also include figures that display the empirical distribution of the parameter estimates related to the sub-epidemics involved in each fit of the top-ranked models. For example, Fig. 8 shows the corresponding estimates for the 1st ranked sub-epidemic model. These parameter estimates are well identified as the confidence intervals lie in a well-defined range of values (Cobelli & Romanin-Jacur, 1976; Raue et al., 2009). The calibration performance metrics capturing the quality of fit of the top-ranked sub-epidemic models are also displayed in Fig. 9. For instance, this figure indicates that WIS metrics ranges from ~117 to ~129 whereas the coverage of the 95% prediction intervals varied little between ~91% and 96% for the top-ranked models. This function will store the following .csv files in the output folder:

1) Calibration performance metrics:

```
   performance-calibration-topRanked-onsetfixed-0-flag1-1-method-0-dist-0-daily-coronavirus
-deaths-USA-area-52-05-11-2020.csv
```

2) Doubling times for the top-ranked models:

```
   doublingTimes-ranked(1)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-0-daily-coronavirus-
deaths-USA-area-52-05-11-2020.csv
   doublingTimes-ranked(2)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-0-daily-coronavirus-
deaths-USA-area-52-05-11-2020.csv
   doublingTimes-ranked(3)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-0-daily-coronavirus-
deaths-USA-area-52-05-11-2020.csv
   doublingTimes-ranked(4)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-0-daily-coronavirus-
deaths-USA-area-52-05-11-2020.csv
```

3.6. *Generate the top-ranked and ensemble sub-epidemic model forecasts and the associated forecasting performance metrics*

Using the function `plotForecast_subepidemicFramework.m`, we can plot the short-term forecasts from the top-ranking sub-epidemic models as well as the ensemble models derived from the top-ranking sub-epidemic models based on the inputs indicated in the `options.m` and the `options_forecast.m` files. However, this function can also receive

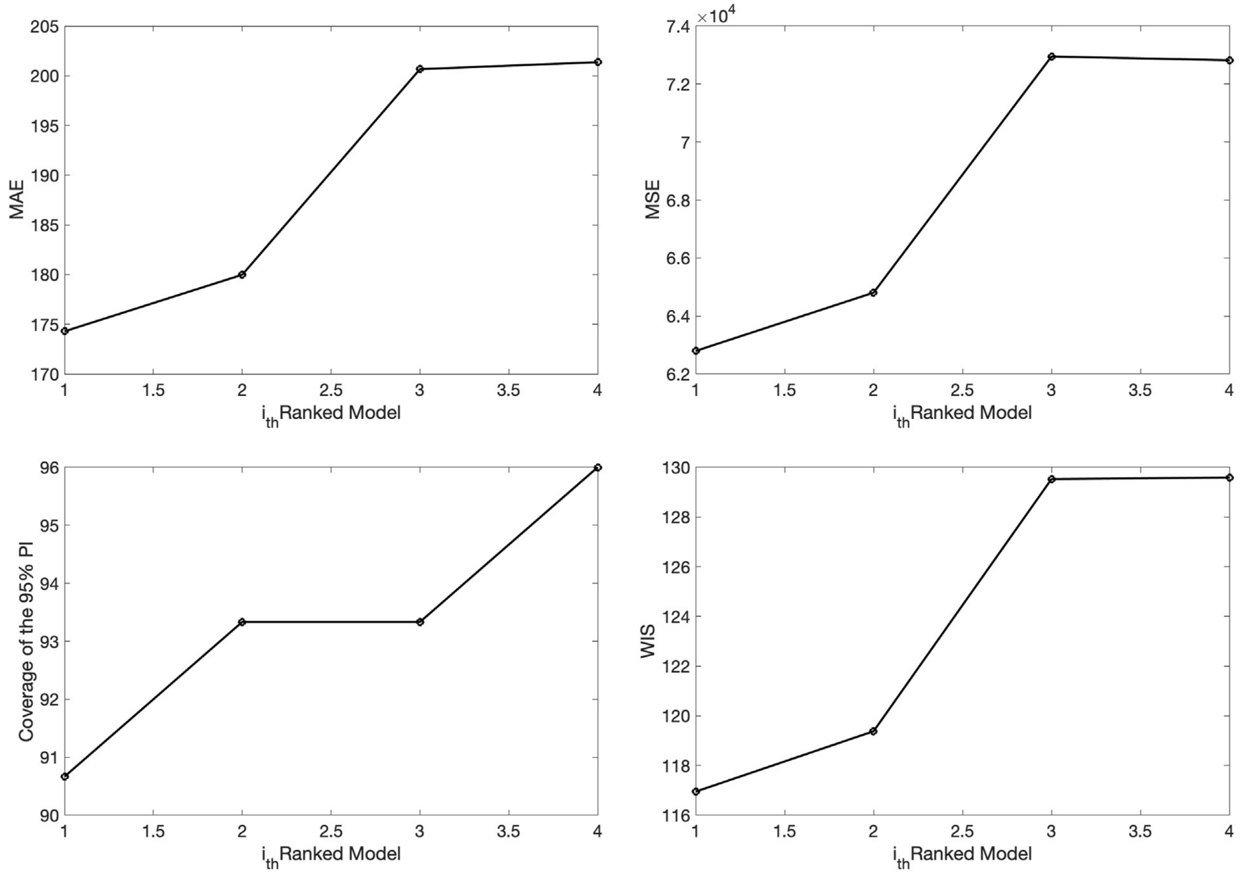


Fig. 9. Calibration performance metrics for the top-ranking sub-epidemic models fit to the daily curve of COVID-19 deaths in the USA from 27-Feb-2020 to 11-May-2020. These metrics are also saved in a .csv data file (`‘performance-calibration-topRanked-onsetfixed-0-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv’`). For instance, these WIS metrics during the calibration period ranged from ~117 to ~129 across the four top-ranked models.

parameters `<outbreakx>`, `<caddate1>`, `<forecastingperiod>`, or `<weight_type1>` as passing input parameters while the remaining inputs are read from the `options.m` and `options_forecast.m` files. Moreover, the data associated with each of the top-ranked model forecasts as well as the ensemble forecasts are saved as .csv files in the output folder.

In addition, this function also plots the forecasting performance metrics (MSE, MAE, 95% PI, WIS) for the top-ranking models as well as the ensemble sub-epidemic models. Finally, this function also stores .csv files in the output folder with the forecasting performance metrics associated with the top-ranking models and the ensemble models, and the estimated doubling times for each of the top-ranked models. Using the default parameter values indicated in the `options.m` and `options_forecast.m` files, the call to this function from MATLAB's command line follows:

```
>> plotForecast_subepidemicFramework(52, ‘05-11-2020’, 30, 1)
```

Figs. 10-14 illustrate the outputs obtained from this function call. Fig. 10 shows the 30-day forecasts derived from the top-ranking sub-epidemic models whereas Fig. 11 shows the sub-epidemic profiles of the forecasts. These forecasts indicate that the top-ranked models performed well. Moreover, the data associated with each of the top-ranked model forecasts are also saved as .csv files in the output folder.

The forecasting performance metrics for the top-ranked models are displayed in Fig. 12, and these metrics are also saved in a .csv file in the output folder. In comparison, the forecast derived from the simpler one sub-epidemic model (`<npatches_fixed>=1`) was substantially worse as shown in Fig. 2 in Appendix A (Fig. A.2).

The corresponding three ensemble forecasts (Ensemble(2), Ensemble(3), and Ensemble(4)) derived from the weighted combination of the top-ranked models based on their relative likelihood or Akaike weights (e.g., `<weigth_type1>=1` in the `options_forecast.m` file) are shown in Fig. 13. Also, the corresponding forecasting performance metrics for the ensemble models are shown in Fig. 14 and are saved in a .csv file in the output folder. The ensemble models performed similarly. The ensemble models' forecasts achieved the same coverage of the 95% PI. This function will store the following.csv files in the output folder:

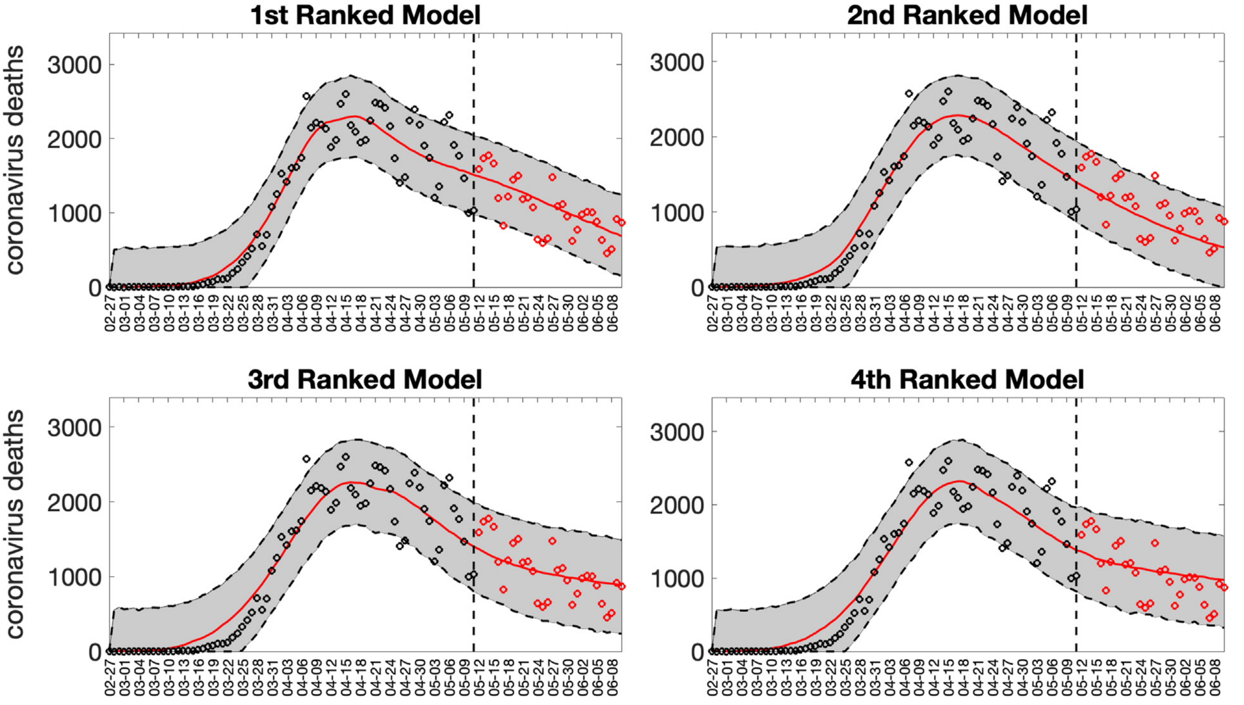


Fig. 10. 30-day forecasts derived from the top-ranking sub-epidemic models to the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 10-June-2020. The model fit (solid line) and 95% prediction interval (shaded area) are also shown. The vertical line indicates the start time of the forecast and separates the calibration and forecast periods. Circles correspond to the data points. Of note, the data associated with each of the top-ranked model forecasts are also saved as .csv files in the output folder.

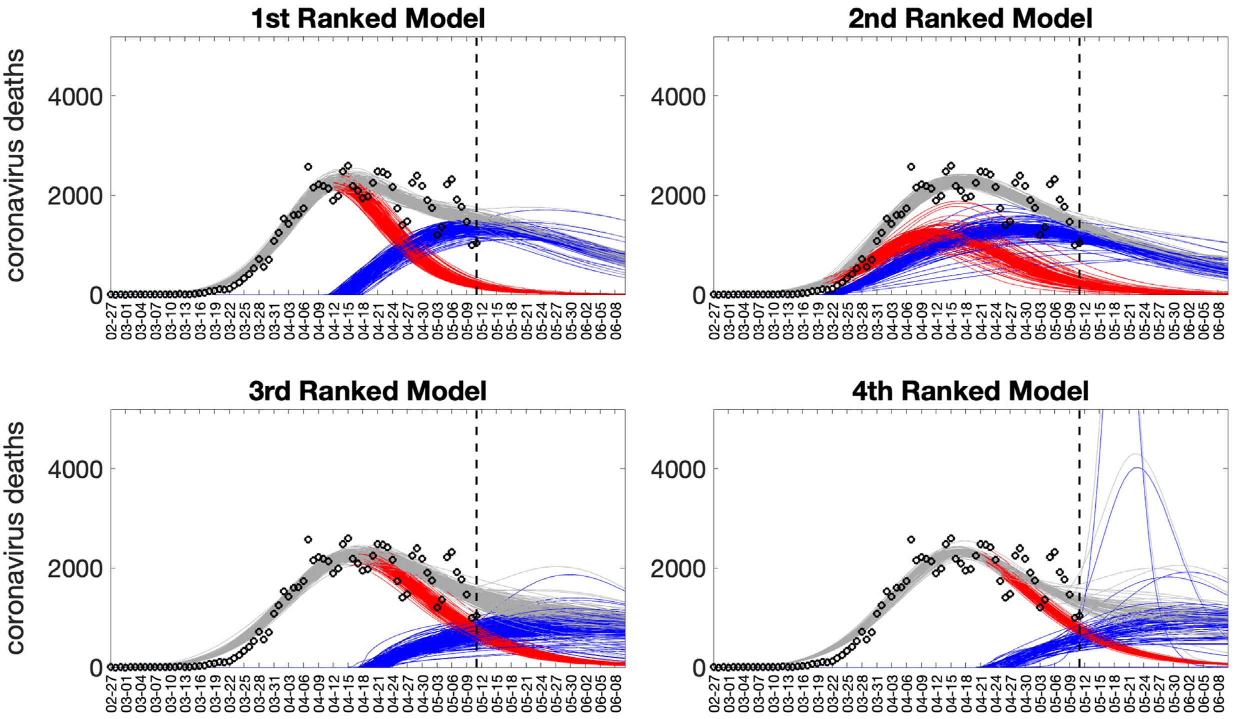


Fig. 11. Sub-epidemic profiles of the 30-day forecasts derived from the top-ranking sub-epidemic models with `<onset_fixed>=0` to the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 10-June-2020. Blue and red curves represent different sub-epidemics of the epidemic wave profile obtained from the parametric bootstrapping with 300 bootstrap realizations. Gray curves correspond to the overall epidemic trajectory obtained by aggregating the sub-epidemic curves. The vertical line indicates the start time of the forecast and separates the calibration and forecast periods.

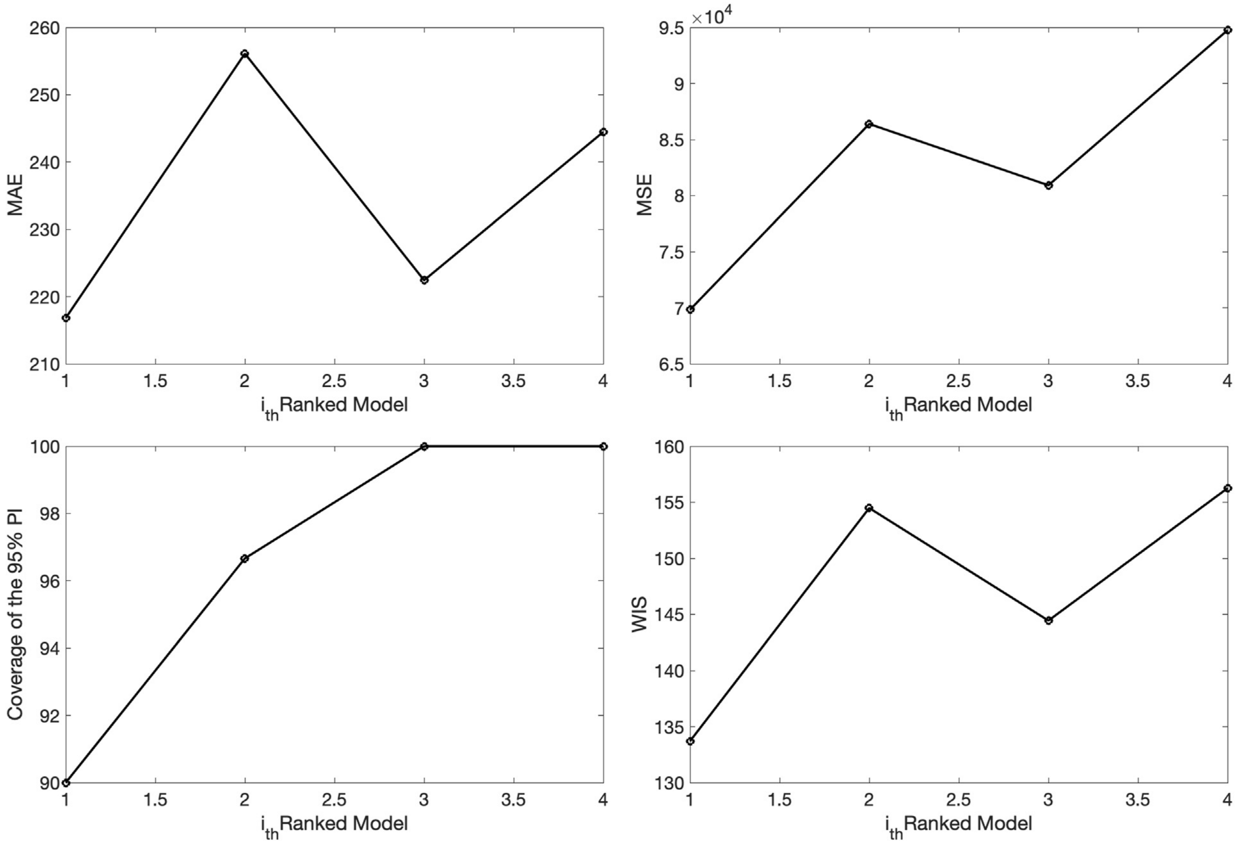


Fig. 12. 30-day forecasting performance metrics derived from the top-ranking sub-epidemic models for the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 11-June-2020. The forecasting performance metrics are also saved in a .csv data file in the output folder (‘`performance-forecasting-topRanked-onsetfixed-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv`’).

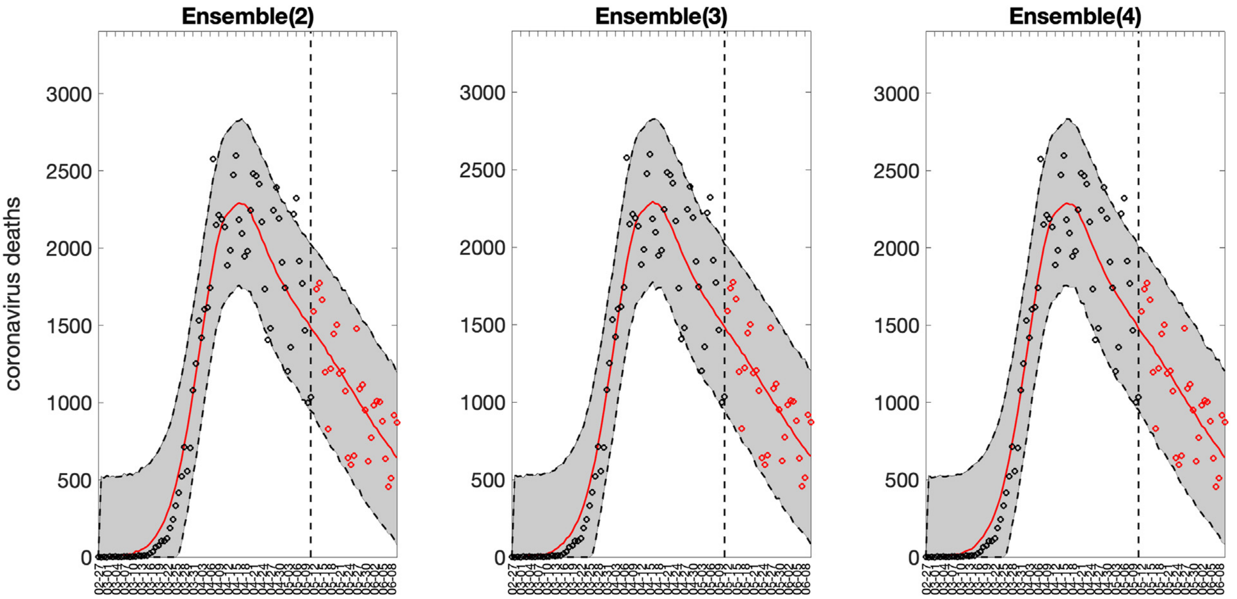


Fig. 13. 30-day sub-epidemic ensemble model forecasts (Ensemble(2), Ensemble(3), Ensemble(4)) of COVID-19 deaths in the USA from 11-May-2020 to 11-June-2020. Circles correspond to the data points. The model fits (solid line), and 95% prediction intervals (shaded area) are shown. The vertical line indicates the start time of the forecast. Of note, the data associated with each of the ensemble model forecasts are also saved as .csv files in the output folder.

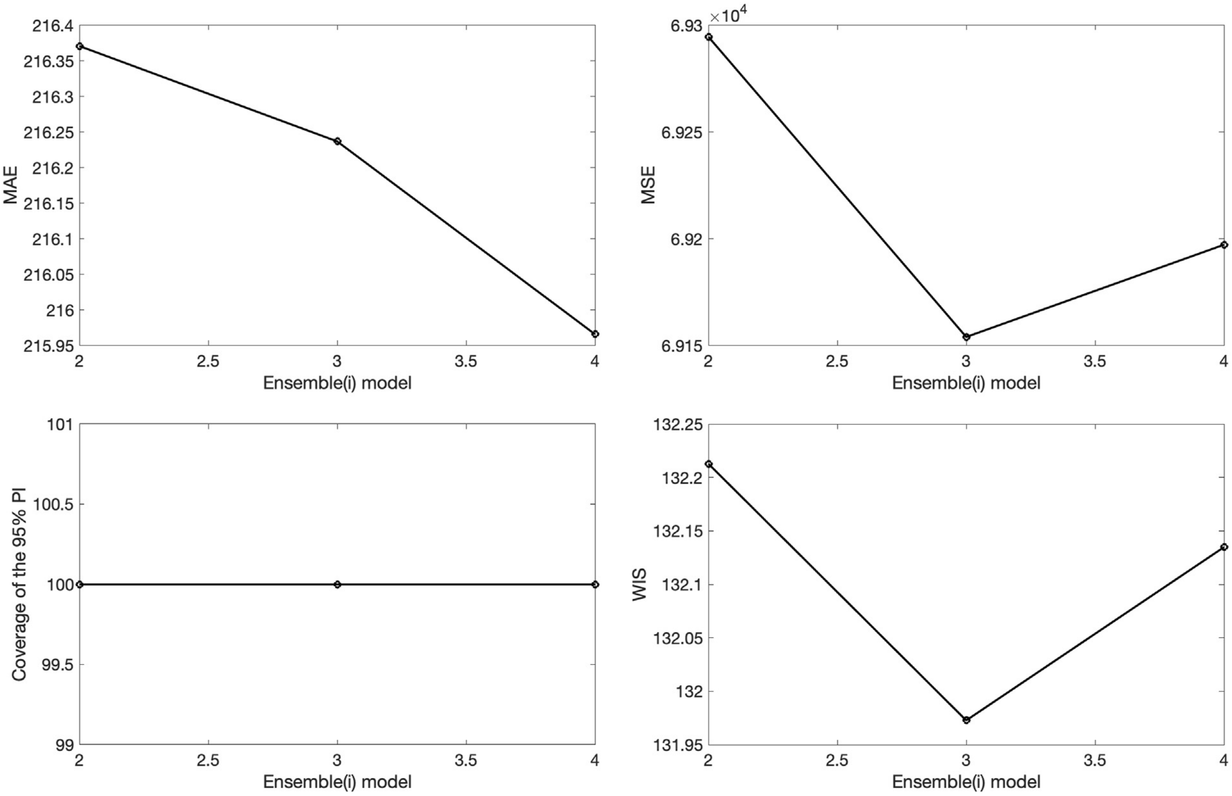


Fig. 14. 30-day forecasting performance metrics derived from the ensemble sub-epidemic models for the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 11-June-2020. The performance metrics are also saved in a .csv data file in the output folder (‘performance-forecasting-Ensemble-onsetfixed-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv’).

1) Forecasting performance metrics of the top-ranked models:
performance-forecasting-topRanked-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

2) Forecasting performance metrics of the ensemble models:
performance-forecasting-Ensemble-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weight_type-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

3) Forecasts of the top-ranked models:
ranked(1)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
ranked(2)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
ranked(3)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
ranked(4)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

4) Forecasts of the ensemble models:
Ensemble(2)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weighttype-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
Ensemble(3)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weighttype-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
Ensemble(4)-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weighttype-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

5) Sequence of doubling times of the top-ranked models:
doublingTimes-**ranked(1)**-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv
doublingTimes-**ranked(2)**-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

doublingTimes-**ranked(3)**-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

6) Sequence of doubling times of the ensemble models:

doublingTimes-**Ensemble(2)**-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weighttype-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

doublingTimes-**Ensemble(3)**-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weighttype-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

doublingTimes-**Ensemble(4)**-onsetfixed-0-flag1-1-method-0-dist-0-horizon-30-weighttype-1-daily-coronavirus-deaths-USA-area-52-05-11-2020.csv

We can also consider models with a fixed sub-epidemic start at time 0 by setting the parameter <onset_fixed>=1 in the options.m file while the other parameters are kept unchanged. Because we have previously set <npatches_fixed>=2, then the maximum number of top-ranked models is two (one model with two sub-epidemics and one model with a single sub-epidemic) since the parameter C_{thr} is no longer in the model.

After running the function Run_Fit_subepidemicFramework.m to calibrate the models to the data, we can run the function plotForecast_subepidemicFramework.m to generate the new set of forecasts with the new models.

Fig. 15 shows the 30-day forecasts derived from the top-ranking *n*-sub-epidemic models whereas Fig. 16 shows the sub-epidemic profiles of the forecasts. These forecasts indicate that the 1st ranked model, composed of two sub-epidemics, performed well while the second-ranked model, composed of a single sub-epidemic, yielded lower performance and underpredicted the trajectory of the epidemic. The corresponding forecasts derived from the weighted Ensemble(2) and the unweighted Ensemble(2) models are shown in Figs. 17 and 18.

The corresponding forecasting performance metrics for the top-ranked models and the weighted and unweighted Ensemble(2) models are shown in Table 2. For comparison, the performance metrics for the previous sub-epidemic models that include the C_{thr} parameter are also shown (e.g., <onset_fixed>=0). The top-ranked models and the ensemble models with <onset_fixed>=0 yielded similar performance whereas among the models when <onset_fixed>=1, the second-ranked model performed poorly relative to the top-ranked sub-epidemic model. Moreover, the weighted Ensemble(2) clearly outperformed the unweighted Ensemble(2), which incorporates a uniform contribution from both top-ranked models when <onset_fixed>=1.

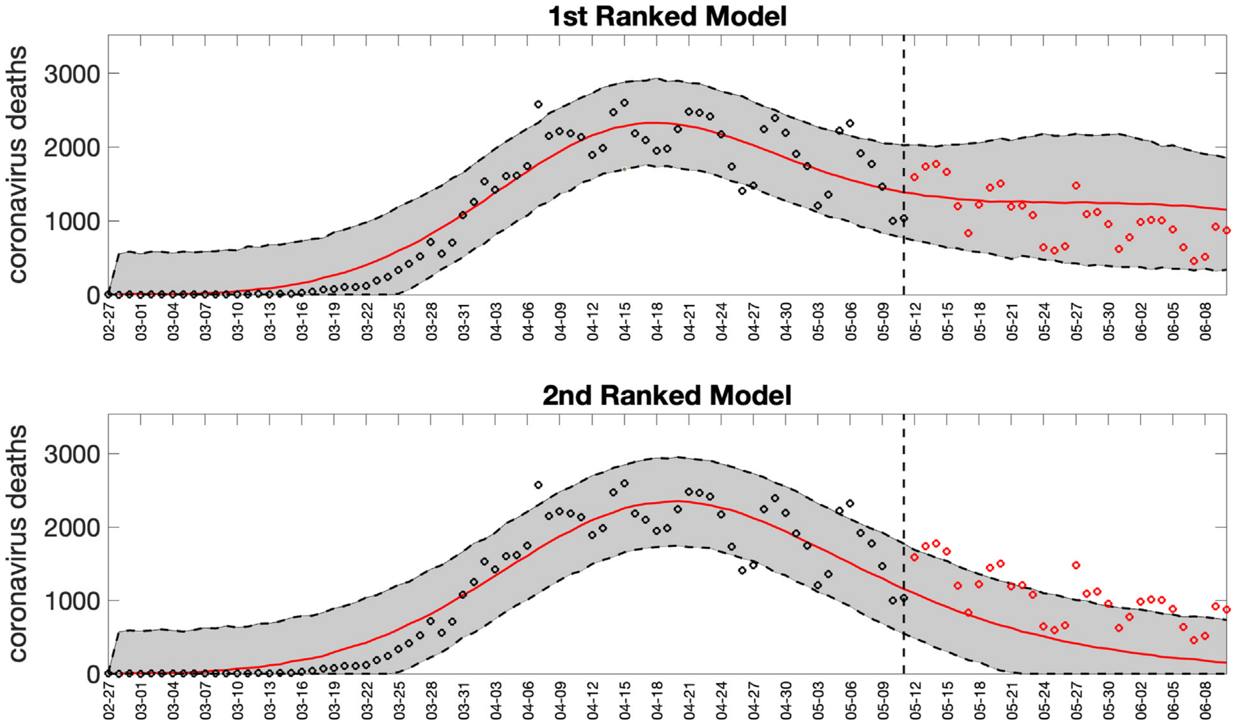


Fig. 15. 30-day forecasts derived from the top-ranking sub-epidemic models with <onset_fixed>=1 to the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 10-June-2020. The model fit (solid line) and 95% PI (shaded area) are also shown. The vertical line indicates the start time of the forecast and separates the calibration and forecast periods. Circles correspond to the data points. Of note, the data associated with each of the top-ranked model forecasts are also saved as.csv files in the output folder.

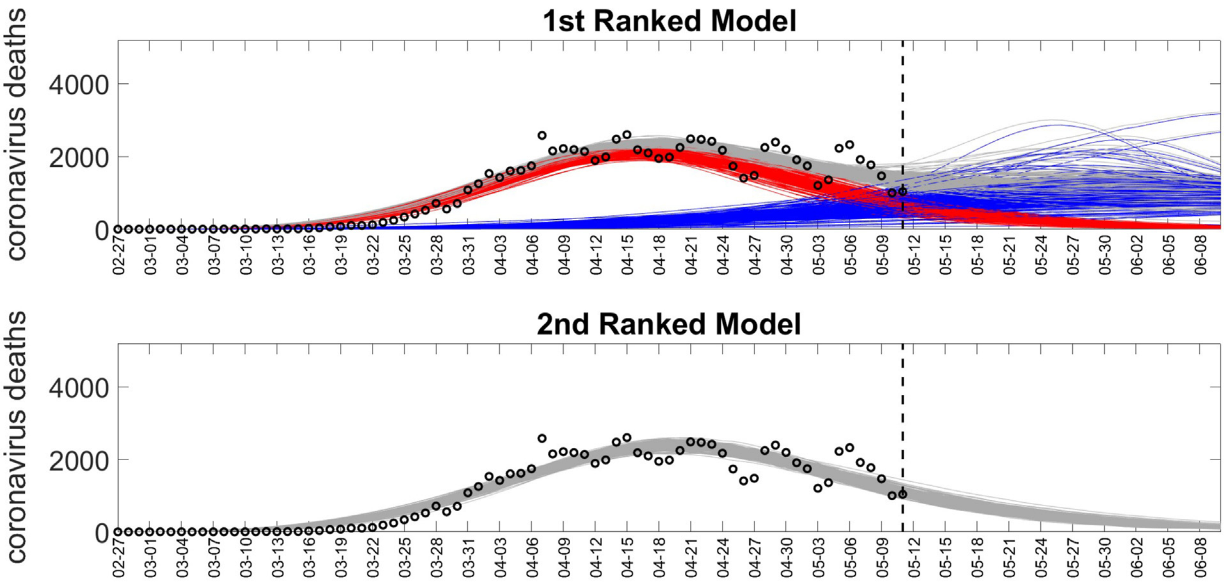


Fig. 16. Sub-epidemic profiles of the 30-day forecasts derived from the top-ranking sub-epidemic models with `<onset_fixed>=1` to the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 10-June-2020. Blue and red curves represent two sub-epidemics of the 1st ranked model whereas the second-ranked model consists of a single sub-epidemic. Gray curves correspond to the overall epidemic trajectory obtained by aggregating the sub-epidemic curves obtained from the parametric bootstrapping with 300 bootstrap realizations. The vertical line indicates the start time of the forecast and separates the calibration and forecast periods.

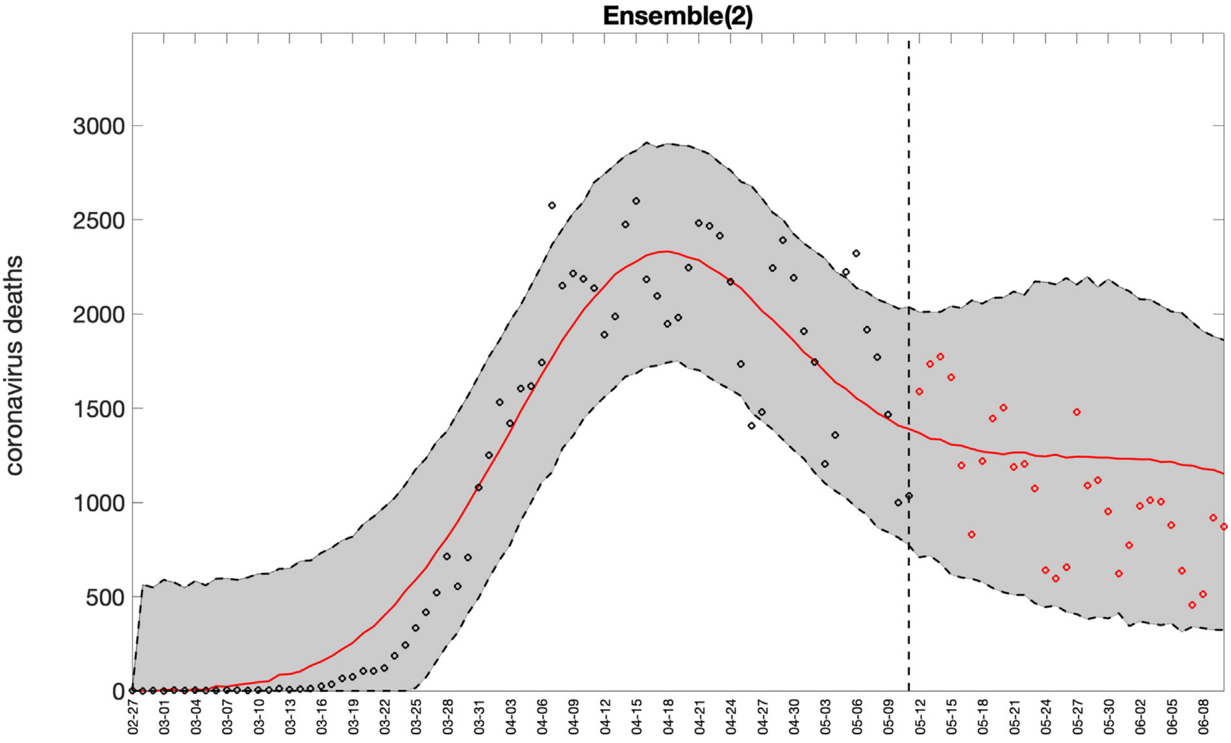


Fig. 17. 30-day sub-epidemic weighted Ensemble(2) (`<onset_fixed>=1`) forecast of COVID-19 deaths in the USA from 11-May-2020 to 11-June-2020. Circles correspond to the data points. The model fits (solid line), and 95% PI (shaded area) are shown. The vertical line indicates the start time of the forecast. Of note, the data associated with each of the ensemble model forecasts are also saved as.csv files in the output folder.

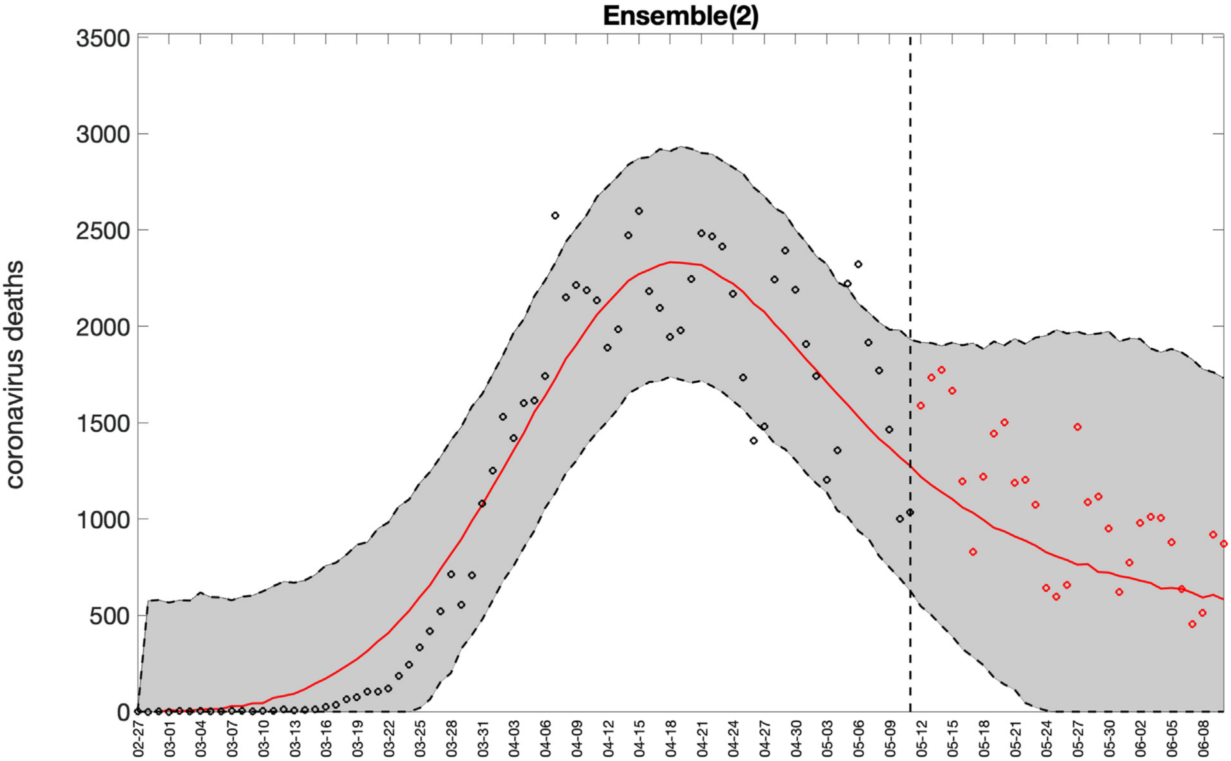


Fig. 18. 30-day sub-epidemic unweighted Ensemble(2) model (<onset_fixed>=1) forecast of COVID-19 deaths in the USA from 11-May-2020 to 11-June-2020. Circles correspond to the data points. The model fits (solid line), and 95% prediction intervals (shaded area) are shown. The vertical line indicates the start time of the forecast. Of note, the data associated with each of the ensemble model forecasts are also saved as.csv files in the output folder.

Table 2
30-day forecasting performance metrics derived from the top-ranked models as well as the weighted and unweighted ensemble(2) models with the sub-epidemic onset fixed at time 0 (<onset_fixed>=1) and the onset not fixed at time 0 (<onset_fixed>=0) based on the fit to the daily curve of COVID-19 deaths in the USA from 11-May-2020 to 11-June-2020. The top-ranked models and the ensemble models with <onset_fixed>=0 yielded similar performance whereas, among the models with <onset_fixed>=1, the second-ranked model composed of a single sub-epidemic performed poorly relative to the top-ranked sub-epidemic model consisting of 2 sub-epidemics. Moreover, the weighted Ensemble(2) clearly outperformed the unweighted Ensemble(2) model, which incorporates a uniform contribution from both top-ranked models with <onset_fixed>=1.

Model	MAE	MSE	Coverage 95% PI	WIS
<onset_fixed>=1				
1st ranked model	353.56	167430.35	100.00	198.67
2nd ranked model	566.59	380552.56	46.67	370.33
Weighted Ensemble (2)	353.56	167430.35	100.00	198.59
Unweighted Ensemble (2)	431.31	236182.87	100.00	194.44
<onset_fixed>=0				
1st ranked model	216.82	69836.18	93.33	134.36
2nd ranked model	256.12	86390.82	96.67	154.28
Weighted Ensemble(2)	216.37	69294.66	100.00	132.21
Unweighted Ensemble(2)	224.51	68614.69	100.00	134.96

4. Conclusion

In this primer we introduced a user-friendly, novel, MATLAB toolbox to fit and forecast time-series trajectories quantify parameter uncertainty, evaluate parameter identifiability, and assess forecasting performance using the ensemble *n*-sub-epidemic framework (Chowell et al., 2022). This framework has shown competitive performance when applied to real-time forecasting of epidemic outbreaks (Bleichrodt et al., 2023; Chowell et al., 2022). However, the system is broad enough and requires minimal trajectory data and should find applications in other scientific fields and help researchers produce short-term forecasts for a diversity of processes found in nature and society. The accessibility of the toolbox allows for those unfamiliar with MATLAB to execute the models with ease, thereby making it a useful forecasting tool for students and those without extensive coding or modeling backgrounds.

Computational details

The results in this paper were obtained using MATLAB Version 9.12.0.2170939 (The MathWorks Inc, 2022) along with the **fmincon** function and **Multistart** feature.

Author's contributions

G.C. conceived and developed the first version of the toolbox and wrote the first draft of the tutorial; G.C., S.D., A.B., A.T., J.M., R.L contributed to analysis and writing subsequent drafts of the tutorial. A.B. produced the tutorial video.

Funding

G.C. is partially supported from NSF grants 2125246 and 2026797 and R01 GM 130900.

Availability of data and materials

The data used in this study is publicly available in the GitHub repository (Chowell et al., 2022). The MATLAB code used in this study are available in GitHub repository: https://github.com/gchowell/ensemble_n-subepidemic_framework. The tutorial video is available at the link https://www.youtube.com/channel/UC6IzIu-pPcMLlLYAho43loQ.

Ethics approval and consent to participate

Not applicable.

CRediT authorship contribution statement

Gerardo Chowell: Writing – original draft, Validation, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Sushma Dahal:** Writing – review & editing, Formal analysis, Data curation. **Amanda Bleichrodt:** Writing – review & editing, Formal analysis, Data curation. **Amna Tariq:** Writing – review & editing, Formal analysis, Data curation. **James M. Hyman:** Writing – review & editing, Formal analysis. **Ruiyan Luo:** Writing – review & editing, Formal analysis.

Declaration of competing of interest

Gerardo Chowell is the section editor of infectious disease modelling, he was not involved in the editorial review or decision to publish this article. All 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.

Acknowledgement

We would also like acknowledge the contributions of students enrolled in the Infectious Disease Modeling Course at Georgia State University by testing the toolbox through various applied exercises.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.idm.2024.02.001.

References

Banks, H. T., Hu, S., & Thompson, W. C. (2014). *Modeling and inverse problems in the presence of uncertainty*. CRC Press.

Bleichrodt, A., Dahal, S., Maloney, K., Casanova, L., Luo, R., & Chowell, G. (2023). Real-time forecasting the trajectory of monkeypox outbreaks at the national and global levels, July–October 2022. *BMC Medicine, 21*(1), 1–20.

Bracher, J., Ray, E. L., Gneiting, T., & Reich, N. G. (2021). Evaluating epidemic forecasts in an interval format. *PLoS Computational Biology, 17*(2), Article e1008618.

Burnham, K. P., & Anderson, D. R. (2004). *Model selection and multimodel inference. A practical information-theoretic approach, 2*.

Chowell, G., Bleichrodt, A., Dahal, S., Tariq, A., Roosa, K., Hyman, J. M., & Luo, R. (2024). GrowthPredict: A toolbox and tutorial-based primer for fitting and forecasting growth trajectories using phenomenological growth models. *Scientific Reports, 14*(1), 1630.

Chowell, G., Dahal, S., Tariq, A., Roosa, K., Hyman, J. M., & Luo, R. (2022). An ensemble n-sub-epidemic modeling framework for short-term forecasting epidemic trajectories: Application to the COVID-19 pandemic in the USA. *PLoS Computational Biology, 18*(10), Article e1010602.

Chowell, G., Dahal, S., Tariq, A., Roosa, K., Hyman, J. M., & Luo, R. (2022). An-ensemblen- sub-epidemic-modeling-framework-for-short-term-forecasting-epidemic-trajectories- Application of the COVID-19. https://github.com/atariq2891/An-ensemble-n-sub-epidemic-modeling-framework-for-short-term-forecasting-epidemic-trajectories.

Chowell, G., Hincapie-Palacio, D., Ospina, J., Pell, B., Tariq, A., Dahal, S., Moghadas, S., Smirnova, A., Simonsen, L., & Viboud, C. (2016). Using phenomenological models to characterize transmissibility and forecast patterns and final burden of Zika epidemics. *PLoS currents, 8*.

Chowell, G., & Luo, R. (2021). Ensemble bootstrap methodology for forecasting dynamic growth processes using differential equations: Application to epidemic outbreaks. *BMC Medical Research Methodology, 21*(1), 1–18.

Chowell, G., Luo, R., Sun, K., Roosa, K., Tariq, A., & Viboud, C. (2020). Real-time forecasting of epidemic trajectories using computational dynamic ensembles. *Epidemics, 30*, Article 100379.

Chowell, G., Tariq, A., & Hyman, J. M. (2019). A novel sub-epidemic modeling framework for short-term forecasting epidemic waves. *BMC Medicine, 17*, 1–18.

Cobelli, C., & Romanin-Jacur, G. (1976). Controllability, observability and structural identifiability of multi input and multi output biological compartmental systems. *IEEE Transactions on Biomedical Engineering*, (2), 93–100.

Cramer, E. Y., Ray, E. L., Lopez, V. K., Bracher, J., Brennen, A., Castro Rivadeneira, A. J., Gerding, A., Gneiting, T., House, K. H., & Huang, Y. (2022). Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States. *Proceedings of the National Academy of Sciences, 119*(15), Article e2113561119.

Dimri, T., Ahmad, S., & Sharif, M. (2020). Time series analysis of climate variables using seasonal ARIMA approach. *Journal of Earth System Science, 129*, 1–16.

Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases, 20*(5), 533–534.

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association, 102*(477), 359–378.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York, NY, USA: Springer series in statistics.

Hurvich, C. M., & Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika, 76*(2), 297–307.

Hwang, E. (2022). Prediction intervals of the COVID-19 cases by HAR models with growth rates and vaccination rates in top eight affected countries: Bootstrap improvement. *Chaos, Solitons & Fractals, 155*, Article 111789.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling, 26*. Springer.

M4Competition. (2018). *Competitor's guide: Prizes and Rules*. https://www.unic.ac.cy/test/wp-content/uploads/sites/2/2018/09/M4-Competitors-Guide.pdf.

Mondal, P., Shit L, & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications, 4*(2), 13.

Muniz-Rodriguez, K., Chowell, G., Cheung, C.-H., Jia, D., Lai, P.-Y., Lee, Y., Liu, M., Ofori, S. K., Roosa, K. M., & Simonsen, L. (2020). Doubling time of the COVID-19 epidemic by province, China. *Emerging Infectious Diseases, 26*(8), 1912.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology, 47*(1), 90–100.

Pell, B., Kuang, Y., Viboud, C., & Chowell, G. (2018). Using phenomenological models for forecasting the 2015 Ebola challenge. *Epidemics, 22*, 62–70. https://doi.org/10.1016/j.epidem.2016.11.002

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., & Timmer, J. (2009). Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics, 25*(15), 1923–1929.

Ray, E. L., & Reich, N. G. (2018). Prediction of infectious disease epidemics via weighted density ensembles. *PLoS Computational Biology, 14*(2), Article e1005910.

Roosa, K., Luo, R., & Chowell, G. (2019). Comparative assessment of parameter estimation methods in the presence of overdispersion: A simulation study. *Mathematical Biosciences and Engineering, 16*(5), 4299–4313.

Roosa, K., Tariq, A., Yan, P., Hyman, J. M., & Chowell, G. (2020). Multi-model forecasts of the ongoing ebola epidemic in the democratic republic of Congo, march–october 2019. *Journal of The Royal Society Interface, 17*(169), Article 20200447.

Shamsnia, S. A., Shahidi, N., Liaghat, A., Sarraf, A., & Vahdat, S. F. (2011). Modeling of weather parameters using stochastic methods (ARIMA model) (case study: Abadeh Region, Iran). In *International conference on environment and industrial innovation*. IPCBEE.

Shanafelt, D. W., Jones, G., Lima, M., Perrings, C., & Chowell, G. (2018). Forecasting the 2001 foot-and-mouth disease epidemic in the UK. *EcoHealth, 15*, 338–347.

Smirnova, A., DeCamp, L., & Chowell, G. (2021). Mathematical and statistical analysis of doubling times to investigate the early spread of epidemics: Application to the COVID-19 pandemic. *Mathematics, 9*(6), 625.

Sugiura, N. (1978). Further analysis of the data by akaike's information criterion and the finite corrections: Further analysis of the data by akaike's. *Communications in Statistics - Theory and Methods, 7*(1), 13–26.

Tariq, A., Chakhaia, T., Dahal, S., Ewing, A., Hua, X., Ofori, S. K., Prince, O., Salindri, A. D., Adeniyi, A. E., & Banda, J. M. (2022). An investigation of spatial-temporal patterns and predictions of the coronavirus 2019 pandemic in Colombia, 2020–2021. *PLoS Neglected Tropical Diseases, 16*(3), Article e0010228.

Tektaş, M. (2010). Weather forecasting using ANFIS and ARIMA models. *Environmental Research, Engineering and Management, 51*(1), 5–10.

The MathWorks Inc. (2006). Fmincon — find minimum of constrained nonlinear multivariable function. https://www.mathworks.com/help/optim/ug/fmincon.html#busog7r_vh.

The MathWorks Inc. (2010). *Multistart — find multiple local minima*. https://www.mathworks.com/help/gads/multistart.html#d124e59127.

The MathWorks Inc. (2022). *Matlab — the language of technical computing*. The MathWorks Inc. Version 9.12.0.2170939 (R2022a) https://www.mathworks.com/products/matlab.html.

Viboud, C., Sun, K., Gaffey, R., Ajelli, M., Fumanelli, L., Merler, S., Zhang, Q., Chowell, G., Simonsen, L., & Vespignani, A. (2018). The RAPIDD ebola forecasting challenge: Synthesis and lessons learnt. *Epidemics, 22*, 13–21.

Wallinga, J., & Lipsitch, M. (2007). How generation intervals shape the relationship between growth rates and reproductive numbers. *Proceedings of the Royal Society B: Biological Sciences, 274*(1609), 599–604.

Yan, P., & Chowell, G. (2019). *Quantitative methods for investigating infectious disease outbreaks, (Vol. 70)*.. Cham, Switzerland: Springer.