Disease outbreaks and metrics

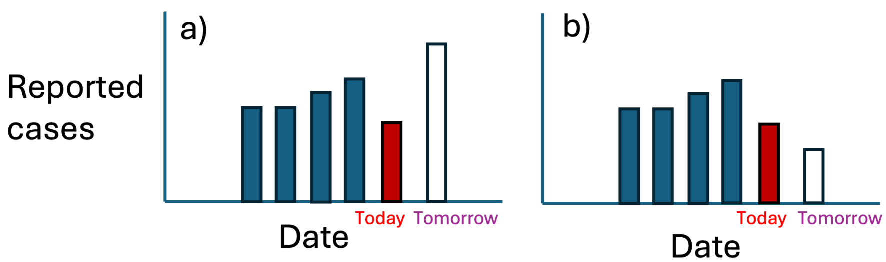
Table of contents

Overview

When an infectious disease outbreak begins, a time-sensitive question arises: "tomorrow, are things getting better or worse?"

A great deal of research has gone into how to answer this question, including the development of forecasting tools to attempt to predict what might be coming, as well as data streams and metrics to summarize and understand that data. Here we focus more on the latter approach.

Suppose you work at a public health agency, and you have the following reported case data in blue:



You may want to know, are cases tomorrow going to be a) higher than today or b) lower than today. Just looking visually, either seems plausible: in case a) perhaps today's cases are a outlier, and the true trend will continue upwards, and in case b) perhaps today's cases are not an outlier, and tomorrow's cases will be lower.

However the process that generates these new cases, i.e. infection, has already occurred in most cases. A more helpful question might be: Are people still infecting other people in sufficient numbers that we can expect cases to generally keep increasing? Reported cases are a lagging indicator of the current state of disease transmission. If we understand this dynamic, then we will be able to predict how many cases we expect to be coming in the near future and if control measures are effectively slowing transmission.

This is what the **effective reproductive number**, $R(t)$, aims to estimate. The reproductive number estimates the average number of people an infectious individual will infect at time t. This is typically done daily. The reproductive number is estimated from case count data like that shown in the plots above. But there is another critical piece of information to indicate when reported cases might have been infected. This is:

How long does it take for an infected person to infect others? This is described by the generation interval. This can be summarized by a mean which would give the average amount of time between an infector and their infectee. But more often it is described by a statistical distribution. For example, the infector of an infected individual would have been infected 1 day prior with 30% probability, 2 days prior with 40% probability, or 3 days prior with 30% probability.

The generation interval is central to estimating the reproductive number. The generation interval can estimated by a number of methods, including analyzing data of infector-infectee pairs.

Knowing $R(t)$ can help you begin to make an informed guess as to the current state of a disease outbreak and near term forecasts, as it has the following values and interpretations *at a specific point in time*:

R(t)	Interpretation at time t	Outbreak is ...
< 1	Each infected person infects *on average* fewer than one additional person	shrinking
$= 1$	Each infected person infects *on average* about one additional person	stable
> 1	Each infected person infects *on average* more than one additional person	growing

However, estimating $R(t)$ is not straightforward, and is the subject of a wealth of academic research and proliferation of software packages. Guidance in choosing a choosing a method (and a package) is the purpose of this website.

How to choose a tool to estimate $R(t)$

There has been a proliferation of software tools that make inference about the current state of an infectious disease outbreak.

Important to keep in mind when choosing a tool to estimate $R(t)$ is this fact: $R(t)$ is a *latent* variable, which means *cannot be measured directly*. Instead, it can only be estimated from observable variables (like reported case counts).

The ideal estimator of $R(t)$ requires a list of the number of newly *infected* cases by infection date and the generation interval. This is because we want to know about the state of disease based on when people are infected, not when they report having symptoms.

In reality we usually only observe the new number of newly *reported* cases and can only estimate the serial interval, which is the time between symptom onset of an infector-infectee pair. In this case the estimate of $R(t)$ will lag reality without some adjustments.

Each software package that estimates $R(t)$ makes different adjustments and assumptions about how these parameters relate, which leads to variations in estimated $R(t)$ *even if the same input* *data are used*. In addition, different packages require different levels on input data to provide additional robustness in estimated outputs.

The purpose of this document

Therefore, the purpose of this document is to provide guidance about which $R(t)$ estimation software to choose for different analytical goals. First, see our Example outbreak for the different components of disease outbreak that can be modeled differently. Next, see our Decision tool for how to choose software for different analytical goals.

Funding, authors, and acknowledgements

This work is supported by CDC grant NU38FT000013.

The lead authors of this document are at Boston University in the School of Public Health:

- Chad Milando, Laura White

Many additional co-authors contributed to this document including:

- Anne Cori, Brennan Klein, Katelyn Gostic, Alessandra Urbinati, Guillaume St-Onge, George Vega Yon, Kaitlyn Johnson, Christine Sangphet, …

Example outbreak

Decision matrix

$R(t)$ has two main uses:

1. **Retrospective** understanding of the dynamics of historical outbreaks, and
2. **Real time tracking** ongoing infectious diseases.

For 1, one might wish to understand the impact on transmission of vaccines or non pharmaceutical interventions, such as masking or physical distancing.

For real time tracking of ongoing infectious diseases, there is often interest in determining if the current outbreak is getting worse, better or staying the same. In this case, live dashboards are often used to track $R(t)$ as new data on diagnosed cases emerges. This is currently done for COVID-19 and Influenza by the CDC and CA (add refs).

In either application, **delay distributions play a key role in estimating $R(t)$ for new infections.** You can use the decision tool below to help choose which software package(s) may be right for your application:

1. Look first for the desired output that you want to produce
2. Then make a decision about whether you want to incorporate delay distributions.
3. Finally, estimate $R(t)$ using the packages that are appropriate for your use case

See below the table for the assessment framework used to decide which packages to recommend.

> **Focus of this matrix**
>
> The table below focuses on **R packages** that esimate $R(t)$ using **reported cases**. Future efforts include expanding this table to include **alternative data sources** (e.g., wastewater) and packages in **other coding languages** (e.g., Python). See the packages list for an index of currently reviewed packages.

Table 2: Decision matrix for choosing an R package for R(t

Desired output	Incorporates D
Forecasting: what will R(t) be next week	
Nowcasting: what was R(t) in the past week	

Historical: what was R(t) over the past month

Assessment framework

An objective comparison of the performance of the methods in these packages would be highly complex, given the following challenges:

- These packages are really a combination of mathematical modeling, available data, and implementation. Any evaluation would have to disaggregate these features.
- Some of the most widely-used packages are not accompanied with a peer-reviewed manuscript that describes or evaluates the theory behind modeling choices.
- Each package contains a subset of the methods below for constraining $R(t)$ in time, but with subtle variations in implementation and presentation that are often not well-documented and have large implications on evaluation metrics.
- Some packages have not been recently updated, and even those that have are not maintained on Installation, instead leaving updates on a development version on GitHub.
- Performance may vary widely considering additional factors like ease of implementation and computational time.
- it also may be the case that some methods of temporal smoothing work better in some cases versus other (very low case counts, rapid changes)

Indeed, many published validation efforts are often not "apples to apples", i.e., comparing two models that are using different amounts of information in estimating R(t). For example, comparing a model that has used only data before time before t to estimate R(t) versus a model that uses the entire historical record to estimate R(t) at time t.

Instead, we present some quantifiable reflections on various aspects of utilizing each package.
:

Table 3: Assessme

Category	Notes
Features	
Ability to nowcast/forecast	Does the package have functionality to incorporate both right-tru
Incorporates delay distributions	Does the package have methodology for incorporating delay distr
Estimates expected cases	Does the package provide an estimate of expected cases and/or in
Communicates uncertainty	Does the package detail how uncertainty is incorporated into pres
Documentation	
Documentation of package methods	Is there a written report (or published manuscript) that describes
Documentation of package implementation	Are there sufficiently detailed vignettes that would permit a new

Part I

Explanation of methods

To aid with interpretation of package outputs, we summarize the currently used inputs, data, methods and assumptions in $R(t)$ estimation across the following categories:

: How the relationship between $R(t)$ and infections is defined : How $R(t)$ is constrained using distributions for key variables : How $R(t)$ is constrained over time : Additional data and distributions that are used to constrain $R(t)$: Inference frameworks that are used to estimate $R(t)$

We limit the methods discussed here to those for estimating historical to present-day $R(t)$ values using **daily case count data**, where a case can be flexibly defined as an individual with a reported positive test (either through healthcare-seeking behavior, routine surveillance, or a hospital admission).

Other methods not discussed here include:

- inference of $R(t)$ exclusively from alternative data sources (e.g., genetic data (Walker et al. 2013), behavioral data (Bokányi et al. 2023), or viral loads in waste-water (Huisman et al. 2022)),
- calculations from compartmental, agent-based models, or network Bettencourt and Ribeiro (2008).

We also limit the discussion to packages in the statistical software R (R Core Team 2022), which may exclude some packages in other software programs that combine many of the methodological considerations discussed below (Yang et al. 2022).

We attempt to harmonize the mathematical choices between each package using terminology from each.

Open research questions

Several open research questions remain, and are not disucssed here further. These include:

- the ability to deal with very low case counts in sub-regions of geographic areas of interest
- how to deal with data drop-out

Relating infections to $R(t)$

Overview

There are two primary classes methods of estimating $R(t)$ from case count data that are used in most R software packages.

(1) The first class of methods assumes there is a formulaic relationship between infections and reproduction number, a relationship known as the renewal equation (Fraser 2007). These infections are then assumed to result in (some fraction of) the observed cases.

(2) A second class of methods involves empirically calculating a quantity that approximates the latent quantity represented by a reproduction number by fitting a curve to the case count time-series and finding the time-varying slope in log space (and then performing other transformations). Empirical calculations are discussed in detail below in our examination of ways in which $R(t)$ is constrained over time.

Renewal equation estimates of R(t)

The renewal equation relates $R(t)$ and infections on day t, $I(t)$, using a third parameter known as the generation interval.

The generation interval, ω, is the time between infection in the infector and infection in the infectee, and assuming independence is the linear combination of incubation time, the time between infection and symptom onset in an individual, and transmission time, the time between symptom onset in the infector and infection of the infectee (Lehtinen et al. 2021).

In this document we use the generation interval ω described by a probability mass function with non-zero values from day 1 (assuming that disease incubation takes at least 1 day) to a maximum day s, i.e., the longest interval between infections in infector and infectee.

Taking care to note that $R(t)$ is undefined on day 0 since there has been no transmission yet (and assuming the initial infections are $I(0)$), we sum over days to acquire the renewal equation:

$$I(t) = R(t) \sum_{i=\max(1,t-s+1)}^{t} \omega(i)\, I(t-i).$$

For brevity, we write the inner sum of (Eq.1) as:

$$\Lambda(t) = \sum_{i=\max(1,t-s+1)}^{t} \omega(i)\, I(t-i).$$

The assumptions of this formulation, as per Green et al. (2022), are that incident infections can be described deterministically within each window of $[t-s+1,t]$ and that the generation interval distribution does not change over the modeling time.

Serial interval

A similar parameter to the generation interval is the serial interval, which is the time between symptom onset in the infector and symptom onset in the infectee. The serial interval and generation interval are interchangeable if the incubation time is independent from the transmission time, and some formulations of the renewal equation use serial interval.

Exponential growth rate

A common reframing of the renewal equation is to equate $R(t)$ with an exponential growth rate, r. Under specific conditions and within a small time window ($[t-s+1,t]$), infections in the early stage of an outbreak can be assumed to grow exponentially at a constant rate r (Wallinga and Lipsitch (2006)). Using the time window $[t-s+1,t]$ and assuming some initial infections k, $R(t)$ for $[t-s+1,t]$ can be defined using r and ω:

$$I(t) = ke^{r}t$$

$$R(t) = [\sum_{i=\max(1,t-s+1)}^{t} \omega(i)\, e^{-ri}]^{-1}$$

Again, we will omit the writing the bounds for time in remaining formulae. A single $R(t)$ value, say R_0, can be substituted into an expression for the infection attack rate, z (Musa et al. (2020)), or in the final size equation (Ma and Earn (2006)), to estimate the proportion of all individuals that were affected by a disease with this R_0:

The major difference between calculating $R(t)$ from a renewal equation or an exponential growth rate equation is whether $I(t)$ is used. If for a given time window both r and   can be estimated independently, then $R(t)$ can be inferred without infection data. Otherwise, infection data are needed to estimate $R(t)$.

Solving for $R(t)$

> ⚠ Solving for $R(t)$
>
> Using the renewal equation and given that $I(t)$ and ω are known, $R(t)$ can be solved for algebraically starting with R(t=1) and iterating forwards in time. However, this will produce highly volatile estimates of $R(t)$ that recover the incidence curve directly.

Solving directly for $R(t)$ at every timestep is undesirable for several reasons:

- observed infections are the result of a noisy process with random day-to-day variations, which do not reflect an underlying change in infectivity;
- real-world infection data are rarely complete, especially in an emerging epidemic, meaning that a certain amount of uncertainty must be incorporated into any estimation framework;
- infection incidences, $I(t)$, are the data of interest but are cannot be observed directly, so many calculations instead use the observed reported cases, $C(t)$, which requires some additional processing to incorporate into calculations of $R(t)$.

Therefore, a variety of constraints on $R(t)$ are added in the inferential process: using distributions on key variables, placing restrictions on how $R(t)$ varies through time, and with additional data sources and delay distributions. These choices dictate which estimation framework is used, which can add additional constraints.

Empirical estimates of $R(t)$

In contrast to models that assume that the renewal equation defines the relationship between infections and $R(t)$, smoothing or regression models calculate time-varying $R(t)$ directly from the slope of the log of the infections time-series. Using this method, the relationship between $R(t)$ and infections is empirically defined, being only constrained by the smoothing parameters of curve fit to infections data. Several R packages contain methods for this type of smoothing, e.g., EpiLPS, EpiNow2

Distributions for key variables

Distributions for key variables

A primary component of constraining $R(t)$ is how distributions are used to constrain key variables in $R(t)$ estimation: for $I(t)$, ω, and for $R(t)$ itself.

Assuming some prior distributions for $R(t)$ and the generation interval permit an analytical solution for the posterior distribution of $R(t)$, as in EpiEstim. These simplifying assumptions greatly constrain the space of possible $R(t)$ values and thus calculation times are relatively fast.

Other software packages, such as EpiNow2, have more flexibility at the cost of somewhat higher computational runtime and resources.

Distributions used to define new offspring from cases

Another primary component of constraining $R(t)$ is how distributions are used to define the next generation of infections, or $I(t)$ as a function $I(t-i)$, for $0 \leq i < t$.

The renewal equation provides a mechanism for estimating the next batch of infectees that occur due to transmission from the current round of infectors, a branching process. The $I(t)$ calculated in the renewal equation represents the expected value of a discrete distribution. To account for stochasticity in estimating $R(t)$, we must specify this distribution.

The **Poisson distribution**, in which the mean and variance parameter $\lambda(t) = I(t)$, is used to represent the probability that a given number of events (i.e. infections) occur within a given time interval (i.e. one day), assuming that these events are independent. This is the simplest option to represent variation in infections. However, infections are often not independent, but can occur in clusters, e.g. in supersreading events. If this is an important factor in the spread of the infection, it is better to use a **Negative Binomial distribution**, with a mean parameter equal to $I(t)$ and a fitted size paramter to account for the "over-dispersion" of infections.

Constraining $R(t)$ over time

Overview

The largest variety in constraints of $R(t)$ exists in methods that impose structure on how $R(t)$ varies with time. Each method confers various assumptions and implications for resulting estimates of $R(t)$, and new methods represent a large area of innovation with regards to real-time infectious disease modeling. With these constraints, we can make inference from sampled case-count data as a signal of unobserved infections in the larger unobserved population.

Fixed sliding windows

A straightforward method of imposing structure on $R(t)$ over time involves constraining $R(t)$ to be drawn from the same distribution within moving time subsets, called sliding windows.

We add the prefix of "fixed-size" to distinguish from methods that may adapt the size of the sliding window over time.

Consider the scenario where $I(t)$ are drawn from a series of Poisson distributions and where $R(t)$ are drawn from a series of Gamma distributions. Using a sliding window size, τ, of 5 days:

- $R(t)$ on days 2 to 6 are assumed to be drawn from the Gamma distribution with parameters a_1 and b_1
- $R(t)$ on days 3 to 7 are drawn from a Gamma distribution with parameters a_2 and b_2

In the above scenario, days 3 through 6 are in both windows and thus will be values that could be reasonably drawn from Gamma distributions with either a_1 and b_1 or a_2 and b_2.

Using an assumption of Gamma distributions for the prior distribution of ω and $R(t)$, Cori et al. (2013) analytically derived a posterior distribution $R(t)$ using fixed-size sliding windows. This distribution has the following directly calculated (rather than inferred) mean and coefficient of variation of $R(t)$:

$$E[R(t)] = \frac{a + \sum_{i=\max(1,t-\tau)}^{t} I(t)}{1/b + \sum_{i=\max(1,t-\tau)}^{t} \Lambda(t)}$$

$$C.V.[R(t)] = \frac{1}{a + \sum_{i=\max(1,t-\tau)}^{t} I(t)}$$

Thus, sliding windows with larger τ improve the stability of the estimate of $R(t)$ (as compared to smaller τ) because the coefficient of variation of $R(t)$ decreases as number of infections increases (see Web Appendix 1 of Cori et al. (2013)). Fixed sliding windows are a key feature of the EpiEstim package.

Drawbacks of fixed sliding windows

There are limitations of the fixed sliding window approach, articulated well in Gostic et al. (2020) and summarized here:

- There is no posterior distribution for the expected value of incidence.
- In the fixed size sliding window approach, τ must be explicitly defined prior to inference. Shorter τ will lead to quicker response but more variable estimates of $R(t)$, which increases the risk of over-fitting. At the extreme, if the τ is set to 1 day, the resulting $R(t)$ will recover exactly the input case data. The default recommendation for τ is one week (7 days)
- In addition, there is debate in the literature about where in time the estimate of $R(t)$ for each window should go: Gostic et al. (2020) recommends using the midpoint of each sliding window rather than time t.
- The choice of both τ and the location of the estimate of $R(t)$ within each window results in gaps in predictions for $R(t)$, barring other modifications:
  - at the beginning of the modeling period, if not enough cases have occurred (Web Index 4 of Cori et al. (2013) suggests that calculation of R(t) should start at least one serial interval's worth of time after the 12th case is recorded)
  - at the end of the modeling period to account for reporting delays or time between the midpoint of   and the end of .

Modifications of EpiEstim

Several packages modify the functionality of EpiEstim. These modifications include:

- fitting τ to minimize the Accumulated Prediction Error, as in the package APEestim
- fitting $R(t)$ via harmonizing the EpiEstim method and methods from the case reproduction number in a variational method, as in the EpiInvert package.
- and dissaggregation of data into smaller time units, as in the ern package.

Random walk

Another method of constraining how $R(t)$ evolves in time is to define the relationship between $R(t)$, infections, and time in a random walk or auto-regressive framework.

In this framework, there are latent or unobserved variables, e.g., $R(t)$, that depend on observed variables, e.g., $I(t)$ via the renewal equation, and the evolution of the unobserved variables through time can be parameterized.

The auto-regressive component means that the current value of $R(t)$ is correlated via some mechanism with $R(t-1)$ (and potentially other past values).

The packages EpiNow2, epinowcast, and WhiteLabRt contain an implementations of a random walk procedure that look generally as follows:

$$f(R(t)) = f(R(t-1)) + N(0, \sigma_R)$$

where σ_R has a user-defined prior, e.g., with hyperparameters ρ and φ:

$$\sigma_R \sim HalfNormal(\rho, \varphi)$$

The random walk implies that adjacent $R(t)$ values may be drawn from similar or even the same distribution, and would be correlated in time based on previous values.

The function f can be a transformation of $R(t)$, e.g. in log space to correct for the skewness of $R(t)$. This can help provide a variable that is more Gaussian, provide a variable that obeys the properties that we expect from $R(t)$ (i.e., is non-negative), and aid in interpretability.

Filtering

Filtering is another way that $R(t)$ is constrained in time. Filtering is similar to a random walk constraint, in that a hyperparameter controls the amount of difference between $R(t)$ in adjacent time-steps, but with a different functional form to a random walk.

One way that a filter could be implemented is in a Hidden Markov Model (Rabiner and Juang (1986)). A simple forward-looking linear filter for $R(t)$ in an Hidden Markov Model might look as follows, with a tuning parameter (η) to influence the amount that $R(t)$ can vary between time-steps and a standard white noise component (ϵ):

$$R(t) = R(t-1) + (\eta \sqrt{(R(t-1))}) * \epsilon(t-1)$$

The package EpiFilter implements a two-stage filtering and smoothing method for estimating $R(t)$. A key innovation of EpiFilter is that the states of historical $R(t)$ are constrained to a predefined set of values; this dramatically reduces calculation time. The smoothing stage refines estimates of $R(t)$ by incorporating future incidence, in this way using all available data in estimates of historical $R(t)$. These modeling steps help avoid $R(t)$ instability when infections are low and instability at the beginning and (more importantly) the end of the modeling period.

Another way that filtering can be implemented is across the entire R(t) time-series, as in the packages RtEstim and EpiLPS.

Notably, $R(t)$ estimated in this way thus contains information about past and pending infections, e.g., for $R(t = i)$, the smoothing step will affect $R(t = i)$ using information from $0 < i \leq t_{max}$. This complicates comparisons to outputs from other methods that only use historical information to estimate $R(t)$, e.g., estimates for $R(t = i)$ containing only information from $t < i$.

Gausian Process models

Gaussian Process models (Schulz et al. (2018)) are yet another flexible method of constraining the evolution of $R(t)$ in time than the methods discussed thus far.

In Gaussian Process modeling, a family of basis functions are fit to available data, permitting inference about continuous processes without needing to a priori define where inflection points occur.

The core of Gaussian Process operations is a kernel, which is used to assess the similarity between input vectors, say x and x'.

There are many options for potential kernels, and each contains different hyperparameters that are used to control the amount of smoothing that is enforced, as well as other factors. One such choice is the squared exponential kernel:

$$k(x, x') = \alpha^2 \exp(\frac{-(x - x')^2}{2l^2})$$

In this kernel, the hyperparameters are the length scale, l, which controls the smoothness of the model, and the magnitude, , which controls the range of values used in the fitting process. These parameters can be given prior distributions and fit using optimization.

EpiNow2 contains options to use Gaussian Process models to control how R(t) evolves in time. As one example, the relationship between first difference values of R(t) can be constrained using a zero-mean Gaussian Process model with the above kernel as the covariance function:

$$\log R(t) = \log R(t-1) + GP(0, k(R(t), R(t')))$$

The advantage of Gaussian Process models is that R(t) is enforced to change smoothly in time . Limitations include complexity and computational time (Riutort-Mayol et al. (2020)), which in general means Gaussian Process runtimes and required computational resources are considerable as compared to other methods.

Additional data

Estimates of R(t) can also be improved using additional data. , you can beef up the calculation by including other pieces of information about counts.

Reconstruction of missing data

Section in progress

Delay distributions

Section in progress

Importantly, the definition of R(t) is linked to the data that are being used, so models that calculate a similar quantity as R(t) but instead from infections, symptom onset, or reports are important quantities but differ in definition from the instantaneous reproduction number R(t) as defined throughout the literature.

Several packages have been created to extend EpiEstim to use delay distributions:

- bayEStim: Our method extends that of Cori et al (2013), adding Bayesian imputation of missing symptom onset dates, imputation of infection times using an external estimate of the incubation period, and an adjustment for reporting delay.
- estimateR involves combining various delay distributions with EpiEstim
- EpiInvert also has methods for including delay disributions with EpiEstim

Additional data streams

Section in progress

Additional data include: clinical data, wastewater

Inference frameworks

This page describes the different software / mathematical implementations of the solving the smoothing equations described previously.

Bayesian optimization

Section in progress

Maximum Likelihood optimization

Section in progress

Part II

Packages

APEestim

REF	Parag and Donnelly (2020)
Docs	None
Github	Github
Last commit	Feb 12, 2021
Installation	None, this is code to augment EpiEstim

Description

Copied from the developer site

APEestim estimates the time-varying reproduction number on cases by date of infection (using a similar approach to that implemented in EpiEstim).

The quality of this estimate is highly dependent on the size of a smoothing window (k) that is employed. This code presents a method for optimally selecting k in a manner that balances reliable R(t) estimation with short-term forecasts of incidence. This method is based on the accumulated prediction error (APE) idea from information theory.

Methods

This package aims to improve upon the limitation of fixed sliding windows, specifically by optimizing the choice of the window size.

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	No
Estimates expected cases	No
Communicates uncertainty	Yes
Validation	
Documentation of package methods	Yes

Documentation of package implementation	No

Sample Code

See this file in the Github repo.

bayEstim

REF	Lytras et al. preprint
Docs	None
Github	github.com/thlytras/bayEStim
Last commit	Aug 3, 2020
Installation	None

Brief description

Package never submitted to CRAN, no further action taken

earlyR

REF	None
Docs	repidemicsconsortium.org/earlyR/articles/earlyR.html
Github	github.com/reconhub/earlyR
Last commit	October 27, 2020
Installation	CRAN

Brief description

Copied from the developer site

Implements a simple, likelihood-based estimation of the reproduction number (R0) using a branching process with a Poisson likelihood. This model requires knowledge of the serial interval distribution, and dates of symptom onsets. Infectiousness is determined by weighting R0 by the probability mass function of the serial interval on the corresponding day. It is a simplified version of the model introduced by Cori et al. (2013).

Methods

This package does not constrain R in time, instead this is meant to predict a single R value (R0) and then uses this to nowcast and forecast cases.

Assessment

Features	
Ability to nowcast/forecast	Yes
Incorporates delay distributions	No
Estimates expected cases	Yes
Communicates uncertainty	Yes
Validation	
Documentation of package methods	No
Documentation of package implementation	Yes

Sample Code

This vignette gives a basic example of usage

Epidemia

REF	Flaxman et al. (2020)
Docs	imperialcollegelondon.github.io/epidemia
Github	github.com/ImperialCollegeLondon/epidemia
Last commit	Feb 12, 2021
Installation	Broken, see below

Brief description

This package cannot currently be installed so no further analysis is provided at this time

EpiEstim

REF	Cori et al. (2013), Nash et al. (2023)
Docs	mrc-ide.github.io/EpiEstim
Github	github.com/mrc-ide/EpiEstim
Last commit	Aug 30, 2024
Installation	CRAN

Brief description

Copied from the developer site

EpiEstim is a tool to estimate the time-varying instantaneous reproduction number during epidemics. In order to estimate Rt, EpiEstim needs to be supplied with an estimate of the serial interval distribution (step A) and the incidence of confirmed cases (step B). Once you have an incidence object (based on the dates of symptom onset) and information on the serial interval distribution, we can use the renewal equation (a form of branching process model) to estimate Rt. The incidence of symptom onset at time t is approximated by a Poisson process using the renewal equation.

Note: EpiEstim runs quickly owing

Methods

This package contains the following methods:

- fixed sliding windows

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	No, although some right-censoring is included
Estimates expected cases	No

Communicates uncertainty	Yes
Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample Code

This vignette gives a basic example of usage of EpiEstim.

The end of this vignette suggests using the `projections` package to estimate future cases, and we cannot recommend this package. The estimation of future values of R(t) in this package comes from resampling different past values of R(t) rather than trends dervied from recent infections.

EpiFilter

REF	Parag (2021)
Docs	
Github	https://github.com/kpzoo/EpiFilter
Last commit	Dec 9, 2023
Installation	

Brief description

Copied from the developer site

Maximally informed, mean square error optimised estimates of reproduction numbers (R) over time.

Uses Bayesian recursive filtering and smoothing to maximise the information extracted from the incidence data used. Takes a forward-backward approach and provides estimates that combine advantages of EpiEstim and the Wallinga-Teunis method. Method is exact (and optimal given a grid over R) and deterministic (produces the same answer on the same data).

Methods

This package contains the following methods to solve for $R(t)$ in time:

- filtering

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	No, although some right-censoring is included
Estimates expected cases	No
Communicates uncertainty	Yes
Validation	

Documentation of package methods	Yes
Documentation of package implementation	No

Sample code

The primary sample code comes from this R script.

EpiFusion

REF	Judge et al. (2024)
Docs	
Github	github.com/ciarajudge/EpiFusion
Last commit	Nov, 2024
Installation	

Brief description

Brief summary of EpiFusion method from the paper

EpiFusion is a Bayesian framework designed to estimate the effective reproduction number by jointly analyzing epidemiological (case incidence) and phylodynamic (genomic) data using particle filtering within a particle Markov Chain Monte Carlo (pMCMC) framework. It addresses the limitations of using only epidemiological or genomic data, particularly in under-sampled outbreaks. EpiFusion combines a stochastic infection dynamics model with dual observation models: one for case incidence data and another for phylodynamic tree data. The approach involves sequential particle filtering to simulate infection trajectories, with particles weighted and resampled based on their fit to both data sources. Parameter inference is achieved through Metropolis-Hastings MCMC. EpiFusion has been validated through simulations, benchmarking against existing tools, and application to real-world outbreaks, including the 2014 Ebola outbreak in Sierra Leone.

Methods

This package contains methods that estimate $R(t)$ from both phylodynamic (time-scaled trees estimated from genetic sequences) and epidemiological (case incidence) data. Therefore, a discussion of these methods is somewhat outside the scope of this document.

Assessment

Features	
Ability to nowcast/forecast	No, Designed for retrospective analysis
Incorporates delay distributions	Yes, Handles delays between infection and reporting implicitly
Estimates expected cases	Yes
Communicates uncertainty	Yes, Highest Posterior Density (HPD) intervals
Validation	
Documentation of package methods	Yes
Documentation of package implementation	No

Sample code

Tutorials for how to use EpiFusion are given in this Github repository.

epigrowthfit

REF	Earn et al. (2020)
Docs	None
Github	github.com/davidearn/epigrowthfit
Last commit	Feb, 2025
Installation	CRAN

Brief description

Copied from the developer site.

Maximum likelihood estimation of nonlinear mixed effects models of epidemic growth using Template Model Builder ('TMB'). Enables joint estimation for collections of disease incidence time series, including time series that describe multiple epidemic waves. Supports a set of widely used phenomenological models: exponential, logistic, Richards (generalized logistic), subexponential, and Gompertz. Provides methods for interrogating model objects and several auxiliary functions, including one for computing basic reproduction numbers from fitted values of the initial exponential growth rate. Preliminary versions of this software were applied in Ma et al. (2014) doi:10.1007/s11538-013-9918-2 and in Earn et al. (2020) doi:10.1073/pnas.2004904117

Methods

Given the lack of package documentation it is difficult to fully assess which methods are being implemented.

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	No
Estimates expected cases	No
Communicates uncertainty	No

Validation	
Documentation of package methods	No
Documentation of package implementation	No

Sample code

There is no available vignette or manual on how to use this package other than function descriptions.

This Github repository describes the methods used in Earn et al. (2020), but documentation is minimal.

EpiInvert

REF	Alvarez et al. (2021)
Docs	lalvarezmat.github.io/EpiInvert/
Github	github.com/lalvarezmat/EpiInvert
Last commit	Dec, 2023
Installation	via devtools

Brief description

Brief summary of the method from the paper

EpiInvert is an epidemiological method that estimates the time-varying reproductive number and restores incidence curves by inverting the renewal equation using variational techniques. The approach corrects biases introduced by reporting inconsistencies, including weekly and festive biases, ensuring robust epidemic trend estimation. EpiInvert estimates R(t) by inverting the renewal equation using signal processing techniques, providing a reliable measure of epidemic dynamics. It corrects systematic underreporting due to weekends and holidays by detecting anomalies based on historical trends, redistributing cases across affected days to reduce artificial fluctuations, and adjusting Rt estimates to reflect true transmission patterns. It also includes a forecasting model that predicts epidemic trends using historical trends.

Methods

This package modifies fixed sliding windows, using a variational method that includes solving R(t) using the classic Wallinga Teunis method (Wallinga and Teunis (2004)).

Assessment

Features	
Ability to nowcast/forecast	Yes, Use 'EpiInvertForecast' for forecasting
Incorporates delay distributions	No
Estimates expected cases	Yes
Communicates uncertainty	Yes

Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample code

See this vignette for an example of forecasting, and this vignette for a comparison between EpiInvert and other related packages.

EpiLPS

REF	Gressani et al. (2022)
Docs	
Github	github.com/oswaldogressani/EpiLPS
Last Commit	Oct, 2024
Installation	

Brief description

Brief summary of the method from the paper

EpiLPS is a Bayesian tool for estimating the time-varying reproduction number using a robust, efficient approach. It models case counts with a Negative Binomial distribution to handle overdispersion and employs Bayesian P-splines for smoothing epidemic curves. The methodology leverages Laplace approximations to estimate the posterior distribution of the spline coefficients rapidly. Two inference methods are provided: a fast maximum a posteriori approach for quick estimates and an MCMC scheme using Langevin dynamics for thorough posterior sampling. EpiLPS delivers accurate estimates without arbitrary smoothing assumptions and has been applied to SARS-CoV-1, H1N1, and COVID-19 datasets.

Methods

This package contains the following methods:

-

Assessment

Features	
Ability to nowcast/forecast	Nowcasting,adjusts for underreporting by estimating unreported infections and combining them with reported cases to reflect actual daily epidemics

Incorporates delay distributions	Some, It accounts for the uncertainty associated with reporting delays
Estimates expected cases	Yes
Communicates uncertainty	Yes, The credible intervals are calculated via the delta method
Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Starter code

EpiNow2

REF	Wellcome report
Docs	Docs
Github	Github
Last commit	Feb 25, 2021
Installation	Installation

Brief Description

Copied from the developer site

EpiNow2 estimates the time-varying reproduction number on cases by date of infection (using a similar approach to that implemented in EpiEstim). True infections, treated as latent and unobserved, are estimated and then mapped to observed data (for example cases by date of report) via one or more delay distributions (in the examples in the package documentation these are an incubation period and a reporting delay) and a reporting model that can include weekly periodicity.

Uncertainty is propagated from all inputs into the final parameter estimates, helping to mitigate spurious findings. This is handled internally. The time-varying reproduction estimates and the uncertain generation time also give time-varying estimates of the rate of growth.

Forecasting is also supported for the time-varying reproduction number, infections, and reported cases using the same generative process approach as used for estimation.

Important links:

Methods

This package contains the following methods:

- Gaussian Process
- random walk

Assessment

Features	
Ability to nowcast/forecast	Yes
Incorporates delay distributions	Yes
Estimates expected cases	Yes
Communicates uncertainty	Yes
Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Starter code

epinowcast

REF	Lison et al. (2024)
Docs	epinowcast.org/
Github	github.com/epinowcast
Last commit	Sep 30, 2024
Installation	CRAN

Description

Copied from the developer site

Tools to enable flexible and efficient hierarchical nowcasting of right-truncated epidemiological time-series using a semi-mechanistic Bayesian model with support for a range of reporting and generative processes. Nowcasting, in this context, is gaining situational awareness using currently available observations and the reporting patterns of historical observations. This can be useful when tracking the spread of infectious disease in real-time: without nowcasting, changes in trends can be obfuscated by partial reporting or their detection may be delayed due to the use of simpler methods like truncation. While the package has been designed with epidemiological applications in mind, it could be applied to any set of right-truncated time-series count data.

Methods

This package contains the following methods:

- random walk

Assessment

Features	
Ability to nowcast/forecast	Nowcasting
Incorporates delay distributions	Yes
Estimates expected cases	Yes

Communicates uncertainty	Yes
Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample code

A list of helpful vignettes are given here.

ern

REF	Champredon et al. (2024)
Docs	
Github	github.com/phac-nml-phrsd/ern
Last commit	May 22, 2024
Installation	via devtools

Brief description

The ern package was developed to adapt the EpiEstim package for real world data, including wastewater and clinical data. Specifically the package:

- disaggregates clinical reports into a shorter time unit to enable estimation of R_t using an intrinsic generation interval on a useful timescale;
- provides a framework to estimate R_t from wastewater data, consistent with an estimation based on clinical data;
- provides a user-friendly interface geared at public-health practitioners that may have limited proficiency in the R programming language;
- uses EpiEstim for efficient and rapid estimation.

Methods

This package combines the following methods:

- fixed sliding windows

including an additional module for disaggregation of data into shorter time units.

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	Includes incubation period and reporting delay for clinical data
Estimates expected cases	Doing this from wastewater or aggregated clincal case data
Communicates uncertainty	Uncertainty from both EpiEstim approach, as well as assumptions made in estimating incident cases through resampling approach
Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample code

There is a vignette and sample code and worked examples in the Plos One publication for this method Champredon et al. (2024).

EstimateR

REF	Scire et al. (2023)
Docs	covid-19-re.github.io/estimateR/index.html
Github	https://github.com/covid-19-Re/estimateR
Last commit	Sep 10, 2024
Installation	via devtools

Brief description

EstimateR is a package that is built on the EpiEstim framework for estimating R_t and includes steps to smooth, backcalculate data to infection dates and create confidence intervals for estimates. Specifically, the method takes observed observations of infection events, such as case confirmations, hospital admissions, intensive care unit admissions, or deaths and performs the following four steps:

- Smooth the data to reduce noise in the data.
- Backcalculate data to date of infection.
- Estimate R_t using EpiEstim.
- Calculate 95% confidence intervals using bootstrapping.

Each of these tasks can be done separately and the users is not required to perform all tasks.

There is apparently an option to nowcast data described and implemented in the package, though provided mathematical details are limited.

Methods

This package contains the following methods:

- Data is smoothed using LOESS with a first order polynomial. Users should adapt the smoothing parameter consistent with the noise in the input data.
- Deconvolution with an Expectation-Maximization (EM) algorithm is used to create an estimate of the time series of infection events.
- EpiEstim is used to estimate R_t with a Bayesian framework, which uses fixed sliding windows.

- Block bootstrapping is used to estimate 95% confidence intervals.

Details of the methods used are provided in the supplement of Scire et al. (2023)

Assessment

Features	
Ability to nowcast/forecast	Nowcasting, but details are limited
Incorporates delay distributions	Includes incubation period and reporting delay for clinical data
Estimates expected cases	Doing this from wastewater or aggregated clincal case data
Communicates uncertainty	Uncertainty from both EpiEstim approach, as well as assumptions made in estimating incident cases through resampling approach
Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample code

See here for an example of calculating R(t) from aggregated incidence data, as well as other examples.

R0

REF	Obadia et al. (2012)
Docs	None
Github	https://github.com/tobadia/R0
Last commit	Feb, 2025
Installation	CRAN

Brief description

A package that implements existing methods to estimate R_0 and R_t. The advantage of this package is that it standardizes data formats and the parameterization of the generation interval.

This package was developed in 2012 before many of the current methods were developed and most of the methods that are described in the package are not commonly used.

Methods

This package contains the following methods:

- Function to define the generation interval. Options include empiric (i.e. multinomial), lognormal, gamma, and weibull distributions.
- Estimation of R_0 as a function of the attack rate (user must provide this).
- Method to estimate R_0 from the exponential growth rate described by `Wallinga and Lipsitch`.
- Maximum likelihood based estimate of R_0 and serial interval introduced by `White and Pagano`.
- Sequential Bayesian method to estimate time-varying reproductive number introduced by `Bettencourt and Ribiero`.
- Retrospective estimation of the time-varying reproductive number introduced by `Wallinga and Teunis`.

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	No
Estimates expected cases	No
Communicates uncertainty	Some methods allow for this
Validation	
Documentation of package methods	Yes
Documentation of package implementation	No

Sample code

No vignettes are given, this script gives a very simple example, as does the tests folder.

RtEstim

REF	Liu et al. (2024)
Docs	https://dajmcdon.github.io/rtestim/
Github	
Last commit	
Installation	via devtools

Brief description

Rtestim is a method that uses the renewal equation and a provided serial interval distribution to estimate R_t. Distinct from other methods, it uses a frequentist approach with an L1 smoothing penalty which decreases computation time and allows for locally adaptive estimates. The method estimates confidence bands for R_t and incidence.

RtEstim | | | Sep 25, 2024|

Methods

This package contains the following methods:

- Locally adaptive estimator using Poisson trend filtering
- L1 smoothing
- Cross validation to select tuning parameters for the smoother

This approach is similar to filtering

Assessment

Features	
Ability to nowcast/forecast	No
Incorporates delay distributions	No
Estimates expected cases	Predicts based on estimated R_t
Communicates uncertainty	Yes

Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample code

The package website has this helpful vignette.

WhiteLabRt

REF	Li and White (2021) Zhou et al. (2022)
Docs	None
Github	https://github.com/cmilando/WhiteLabRt
Last commit	Aug 16, 2024
Installation	CRAN

Brief description

This package implements methods described in Li and White (2021) for backcalculation and nowcasting and Zhou et al. (2022) for small area estimation using mobility data. The package uses STAN to improve computational efficiency and stability. All methods are implemented in a Bayesian framework. Currently the package does not allow the user to incorporate both mobility data and do nowcasting and account for reporting delays.

Methods

This package contains the following methods:

Li and White (2021) - Adjustment for reporting delay and nowcasting estimates - this uses the fixed sliding windows appraoch to estimate $R(t)$

Zhou et al. (2022) - Mobility data in a hierarchical model to obtain spatially granular estimates. - this a weekly random walk to estimate $R(t)$

Assessment

Features	
Ability to nowcast/forecast	Nowcasting, not forecasting
Incorporates delay distributions	No, but some functions that calculate a reporting delay distribution from missing line-list data
Estimates expected cases	Yes
Communicates uncertainty	Yes

Validation	
Documentation of package methods	Yes
Documentation of package implementation	Yes

Sample code

See vignettes for back-calculation of missing reporting delay information and spatial R(t) between various regions

Glossary

Effective reproduction number

From Gostic et al. (2020):

> The effective reproductive number, denoted as or R_e or R_t, is the expected number of new infections caused by an infectious individual in a population where some individuals may no longer be susceptible

Also called the instantaneous reproductive number.

Generation interval

The time between the infection date of an individual and the infection date of the person who infected them. This is typically described by a statistical distribution, such as a gamma, lognormal or weibull.

Serial interval

The time between the infection date of an individual and the infection date of the person who infected them. This is typically described by a statistical distribution, such as a gamma, lognormal or weibull.

References

Almutiry W, V VWK, Deardon R. 2021. Continuous Time Individual-Level Models of Infectious Disease: Package EpiILMCT. Journal of Statistical Software 98:1–44; doi:10.18637/jss.v098.i10.

Alvarez L, Colom M, Morel J-D, Morel J-M. 2021. Computing the daily reproduction number of COVID-19 by inverting the renewal equation using a variational technique. Proceedings of the National Academy of Sciences 118:e2105112118; doi:10.1073/pnas.2105112118.

Bettencourt LMA, Ribeiro RM. 2008. Real Time Bayesian Estimation of the Epidemic Potential of Emerging Infectious Diseases. PLOS ONE 3:e2185; doi:10.1371/journal.pone.0002185.

Bokányi E, Vizi Z, Koltai J, Röst G, Karsai M. 2023. Real-time estimation of the effective reproduction number of COVID-19 from behavioral data. Scientific Reports 13:21452; doi:10.1038/s41598-023-46418-z.

Champredon D, Papst I, Yusuf W. 2024. Ern: An R package to estimate the effective reproduction number using clinical and wastewater surveillance data. PLOS ONE 19:e0305550; doi:10.1371/journal.pone.0305550.

Cori A, Ferguson NM, Fraser C, Cauchemez S. 2013. A New Framework and Software to Estimate Time-Varying Reproduction Numbers During Epidemics. American Journal of Epidemiology 178:1505–1512; doi:10.1093/aje/kwt133.

Driessche P van den, Watmough J. 2002. Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. Mathematical Biosciences 180:29–48; doi:10.1016/S0025-5564(02)00108-6.

Earn DJD, Ma J, Poinar H, Dushoff J, Bolker BM. 2020. Acceleration of plague outbreaks in the second pandemic. Proceedings of the National Academy of Sciences 117:27703–27711; doi:10.1073/pnas.2004904117.

Flaxman S, Mishra S, Gandy A, Unwin HJT, Mellan TA, Coupland H, et al. 2020. Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. Nature 584:257–261; doi:10.1038/s41586-020-2405-7.

Fraser C. 2007. Estimating Individual and Household Reproduction Numbers in an Emerging Epidemic. PLOS ONE 2:e758; doi:10.1371/journal.pone.0000758.

Gostic KM, McGough L, Baskerville EB, Abbott S, Joshi K, Tedijanto C, et al. 2020. Practical considerations for measuring the effective reproductive number, Rt. PLOS Computational Biology 16:e1008409; doi:10.1371/journal.pcbi.1008409.

Green WD, Ferguson NM, Cori A. 2022. Inferring the reproduction number using the renewal equation in heterogeneous epidemics. Journal of The Royal Society Interface 19:20210429; doi:10.1098/rsif.2021.0429.

Gressani O, Wallinga J, Althaus CL, Hens N, Faes C. 2022. EpiLPS: A fast and flexible Bayesian tool for estimation of the time-varying reproduction number. PLOS Computational Biology 18:e1010618; doi:10.1371/journal.pcbi.1010618.

Huisman JS, Scire J, Caduff L, Fernandez-Cassi X, Ganesanandamoorthy P, Kull A, et al. 2022. Wastewater-Based Estimation of the Effective Reproductive Number of SARS-CoV-2. Environmental Health Perspectives 130:057011; doi:10.1289/EHP10050.

Judge C, Vaughan T, Russell T, Abbott S, Plessis L, Stadler T, et al. 2024. EpiFusion: Joint inference of the effective reproduction number by integrating phylodynamic and epidemiological modelling with particle filtering. PLOS Computational Biology 20:e1012528; doi:10.1371/journal.pcbi.1012528.

Lehtinen S, Ashcroft P, Bonhoeffer S. 2021. On the relationship between serial interval, infectiousness profile and generation time. Journal of The Royal Society Interface 18:20200756; doi:10.1098/rsif.2020.0756.

Li T, White LF. 2021. Bayesian back-calculation and nowcasting for line list data during the COVID-19 pandemic. PLOS Computational Biology 17:e1009210; doi:10.1371/journal.pcbi.1009210.

Lison A, Abbott S, Huisman J, Stadler T. 2024. Generative bayesian modeling to nowcast the effective reproduction number from line list data with missing symptom onset dates. PLoS Computational Biology 20: e1012021.

Liu J, Cai Z, Gustafson P, McDonald DJ. 2024. Rtestim: Time-varying reproduction number estimation with trend filtering. PLOS Computational Biology 20:e1012324; doi:10.1371/journal.pcbi.1012324.

Ma J. 2020. Estimating epidemic exponential growth rate and basic reproduction number. Infectious Disease Modelling 5:129–141; doi:10.1016/j.idm.2019.12.009.

Ma J, Earn DJD. 2006. Generality of the Final Size Formula for an Epidemic of a Newly Invading Infectious Disease. Bulletin of Mathematical Biology 68:679–702; doi:10.1007/s11538-005-9047-7.

Musa SS, Zhao S, Wang MH, Habib AG, Mustapha UT, He D. 2020. Estimation of exponential growth rate and basic reproduction number of the coronavirus disease 2019 (COVID-19) in Africa. Infectious Diseases of Poverty 9:96; doi:10.1186/s40249-020-00718-y.

Nash RK, Bhatt S, Cori A, Nouvellet P. 2023. Estimating the epidemic reproduction number from temporally aggregated incidence data: A statistical modelling approach and software tool. PLOS Computational Biology 19:e1011439; doi:10.1371/journal.pcbi.1011439.

Obadia T, Haneef R, Boëlle P-Y. 2012. The R0 package: A toolbox to estimate reproduction numbers for epidemic outbreaks. BMC medical informatics and decision making 12: 1–9.

Parag KV. 2021. Improved estimation of time-varying reproduction numbers at low case incidence and between epidemic waves. PLOS Computational Biology 17:e1009347; doi:10.1371/journal.pcbi.1009347.

Parag KV, Donnelly CA. 2020. Using information theory to optimise epidemic models for real-time prediction and estimation. PLoS Computational Biology 16:e1007990; doi:10.1371/journal.pcbi.1007990.

R Core Team. 2022. R-4.1.3.

Rabiner L, Juang B. 1986. An introduction to hidden markov models. ieee assp magazine 3: 4–16.

Riutort-Mayol G, Bürkner P, Andersen M, Solin A, Vehtari A. 2020. Practical hilbert space approximate bayesian gaussian processes for probabilistic programming. ArXiv prepr. arXiv preprint arXiv:200411408.

Schulz E, Speekenbrink M, Krause A. 2018. A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. Journal of mathematical psychology 85: 1–16.

Scire J, Huisman JS, Grosu A, Angst DC, Lison A, Li J, et al. 2023. estimateR: An R package to estimate and monitor the effective reproductive number. BMC Bioinformatics 24:310; doi:10.1186/s12859-023-05428-4.

Walker TM, Ip CL, Harrell RH, Evans JT, Kapatai G, Dedicoat MJ, et al. 2013. Whole-genome sequencing to delineate Mycobacterium tuberculosis outbreaks: A retrospective observational study. The Lancet Infectious Diseases 13:137–146; doi:10.1016/S1473-3099(12)70277-3.

Wallinga J, Lipsitch M. 2006. How generation intervals shape the relationship between growth rates and reproductive numbers. Proceedings of the Royal Society B: Biological Sciences 274:599–604; doi:10.1098/rspb.2006.3754.

Wallinga J, Teunis P. 2004. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. American Journal of epidemiology 160: 509–516.

Yang X, Wang S, Xing Y, Li L, Xu RYD, Friston KJ, et al. 2022. Bayesian data assimilation for estimating instantaneous reproduction numbers during epidemics: Applications to COVID-19. PLoS Computational Biology 18:e1009807; doi:10.1371/journal.pcbi.1009807.

Zhou Z, Kolaczyk ED, Thompson RN, White LF. 2022. Estimation of heterogeneous instantaneous reproduction numbers with application to characterize SARS-CoV-2 transmission in massachusetts counties. PLOS Computational Biology 18: e1010434.