Reproductive Number $R(t)$

Table of contents

Overview

Since the onset of the COVID-19 pandemic in early 2020, there has been a proliferation of software tools that make inference about the current state of an infectious disease outbreak.

A widely used parameter in these tools is the effective reproductive number, $R(t)$, defined in Gostic et al. (2020) as: "... the expected number of new infections caused by an infectious individual in a population where some individuals may no longer be susceptible." $R(t)$ has the following values and interpretations:

R(t)	Interpretation at time t	Outbreak is ...
< 1	Each infected person infects *on average* less 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

Importantly, $R(t)$ *cannot be measured directly*, it can only be estimated from observable variables (like reported case counts). It represents a combination of dynamic processes, including disease characteristics (e.g., infectiousness under various conditions, mode of transport) and extrinsic factors (e.g., lockdowns that reduce person-to-person contact).

The purpose of this document

Each software package that estimates $R(t)$ makes different assumptions, which leads to variations in estimated $R(t)$ *even if the same input data are used.*

Therefore, the purpose of this document is provide guidance about which software packages to choose for different analytical goals: see our Decision Matrix.

Funding, authors, and acknowledgements

This work is supported by InsightNet grant #

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

- Chad Milando, Laura White, and Christine Sangphet

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, …

Decision matrix

You can use the decision tool below to help choose which R packages may be right for your application:

1. Look first for the desired output that you want to produce
2. Then see which set of required data you have. If you do not have the required data for a specific package, use a different one
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.

> ! Strong recommendation
>
> **Use multiple packages** in your analysis; an ensemble of approaches will be the best way to ensure a robust estimate of $R(t)$ for your use case.

Table 2: Decision matrix for choosing R(t) estimation tool

Desired output	Data required	Package options
What will R(t) be next week	Daily reported case counts	EpiNow2, Option2
i.e., forcasting or nowcasting applications	Serial interval	
	Reporting delay distribution	
	Wastewater suriveillance data	no plan yet

What was R(t) in the past week	Daily reported case counts	RtEstim, Option2
	Serial interval	
	Reporting delay distribution	
	Wastewater	no plan yet
What was R(t) historically	Daily reported case counts	EpiEstim, Option2
	Serial interval	
	Reporting delay distribution	
	Wastewater	no plan yet

Assessment framework

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

- 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 CRAN, 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: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Part I

Estimating R(t)

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 also present:

: An Rshiny application for simulation of case data and $R(t)$: A list of open research questions

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,2 behavioral data,3 or viral loads in waste-water4),
- calculations from compartmental, agent-based models, or network.5–7

We also limit the discussion to packages in the statistical software R,8 which may exclude some packages in other software programs that combine many of the methodological considerations discussed below.9

The methods discussed below and references to specific R packages are current as of December 1, 2024. We attempt to harmonize the mathematical choices between each package using terminology from each.

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. The first class of methods assumes there is a formulaic relationship between infections and reproduction number, a relationship known as the renewal equation.10 These infections are then assumed to result in (some fraction of) the observed cases. 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, , 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.11 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 generation interval. In this paper we use the generation interval 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 symptom onset 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)), the formulation of the renewal equation is thus:

I(t)=R(t) _(i=max(1, t-s+1))^t (i) I(t-i) (Eq.1)

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

Λ(t)= _(i=max(1,   t-s+1))^t (i) I(t-i)   (Eq.2)

The assumptions of this formulation, as per Green et. al. 2022,12 are that incident infections can be described deterministically within each window of t [t-s+1,t] and that the generation interval distribution does not change over the modeling time.

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 [t-s+1,t]), infections can be assumed to grow exponentially at a constant rate (r).12–14 Using Eq. 1 in the time window t [t- s+1,t] and assuming some initial infections k, R(t) for t [t- s+1,t] can be inferred from only r and :

I(t)=ke^rt,t [t-s+1,t] (Eq.3)

R(t)=[_(i=max(1, t-s+1))^t   (i) e^(-ri)]^(-1) ,t [t- s+1,t] (Eq.4)

Again, we will omit the writing the bounds for time in remaining formulae. A single R(t) value, say R_0, can also be put in the form of an infection attack rate, z,15 or in the final size equation,16 to estimate the proportion of all individuals that were affected by a disease with this R_0:

z=1-exp(-R_0 z) (Eq.5)

The attack rate function and others are implemented in the package epigrowthfit.17 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).

Using the renewal equation (Eq. 1) and given that I(t) and   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. This is undesirable for several reasons: real-world infectivity likely does not vary dramatically from day to day, and 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. In addition, infection incidence, I(t), are the data of interest but it is impossible to observe, so many calculations instead may 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 renewal equation defines the relationship between infections and R(t), smoothing or regression models calculate time-varying R(t) directly from 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.

EPINOW2 also has a non-renewal equation-based approach https://epiforecasts.io/EpiNow2/articles/estimate_i mechanistic-infection-model

Zac's Gam Rt: https://github.com/CDCgov/cfa-gam-rt A GAM model will estimate this using a hierarchical spline w different components and weights

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), , 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 Cori et. al. (2013) and the R package EpiEstim.18

These simplifying assumptions greatly constrain the space of potential R(t) and thus calculation times are relatively fast. Other software packages, such as EpiNow2,19 do not assume any distributional structure for R(t) or ; this increases model flexibility at the cost of 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) from I(t-1).

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. For time = t-1 the I(t) calculated in the renewal equation provides the expected value for a draw from a discrete distribution, the value of which represents the next generation of infectees. The discrete distribution chosen is commonly a Poisson distribution (in which the mean and variance parameter (t)=I(t)). Thus, using this constraint, the time-series of I(t) represents draws from a series of Poisson distributions with means of (t). Alternatively, a Negative Binomial distribution can be used (with a mean parameter again equal to I(t)), although this requires additionally fitting the size parameter (roughly, the spread of the distribution) to account the infectee distribution being “over-dispersed”.20

Importantly, if additional delay distributions are included in the process of estimating R(t), the parameter that distributions are being used to estimate for the next generation may change (e.g., from I(t) to D(t), the mean value for daily reported cases calculated after applying delay distributions to I(t)).

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. » make a destinction between deriving it and doing it by Bayesian.

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, , 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) days 3 to 7 are drawn from a Gamma distribution with parameters a_2 and b_2, and so on. 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   and R(t), Cori et. al. (2013)18 analytically derived a posterior distribution R(t) using fixed-size sliding windows, which has the following directly calculated (rather than inferred) mean and coefficient of variation of R(t):

E[R(t)]=[a+ _(i=max(1, t-))^t I(i)]/[1/b+ _(i=max(1, t-))^t Λ(i)] (Eq.6) C.V.[R(t)]=[a+ _(i=max(1, t-))^t I(i)]^(-1) (Eq.7)

Thus, sliding windows with larger   improve the stability of the estimate of R(t) over smaller   because the coefficient of variation of R(t) decreases as number of infections increases (see Web Appendix 1 of Cori et. al., 2013).18 Sliding windows are a key feature of EpiEstim.21 There are limitations of this derived sliding window approach, articulated well in Gostic et. al., (2020)1 and summarized here. There is no posterior distribution for the expected value of incidence In the fixed size sliding window approach,   must be explicitly defined prior to inference. Shorter will lead to quicker response but more variable estimates of R(t), which increases the risk of over-fitting. At the extreme, if the is set to 1 day, the resulting R(t) will recover exactly the infection data. 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)1 recommends using the midpoint of each sliding window rather than time t. The choice of both 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 end of the modeling period to account for reporting delays or time between the midpoint of and the end of , and at the beginning of the time period to allow for enough cases to materialize. Web Appendix 4 of Cori. et. al (2013) gives the following recommendation for when to calculate R(t): "Overall, we suggest starting estimating once those three criteria are fulfilled: at least after , at least after one mean serial interval, and when at least 12 cases have been observed since the beginning of the epidemic." The default recommendation for is one week (7 days);18 alternatively the package APEestim integrates with EpiEstim to propose a non-default choice of that minimizes one-step-ahead prediction errors.22

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 epidemia23 and EpiNow2 contain an implementations of a random walk procedure that look generally as follows:

f(R(t))=f(R(t-1))+N(0, _R) (Eq.8) _R ~ HalfNormal(,) (Eq.9)

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 variables and are hyperparameters. The function f can be a transformation of R(t), e.g. in log space as in EpiNow2 to correct for the skewness of R(t), 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. The function f in epidemia contains more layers for pooled effects and group-level variables.

Filtering

Filtering is another way that R(t) is constrained in common packages. Filtering means [...]. One way that a filter could be implemented is in a Hidden Markov Model.24 A simple forwardlooking linear filter for R(t) in an Hidden Markov Model might look as follows, with a tuning parameter ( ) to influence the amount that R(t) can vary between time-steps and a standard white noise component ( ):

R(t)=R(t-1)+(√(R(t-1))) (t-1) (Eq.10)

The package EpiFilter25 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.

```
RtEstim:28 https://dajmcdon.github.io/rtestim/articles/delay-distributions.html

We propose a discrete spline-based approach, RtEstim, that solves a convex optimization probl
Poisson trend filtering-using the proximal Newton method. It produces a locally adaptive esti
EpiLPS:29
```

In EpiFilter, RtEstim, and EpiLPS, each 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 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 models26 are a more flexible method of constraining the evolution of R(t) in time than the methods discussed thus far (in fact, a random-walk process can be thought of as a simplified case of Gaussian Process model). 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^')= ^2 exp[-(x-x^')$^{2/(2l}$2)] (Eq.10)

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 uses contains options to use Gaussian Process models to control how R(t) 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)=logR(t-1)+ GP(0,k(R(t), R(t ^')))        (Eq.11)
```

The advantage of Gaussian Process models is that R(t) is enforced to change smoothly in time using Eq.10. Limitations include complexity and computational time: Gaussian Process models have a computational complexity of O(n^3) for n observations.27 Although EpiNow2 in practice implements faster approximations of Gaussian Process models,27 in general Gaussian Process runtimes and required computational resources are considerable as compared to other methods.

Accumulated Prediction Error (APE) framework

In APEestim, Parag et al. adapt an approximation to MDL known as the accumulated prediction error (APE) to identify the k (window length) best justified by the available epi-curve, k*, to opitmizing the window length k.

We analytically derive the posterior predictive incidence distribution of the renewal model, which allows us to evaluate cumulative log-loss prediction scores at any k exactly. The APE choice, k*, minimises these scores and, additionally, optimises short-term prediction accuracy. This method is valid at all sample sizes, easily computed for arbitrary model dimensions and, unlike many selection criteria of similar computability, includes parametric complexity. Parametric complexity measures how functional relationships among parameters influence complexity, and when ignored (as in the Bayesian or Akaike information criteria) can lead to biased renewal model selection.

The APE metric designates the model that best predicts unseen data from the same underlying process, and not the one that best fits existing data, as optimal and of justified complexity. This is equivalent to minimising what is called the generalisation or out-of-sample error in machine learning, and is known to balance under and overfitting. Our APE-based approach therefore pinpoints and characterises only those Rt fluctuations that are integral to achieving reliable short-term incidence growth predictions.

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

Extending EpiEstim • The package bayEStim30 also extends EpiEstim o 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. • Tenglong's work31 and the accompanying package WhiteLabRt32 use the sliding window approach to estimating missing reporting delay information from line-list data (originally implemented as a Gibbs sampler, later updater to STAN). • The package estimateR involves estimating missing count data using smoothing [confirm].33

- Does EpiNow2 do this?

Epidemia: 23 . We introduce a Bayesian mechanistic model linking the infection cycle to observed deaths, inferring the total population infected (attack rates) as well as Rt.

Delay distributions

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. sometimes R(t) is calculated directly from reported case data and then shifted backwards by a delay distribution, whereas other times R(t) is calculated from inferred dates of infection using reported case data.

Reporting delay, Onset delays etc Delay PMFs that you can pass in series which have cascading impacts.

```
EpiNow2 has this:
Our estimates overcome some of the limitations of naive implementations that derive estimates
Our approach also incorporates multiple sources of uncertainty that if excluded can bias esti
```

```
EpiFilter was also recently generalized to incorporate hetereogeonous transmission rates and
```

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

Clinical data distributions

Again, some packages just modify EpiEstim:

- The ern package ultimately uses the EpiEstim package for the core of the computation as EpiEstim already provides a robust and one of the fastest implementations of well-tested estimation algorithms. However, ern wraps complex and critical features for estimating from real-world clinical and wastewater data that have not all been implemented in any one existing package for estimation

Here, we present the library ern to address the gaps identified above, specifically: o to disaggregate the clinical reports into a shorter time unit to enable estimation of using an intrinsic generation interval on a useful timescale; o to provide a framework to estimate from wastewater data, consistent with an estimation based on clinical data; o to provide a user-friendly interface geared at public-health practitioners that may have limited proficiency in the programming language; o to perform an efficient and rapid estimation.

Linear predictor model components

ViaEpidemia School closures etc

EpiFusion:37 We propose a model of Rt that estimates outbreak trajectories conditional upon both phylodynamic (time-scaled trees estimated from genetic sequences) and epidemiological (case incidence) data.

Inference frameworks

Finally there are different ways of acutally calculating the numbers once you have the theory lined up.

Bayesian optimization

Assumes a distribution –> solved analytically • EpiEstim o restricted set of GI options (gamma?) enables analytical solve for the posterior estimate of R(t) which is also a Gamma, using conjugate priors

Doesn't assume a distribution of R(t) or I(t) –> Uses MCMC • EpiNow2, implemented in STAN • Hierarchical NUTS

MaxLiklihood optimization

- Frequentist o RtEstim

Wallinga, J., and P. Teunis. "Different Epidemic Curves for Severe Acute Respiratory Syndrome Reveal Similar Impacts of Control Measures." American Journal of Epidemiology 160,no. 6 (2004): 509. • ^ this has the likelihood calculation in it

One of the most widely used methods for estimating time-varying reproduction number is a maximum likelihood-based approach {White, 2008}. • White LF, Pagano M. A likelihood-based method for real-time estimation of the serial interval and reproductive number of an epidemic. Stat Med 2008; 27(16): 2999–3016.

Simulation tool

The tool below walks through the steps of simulating estimates of the instantaneous reproduction number,which can be helpful for surveillance and intervention planning of infectious diseases. For this simulation, we take several steps to simulate how cases spread from one person to another: (1) Simulate the individual-level incubation time distribution, then (2) simulate the individual-level transmission time distribution (assumed to be independent from the incubation time distribution). We then can derive distributions for the generation time and serial interval using the relationships, simulate the individual-level administrative delay in reporting, and simulate the population-level infectivity dynamics.

Open research questions

- Need to add a page of existing research questions
  - sub-regional or pooling ...
  - other stuff, i think you had a list of this somehwere

Part II

Packages

APEestim

| Parag and Donnelly (2020) | | Feb 12, 2021 |

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.

Important links:

REF: PLOS Computational Biology (Peer reviewed:)	Github (Updated: Mar 2025)	CRAN (Updated: Feb 2025)

Methods

This package aims to improve upon the limitation of fixed sliding windows. They employ:

- APE Framework

Assessment

Table 5: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

bayEStim

Brief description

bayEStim | Lytras T, Sypsa V, Demosthenes P, Tsiodr S | | Aug 3, 2020 |

Methods

This package contains the following methods:

-

Assessment

Table 6: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

earlyR

Brief description

earlyR | Jombart T, Cori A, Nouvellet P, Skarp, J | | Oct 27, 2020 |

Methods

This package contains the following methods:

-

Assessment

Table 7: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

Epidemia

| Flaxman et al. (2020) | | Jun 23, 2021 |

Brief description

Methods

This package contains the following methods:

-

Assessment

Table 8: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

EpiEstim

EpiEstim | Cori et al. (2013), Nash et al. (2023) | | Aug 30, 2024 |

Brief description

Methods

This package contains the following methods:

- fixed sliding windows

Assessment

Table 9: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

EpiFilter

Brief description

EpiFilter | Parag (2021) | | Dec 9, 2023 |

Methods

This package contains the following methods:

-

Assessment

Table 10: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

EpiFusion

Brief description

EpiFusion | Judge et al. (2024) | | Nov 30, 2024|

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 the following methods:

-

Assessment

Table 11: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no

Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

epigrowthfit

Brief description

epigrowthfit | Earn et al. (2020) | | Aug 12, 2024|

Methods

This package contains the following methods:

-

Assessment

Table 12: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

EpiInvert

Brief description

EpiInvert | Alvarez et al. (2021) | | Dec 31, 2023|

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 Rt 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 contains the following methods:

-

Assessment

Table 13: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no

Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

EpiLPS

Brief description

EpiLPS | Gressani et al. (2022) | | Oct 24, 2024|

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

Table 14: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		

Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Starter code

EpiNow2

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:

REF: Wellcome report (Peer reviewed:)	Docs	Github (Updated: Mar 2025)	CRAN (Updated: Feb 2025)

Methods

This package contains the following methods:

- Gaussian Process

Assessment

Table 16: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Starter code

- copy in from SummRT
- all things should use the same datasets
- could make these shiny apps if we wanted

epinowcast

Description

epinowcast | Abbott S, Lison A, Funk S, Pearson C, Gruson H, Guenther F, DeWitt M | | Sep 30, 2024 |

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:

-

Assessment

Table 17: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no

Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

ern

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.

ern | Champredon et al. (2024) | | May 22, 2024|

Methods

This package contains the following methods:

- EpiEstim for estimation of R_t.
- Disagregation of data into shorter time units, as necessary.

Assessment

Table 18: Assessment rubric

Category	Notes
Usage	
Runtime length	Runs with EpiEstim platform, which is fast and efficient.
Features	
Ability to nowcast/forecast	No documentation of this capability.
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 e

Validation	
Peer reviewed validation	Some description
Replicable documentation	Some description

-

Sample code

There is a lot of sample code and worked examples in the Plos One publication for this method
.

EstimateR

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.

EstimateR | Scire et al. (2023) | | Sep 10, 2024|

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.
- 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

Table 19: Assessment rubric

Category	Notes
Usage	
Runtime length	Comparison of Estimate R with epidemia and EpiNow2 on simulated data.
Features	
Ability to nowcast/forecast	Appears possible; limited details provided.
Incorporates delay distributions	Uses deconvolution.
Estimates expected cases	This is a separate module that is calculated.
Communicates uncertainty	Uses Block boostrapping to create 95% CIs.
Validation	
Peer reviewed validation	Peer reviewed publication.
Replicable documentation	GitHub site and R package.

Sample code

R0

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.

R0 | Obadia et al. (2012) | | Sep 20, 2023|

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

Table 20: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Not described. But methods used are not computationally complex.	N/A

Features		
Ability to nowcast/forecast	Not available in the provided methods.	No
Incorporates delay distributions	Not available in the provided methods	No
Estimates expected cases	Not available in the provided methods	No
Communicates uncertainty	Some methods allow for this.	Yes
Validation		
Peer reviewed validation	Peer reviewed paper published describing the package.	Yes
Replicable documentation	Package is on CRAN.	Yes

Sample code

RtEstim

Brief description

RtEstim | Liu et al. (2024) | | Sep 25, 2024|

Methods

This package contains the following methods:

-

Assessment

Table 21: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

WhiteLabRt

Brief description

WhiteLabRt | Li and White (2021) | | Aug 16, 2024|

Methods

This package contains the following methods:

-

Assessment

Table 22: Assessment rubric

Category	Notes	Scales
Usage		
Runtime length	Some description	Time (minutes)
Features		
Ability to nowcast/forecast	Some description	Yes/no
Incorporates delay distributions	Some description	Yes/no
Estimates expected cases	Some description	Yes/no
Communicates uncertainty	Some description	Yes/no
Validation		
Peer reviewed validation	Some description	Yes/no
Replicable documentation	Some description	Yes/no

Sample code

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.

References

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.