RESEARCH ARTICLE

Fast estimation of time-varying infectious disease transmission rates

Mikael Jagan$^{1,2,*}$, Michelle S. deJonge$^{1,2,}$, Olga Krylova$^{1,3}$, David J. D. Earn$^{1,2,*}$

1 Department of Mathematics & Statistics, McMaster University, Hamilton, Ontario, Canada, 2 M.G. DeGroote Institute for Infectious Disease Research, McMaster University, Hamilton, Ontario, Canada

* These authors contributed equally to this work.

✉ Current address: Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada
✉ Current address: Integrated Decision Support, Hamilton Health Sciences, Hamilton, Ontario, Canada
✉ Current address: Advanced Analytics, Canadian Institute for Health Information, Ottawa, Ontario, Canada
✉ earn@math.mcmaster.ca

Abstract

Compartmental epidemic models have been used extensively to study the historical spread of infectious diseases and to inform strategies for future control. A critical parameter of any such model is the transmission rate. Temporal variation in the transmission rate has a profound influence on disease spread. For this reason, estimation of time-varying transmission rates is an important step in identifying mechanisms that underlie patterns in observed disease incidence and mortality. Here, we present and test fast methods for reconstructing transmission rates from time series of reported incidence. Using simulated data, we quantify the sensitivity of these methods to parameters of the data-generating process and to mis-specification of input parameters by the user. We show that sensitivity to the user’s estimate of the initial number of susceptible individuals—considered to be a major limitation of similar methods—can be eliminated by an efficient, “peak-to-peak” iterative technique, which we propose. The method of transmission rate estimation that we advocate is extremely fast, for even the longest infectious disease time series that exist. It can be used independently or as a fast way to obtain better starting conditions for computationally expensive methods, such as iterated filtering and generalized profiling.

Author summary

Many pathogens cause recurrent epidemics. Patterns of recurrence are strongly affected by seasonality of the transmission rate, which can arise from seasonal changes in weather and host population behaviour (e.g., aggregation of children in schools). To understand and predict recurrent epidemic patterns, it is essential to reconstruct the time-varying transmission rate, which is never observed directly. Existing transmission rate estimation methods tend to fall into one of two categories: accurate but too slow to apply to long time series of reported incidence, or fast but inaccurate unless the number of individuals initially susceptible to infection is known precisely. Here, we introduce and compare fast methods inspired by the algorithm that Fine and Clarkson pioneered in the early 1980s.
The method that we suggest accurately reconstructs seasonally varying transmission rates, even with crude information about the initial size of the susceptible population.

1 Introduction

The transmission rate of an infectious disease is a salient quantity in the study of epidemics. Changes in the transmission rate over time greatly influence the spread of infection [1, 2]. Quantifying how it changes over time can elucidate factors governing disease spread (e.g., weather [3], contact patterns [4]), inform epidemic forecasts, and suggest strategies for epidemic control [5].

In practice, we do not observe transmission directly. Instead, we observe the number of cases of infection (disease incidence) or number of deaths from infection (disease mortality) reported over time, and must reconstruct time-varying transmission rates from these data [6–13]. Utilizing historical mortality records, it is possible to identify patterns in transmission dating far back in time. Most notably, the London Bills of Mortality and the Registrar General’s Weekly Returns enable investigation of transmission patterns continuously from the mid-17th century to the present, for a number of infectious diseases including cholera [14] and smallpox [15].

A mechanistic understanding of long infectious disease time series—three centuries of weekly data in the case of smallpox [15]—requires methods of transmission rate estimation that are both accurate and fast, and therefore tractable for long time scales. Simulation-based methods of transmission rate estimation from reported incidence or mortality have been developed using the susceptible-infected-removed (SIR) model for infectious disease dynamics [16]. Markov chain Monte Carlo (MCMC [17, 18]) and sequential Monte Carlo (as in iterated filtering [8, 19, 20]) methods are statistically rigorous, but not tractable for long time scales owing to high computational cost. Generalized profiling [21, 22], which combines trajectory and gradient matching, is faster, but still too slow for convenient exploration of time series spanning hundreds of years. (Several CPU hours were required to apply generalized profiling to just 26 years of weekly data [22].)

In comparison, Finkenstädt and Grenfell’s popular “time series SIR” (tSIR) method [7, 23] is extremely fast, using a simple discretization of a continuous-time SIR model to reduce transmission rate estimation to a local regression problem. However, the tSIR method assumes that the duration of infection is equal to the time step, that natural death of susceptible individuals can be ignored, and that cumulative incidence approximates cumulative births. The latter two assumptions are reasonable for pre-vaccination measles, when most susceptible individuals were eventually infected [6]. However, in many contexts (e.g., with pathogens less transmissible than measles), susceptible mortality over time scales of interest and the difference between incidence and births are non-negligible.

In their unpublished PhD and MSc theses, Krylova (Ch. 4 in [24]) and deJonge [25] separately modified a fast discretization method originally proposed by Fine and Clarkson [6]. Krylova’s approach has been employed to estimate the amplitude of seasonal variation in measles transmission in 20th century New York City [9]. Unlike the tSIR method and unlike Fine and Clarkson, Krylova’s and deJonge’s methods do not place constraints on the infectious period or ignore susceptible mortality.

Here, we present a new algorithm based on deJonge’s method and compare its performance to the methods of Fine and Clarkson and Krylova. Our main investigative approach is to apply...
each method to simulated reported incidence data with known underlying transmission rate, so that error in transmission rate estimates can be quantified exactly.

Our analysis of the methods reveals a shared sensitivity to process and observation error. We mitigate this issue by introducing a smoothing step. The methods are additionally sensitive to error in the user’s estimate of the initial number of susceptible individuals, which is rarely known with any precision. We propose a fast, iterative technique for estimating this parameter from time series of incidence, births, and natural mortality, eliminating a long-standing barrier to the use of fast methods of transmission rate reconstruction.

2 Methods

In §2.1 and 2.2 below, we present three fast methods for estimating time-varying transmission rates, based on a mechanistic model of disease spread. In §2.3–2.7, we outline our systematic analysis of the sensitivity of the methods to parameters of the data-generating process and to error in the user-specified values of input parameters. Finally, in §2.8, we introduce peak-to-peak iteration (PTPI), a technique for estimating the initial number of susceptible individuals. Essential notation is summarized in Table 1.

2.1 Model of disease transmission

We assume that the principal mechanisms of disease spread in the focal population are captured by the SIR model [16], formulated with time-varying rates of birth, death, and transmission. Expressing the model as a system of ordinary differential equations, we write

\[
\begin{align*}
\frac{dS}{dt} &= \nu(t)\hat{N}_0 - \beta(t)SI - \mu(t)S, \\
\frac{dI}{dt} &= \beta(t)SI - \gamma I - \mu(t)I, \\
\frac{dR}{dt} &= \gamma I - \mu(t)R,
\end{align*}
\]

(1)

where \(S\), \(I\), and \(R\) are the numbers of individuals who are susceptible, infected, and removed, respectively; \(N = S + I + R\) is the population size; and \(\hat{N}_0 = N(0)\) is the population size at an initial time, defined to be 0 years for simplicity. (We reserve the notation \(N_0\) for \(N(t_0)\), where \(t_0 > 0\) years is the length of a transient; see Table 1.)

The time-varying parameters are

- \(\nu(t)\) birth rate, the number of births per unit time relative to \(\hat{N}_0\);
- \(\mu(t)\) natural mortality rate, the number of natural deaths per unit time \textit{per capita} (i.e., relative to \(N\)); and
- \(\beta(t)\) transmission rate, the number of infections per unit time \textit{per susceptible per infected}.

The constant parameter \(\gamma\) is the rate of removal from the infected compartment (due to recovery or death from disease) \textit{per infected individual}.

In Eq (1a) and Eq (1b), we use mass action incidence \(\beta(t)SI\) rather than standard incidence \(\beta(t)SI/N\). Mass action incidence is essential for reproducing transitions in epidemic patterns resulting from changes in the birth rate [2, 28]. In Eq (1a), we write the net birth rate as \(\nu(t)\hat{N}_0\) rather than \(\nu(t)\). This formulation is for convenience: the factor of \(\hat{N}_0\) does not affect dynamics, but ensures that \(\nu(t)\) and \(\mu(t)\) have the same scale.
Table 1. Notation. Unless otherwise stated, simulations of reported incidence time series use the reference values listed here. If a symbol is to be interpreted differently in relation to disease incidence and disease mortality data, then the correct definition is indicated by (I) and (M), respectively.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Definition</th>
<th>Ref. val.</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_k$</td>
<td>$k$th observation time</td>
<td>Time of the $k$th observation in time series data, for $k = 0, \ldots, n$.</td>
<td>$t_0 + k\Delta t$</td>
<td>years</td>
<td></td>
</tr>
<tr>
<td>$t_0$</td>
<td>Transient period</td>
<td>Duration of the transient in system (1) that is ignored in simulations of reported incidence, before observations are recorded.</td>
<td>2000</td>
<td>years</td>
<td>System (1) is numerically integrated between $t = 0$ years and $t = t_0$, and observed starting at $t = t_0$. This is done so that simulations reflect dynamics near the attractor of system (1).</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Observation interval</td>
<td>Time between successive observations in time series data.</td>
<td>1</td>
<td>weeks</td>
<td>Disease mortality is reported weekly in the London Bills of Mortality.</td>
</tr>
<tr>
<td>$n$</td>
<td>Time series length</td>
<td>Time between the initial and final observations in time series data, in units $\Delta t$, given by $(t_n - t_0)/\Delta t$.</td>
<td>1042</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lceil t \rceil_{\Delta t}$</td>
<td>Nearest $k\Delta t$ rounding</td>
<td>For time lengths $t$, $\lceil t \rceil_{\Delta t} = \left\lfloor \frac{t}{\Delta t} \right\rfloor \Delta t$, where $\lfloor \cdot \rceil$ denotes nearest integer rounding.</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$\langle \cdot \rangle$</td>
<td>Long-term averaging</td>
<td>For functions $x(t)$, $\langle x \rangle = \lim \frac{1}{t} \int_0^t x(s) , ds$. For sequences $x_h(s)$, $\langle x_h \rangle = \lim_{n \to \infty} \frac{1}{\sum_{k=0}^{\infty}} x_h$.</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$(S(t), I(t), R(t))$</td>
<td>State</td>
<td>Number of (susceptible, infected, removed) individuals in the population at time $t$.</td>
<td>—</td>
<td>—</td>
<td>“Removed” individuals have either recovered from the disease and gained permanent immunity or died from the disease.</td>
</tr>
<tr>
<td>$N(t)$</td>
<td>Population size</td>
<td>$S(t) + I(t) + R(t)$.</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$B(t)$</td>
<td>Births</td>
<td>Number of births that occur during the time interval $[t - \Delta t, t)$.</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$Q(t)$</td>
<td>Cumulative incidence</td>
<td>Number of susceptibles who become infected during the time interval $[t_0, t)$.</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$Z(t)$</td>
<td>Incidence</td>
<td>Number of susceptibles who become infected during the time interval $[t - \Delta t, t)$, $Q(t) - Q(t - \Delta t)$.</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$C(t)$</td>
<td>Reported disease (I) incidence or (M) mortality</td>
<td>Number of (I) infections or (M) disease-induced deaths reported during the time interval $[t - \Delta t, t)$.</td>
<td>Eq (31)</td>
<td></td>
<td>C is an abbreviation of “cases”, which are reported as infections or as deaths.</td>
</tr>
<tr>
<td>$(S_0, I_0, R_0)$</td>
<td>Initial state $(t = t_0)$</td>
<td>$(S(t_0), I(t_0), R(t_0))$.</td>
<td>$(S', I', R')$</td>
<td>—</td>
<td>$(S', I', R')$ denotes the state of system (1) after numerical integration between $t = 0$ years and $t = t_0$ with seasonally forced transmission rate $\beta(t)$ (Eq (27)), constant vital rates $\nu$ and $\mu$, and initial state $(t = 0)$ $(S, I, R)$ (Eq (32); see below).</td>
</tr>
<tr>
<td>$N_0$</td>
<td>Initial population size $(t = t_0)$</td>
<td>$N(t_0)$.</td>
<td>$S' + I' + R'$</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$(\hat{S}, \hat{I}, \hat{R})$</td>
<td>Endemic equilibrium</td>
<td>Endemic equilibrium of system (1) with constant transmission rate $(\beta \equiv \hat{\beta})$ and constant vital rates $(\nu \equiv \mu \equiv \mu_c)$.</td>
<td>Eq (32)</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>$N_\infty$</td>
<td>Initial population size $(t = 0$ years)</td>
<td>$N(0)$.</td>
<td>$10^6$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_k$</td>
<td>Estimation input/ output</td>
<td>Within an estimation algorithm (Boxes 1–3), the supplied or estimated value of $s(t_k)$ $(x = C, B, \mu, Z, S, I, \beta)$.</td>
<td>—</td>
<td>varies</td>
<td></td>
</tr>
<tr>
<td>$v(t)$</td>
<td>Birth rate</td>
<td>Number of births per unit time relative to $N_\infty$ at time $t$.</td>
<td>$v_c$</td>
<td>year$^{-1}$</td>
<td>In simulations of reported incidence, $v(t)$ is modeled as a constant $v_c$. In general, estimation of $\beta(t)$ from data does not require the underlying $v(t)$ to be constant.</td>
</tr>
<tr>
<td>$\nu_c$</td>
<td>Birth rate (constant)</td>
<td>See $v(t)$ above.</td>
<td>0.04</td>
<td>year$^{-1}$</td>
<td></td>
</tr>
<tr>
<td>$\mu(t)$</td>
<td>Natural mortality rate</td>
<td>Number of natural deaths per unit time per capita, at time $t$.</td>
<td>$\mu_c$</td>
<td>year$^{-1}$</td>
<td>In simulations of reported incidence, $\mu(t)$ is modeled as a constant $\mu_c$. In general, estimation of $\beta(t)$ from data does not require the underlying $\mu(t)$ to be constant.</td>
</tr>
<tr>
<td>$\mu_c$</td>
<td>Natural mortality rate (constant)</td>
<td>See $\mu(t)$ above.</td>
<td>0.04</td>
<td>year$^{-1}$</td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
Table 1. (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Definition</th>
<th>Ref. val.</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{gen}}$</td>
<td>Mean generation interval</td>
<td>Mean time between onset of infection (infector) and subsequent transmission of infection (by infector) [26, 27].</td>
<td>13</td>
<td>days</td>
<td>The reference value is the sum of the observed mean latent and infectious periods, which for measles are 8 days and 5 days, respectively [16].</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Removal rate</td>
<td>Number of removals (recoveries or deaths from disease) per unit time per infected.</td>
<td>$1/N_{\text{gen}}$</td>
<td>day$^{-1}$</td>
<td>— If we simulate data with under-reporting, then we use $\gamma_{\text{rep}} = 0.25$ as a reference value. Otherwise, we set $\gamma_{\text{rep}} = 1$.</td>
</tr>
<tr>
<td>$p_{\text{rep}}$</td>
<td>Case reporting probability</td>
<td>(I) Probability that an infection is reported, or (M) the case fatality ratio times the probability that a death from disease is reported.</td>
<td>0.25 or 1</td>
<td>—</td>
<td>— If we simulate data with under-reporting, then we use $p_{\text{rep}} = 0.25$ as a reference value. Otherwise, we set $p_{\text{rep}} = 1$.</td>
</tr>
<tr>
<td>$t_{\text{rep}}$</td>
<td>Mean case reporting delay</td>
<td>Mean time between infection and reporting of (I) infection or (M) disease-induced death.</td>
<td>2 or 0 weeks</td>
<td>—</td>
<td>— If we simulate data with reporting delays, then we use $t_{\text{rep}} = 2$ weeks as a reference value. Otherwise, we set $t_{\text{rep}} = 0$ weeks.</td>
</tr>
<tr>
<td>$\beta(t)$</td>
<td>Transmission rate</td>
<td>Number of infections per unit time per susceptible per infected, at time $t$.</td>
<td>Eq (27) year$^{-1}$</td>
<td>—</td>
<td>In simulations of reported incidence, $\beta(t)$ is modeled by the seasonal forcing function defined in Eq (27). In general, estimation of $\beta(t)$ from data does not require it vary seasonally or even periodically.</td>
</tr>
<tr>
<td>$\langle \beta \rangle$</td>
<td>Mean transmission rate</td>
<td>Continuous-time average of the seasonally forced $\beta(t)$, equal to $\lim_{t \to \infty} \frac{1}{t} \int_0^t \beta(s) , ds$.</td>
<td>$\beta^\ast$ year$^{-1}$</td>
<td>—</td>
<td>$\beta^\ast \approx 5.6 \times 10^{-3}$ year$^{-1}$ is the value of $\langle \beta \rangle$ that satisfies Eq (2) with $R_0 = 20$, $v_c = \mu_c = 0.04$ year$^{-1}$, $t_{\text{gen}} = \gamma^{-1} = 13$ days, and $N_{\text{inf}} = 10^6$.</td>
</tr>
<tr>
<td>$R_0$</td>
<td>Basic reproduction number</td>
<td>Number of individuals that a typical infected person is expected to infect in an otherwise completely susceptible population.</td>
<td>Eq (2)</td>
<td>—</td>
<td>— For measles in the 20th century, $R_0 \approx 20$ [16].</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Seasonal amplitude</td>
<td>Amplitude of the seasonally forced $\beta(t)$ relative to $\langle \beta \rangle$.</td>
<td>0.08</td>
<td>—</td>
<td>— For measles, $\alpha \approx 0.08$ [2]. We require $\alpha \in [0, 1]$ to ensure that the seasonal forcing function defined in Eq (27) is non-negative.</td>
</tr>
<tr>
<td>$\phi(t; \epsilon)$</td>
<td>Environmental noise (realized)</td>
<td>Phase shift in the seasonally forced $\beta(t)$, at time $t$.</td>
<td>Normal $(0, \epsilon^2)$</td>
<td>—</td>
<td>$\phi$ is a realization of a continuous-time stochastic process defined by a set ${\Phi(t; \epsilon)}$ of independent and Normal$(0, \epsilon^2)$-distributed random variables.</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Standard deviation of environmental noise</td>
<td>See $\phi(t; \epsilon)$ above.</td>
<td>0.5</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>$q$</td>
<td>Loess smoothing parameter</td>
<td>Rough number of nearest neighbours weighted in local regression (i.e., when fitting loess curves to time series), determining the degree of smoothing.</td>
<td>—</td>
<td>—</td>
<td>— See §2.2.6 for an exact definition.</td>
</tr>
</tbody>
</table>

https://doi.org/10.1371/journal.pcbi.1008124.t001

The SIR model (1) assumes that the focal population is homogeneously mixed and subject to the mass action principle, which states that incidence is proportional to the product of the densities of susceptibles and infecteds [16]. The model further assumes that the latent period (time from infection to onset of infectiousness) can be ignored and that the infectious period (time from onset of infectiousness to recovery or death from disease) is exponentially distributed [29]. The distributions of the latent and infectious periods affect disease dynamics [28, 30, 31], but Krylova and Earn [28] showed that the effect on long-term dynamical structure is typically small when the mean generation interval is fixed (see Fig 1 in [28]). For this reason, we assign the mean generation interval implied by the SIR model (1) ($t_{\text{gen}} = \gamma^{-1}$) the value of the sum of the observed mean latent and infectious periods. This sum is the true mean generation interval if the latent and infectious periods are both exponentially distributed, and is a good estimate of the true mean generation interval more generally [28].

Transmissibility of infection is typically measured by the basic reproduction number $R_0$, defined as the number of individuals that a typical infected person is expected to infect in an otherwise completely susceptible population [16]. If the birth and death rates are constant ($\nu = v_c$ and $\mu = \mu_c$), and if the transmission rate has a well-defined average $\langle \beta \rangle$ [32], then the
basic reproduction number for the SIR model (1) can be written as [28]

$$R_0 = \frac{vN_0}{\mu_c} \cdot \frac{\langle \beta \rangle}{\gamma + \mu_c}.$$  \hspace{1cm} (2)

### 2.2 Estimating $\beta(t)$ from time series data

Here, we examine three fast methods for estimating time-varying transmission rates $\beta(t)$. The methods take as input (i) time series of reported disease incidence or disease mortality, (ii) time series of births and natural mortality, and (iii) values for input parameters, such as the mean generation interval $t_{gen}$. By assumption, the time series data are available at discrete, equally spaced time points

$$t_k = t_0 + k\Delta t, \quad k = 0, \ldots, n,$$  \hspace{1cm} (3)

where $\Delta t$ is the observation interval. The methods return as output a time series estimate of $\beta(t)$, denoted by $\beta_k$ which can be averaged (§2.2.5) or smoothed (§2.2.6) to distill temporal patterns of interest.

Missing data must be imputed: the three methods are recursive, so they break down as soon as they encounter a missing value. Imputation can be accomplished most simply via linear interpolation between available data. More sophisticated techniques accounting for uncertainty in missing values are described in [33].

#### 2.2.1 FC method

We review the method first described by Fine and Clarkson [6], referred to here as the "FC method". Let $S(t)$ and $I(t)$ be the number of susceptibles and infecteds in the population at time $t$. $S$ decreases when susceptibles become infected or die and increases when susceptibles are born. Let $Z(t)$ and $B(t)$ be the number of infections and births, respectively, that occur during the time interval $[t - \Delta t, t)$. Assuming that natural mortality was negligible, Fine and Clarkson reconstructed $S$ from $Z$ and $B$ with the recursion

$$S(t + \Delta t) \approx S(t) + B(t + \Delta t) - Z(t + \Delta t).$$  \hspace{1cm} (4)

Fine and Clarkson further assumed that the observation interval $\Delta t$ was equal to the mean generation interval $t_{gen}$, so that prevalence could be approximated by incidence. That is,

$$I(t) \approx Z(t)$$  \hspace{1cm} (5)

for all $t$. They derived an expression for $Z(t + \Delta t)$ via the mass action principle

$$Z(t + \Delta t) \approx \beta(t)S(t)I(t)\Delta t.$$  \hspace{1cm} (6)

Rearranging Eq (6), they obtained an estimate of $\beta(t)$, given by

$$\beta(t) \approx \frac{Z(t + \Delta t)}{S(t)I(t)\Delta t}.$$  \hspace{1cm} (7)

Fine and Clarkson applied Eqs (4), (5), and (7) to estimate $S(t_k)$, $I(t_k)$, and $\beta(t_k)$ for $k = 0, \ldots, n$, after specifying (i) the initial number of susceptibles $S_0 = S(t_0)$, and (ii) values of $Z(t_k)$ and $B(t_k)$ from incidence and birth data, respectively.

A limitation of the FC method is the constraint requiring $\Delta t = t_{gen}$. For some diseases, this is a minor issue, because incidence and birth data can be aggregated so that the time between successive aggregates is approximately equal to $t_{gen}$. For example, the mean generation interval of measles is approximately two weeks, so Fine and Clarkson [6] aggregated pairs of weekly
observations. A second, more serious limitation is the assumption, implicit in Eqs (4) and (5), that natural mortality is negligible. We discuss this issue in §3.1.

2.2.2 S method. Krylova (Ch. 4 in [24]) generalized the FC method in order to eliminate the constraint requiring \( \Delta t = t_{\text{gen}} \) and account for natural mortality. Her approach is based on the SEIR model, which distinguishes “exposed” individuals in the latent stage of infection from infectious individuals. Here, we adapt Krylova’s approach to the SIR model (1) and refer to our approach as the “S method.”

We define \( S, I, Z, \) and \( B \) as in the FC method. Let \( \mu(t) \) be the per capita natural mortality rate at time \( t \), and let \( Q(t) \) be the total number of infections occurring between the initial observation time \( t_0 \) and current time \( t \) (i.e., cumulative incidence). The observation interval \( \Delta t \) is no longer constrained to be equal to the mean generation interval \( t_{\text{gen}} \).

We reconstruct \( S \) recursively by discretizing Eq (1a):

\[
S(t + \Delta t) \approx S(t) + B(t + \Delta t) - Z(t + \Delta t) - \mu(t)S(t)\Delta t .
\]

Eq (8) is the result of applying the forward Euler method for numerical integration to Eq (1a), and replacing the incidence and birth terms with \( Z(t + \Delta t) \) and \( B(t + \Delta t) \), respectively. Eq (8) is identical to Eq (4) of the FC method, except that it includes a natural mortality term.

In order to estimate \( \beta(t) \), we note that, by definition, \( \frac{dQ}{dt} \) is the rate at which individuals enter the infected compartment. From Eq (1b), this is

\[
\frac{dQ}{dt} = \beta(t)S(t)I(t) .
\]

If the mean generation interval \( t_{\text{gen}} \) is short enough that \( I \) and \( \mu \) are roughly constant between times \( t \) and \( t + t_{\text{gen}} \), then \( \frac{dI}{dt} \approx 0 \) in that interval, and using Eq (1b) we can write

\[
\beta(t)S(t)I(t) \approx (\gamma + \mu(t))I(t) \approx (\gamma + \mu(t + t_{\text{gen}}))I(t + t_{\text{gen}}) .
\]

In this case, \( \frac{dQ}{dt} \) is also (approximately) the rate at which individuals leave the infected compartment, \( t_{\text{gen}} \) time after infection:

\[
\frac{dQ}{dt} \approx (\gamma + \mu(t + t_{\text{gen}}))I(t + t_{\text{gen}}) .
\]

Note that Eq (11) is also valid if the generation interval is narrowly distributed around its mean \( t_{\text{gen}} \) (even if \( t_{\text{gen}} \) is long).

Discretizing Eqs (9) and (11) using forward Euler, we obtain two approximations of \( Z(t + \Delta t) \):

\[
Z(t + \Delta t) = Q(t + \Delta t) - Q(t) \approx \left\{ \begin{array}{ll}
\beta(t)S(t)I(t)\Delta t & \text{from Eq (9),} \\
(\gamma + \mu(t + t_{\text{gen}}))I(t + t_{\text{gen}})\Delta t & \text{from Eq (11).}
\end{array} \right.
\]

Rearranging Eq (12) yields an estimate of \( \beta(t) \), given by

\[
\beta(t) \approx \frac{Z(t + \Delta t)}{S(t)I(t)\Delta t} ,
\]

and an estimate of \( I(t) \), given by

\[
I(t) \approx \frac{Z(t + \Delta t - t_{\text{gen}})}{(\gamma + \mu(t))\Delta t} .
\]

Since data are available only at the observation times \( t_k \) (Eq (3)), the value of \( Z(t + \Delta t - t_{\text{gen}}) \) in
Eq (14) will be observed only if \( t_{\text{gen}} \) is an integer multiple of \( \Delta t \). In general, \( t_{\text{gen}} \) is not divisible by \( \Delta t \). Therefore, in practice, we replace \( t_{\text{gen}} \) in \( Z(t + \Delta t - t_{\text{gen}}) \) with the nearest integer multiple of \( \Delta t \), denoted here by \( [t_{\text{gen}}]_\Delta t \):

\[
I(t) \approx \frac{Z(t + \Delta t - t_{\text{gen}})}{(\gamma + \mu(t))\Delta t}.
\] (15)

Thus, the S method is defined by Eq (13), coupled with Eqs (8) and (15) for the reconstruction of \( S \) and \( I \). The S method requires users to specify (i) input parameters \( S_0 = S(t_0) \) and \( t_{\text{gen}} = \gamma^{-1} \), and (ii) values of \( Z(t_k) \), \( B(t_k) \), and \( \mu(t_k) \) from incidence, birth, and natural mortality data, respectively.

The FC method is a special case of the S method, obtained by setting \( \Delta t = t_{\text{gen}} \) and \( \mu(t) \equiv 0 \).

**2.2.3 SI method.** DeJonge [25] improved Krylova’s method (Ch. 4 in [24]) by reconstructing \( I \) directly from Eq (1b) instead of relying on the approximation in Eq (11). Here, we improve deJonge’s discretization, which employs the forward Euler method, by instead combining forward and backward Euler. One way to do this is to use the trapezoidal method: whereas forward and backward Euler take \( f’(t)\Delta t \) and \( f’(t + \Delta t)\Delta t \), respectively, to approximate integrals \( \int_{t}^{t+\Delta t} f’(\tau) \, d\tau \), the trapezoidal method takes the average \( [\frac{1}{2}f’(t) + f’(t + \Delta t)]\Delta t \), which is less prone to error. Our discretization, which we call the “SI method”, is consistently more accurate than deJonge’s and others (see §S9 of S1 Text for a comparison of nine possible algorithms).

Numerically integrating Eq (1a) and Eq (1b) using the trapezoidal method, and replacing the incidence and birth terms with \( Z(t + \Delta t) \) and \( B(t + \Delta t) \), respectively, we obtain

\[
S(t + \Delta t) \approx \frac{[1 - \frac{1}{2}\mu(t)\Delta t] S(t) + B(t + \Delta t) - Z(t + \Delta t)}{1 + \frac{1}{2}\mu(t + \Delta t)\Delta t}.
\] (16)

and

\[
I(t + \Delta t) \approx \frac{[1 - \frac{1}{2}(\gamma + \mu(t))\Delta t] I(t) + Z(t + \Delta t)}{1 + \frac{1}{2}(\gamma + \mu(t + \Delta t))\Delta t}.
\] (17)

Eq (17) eliminates an important problem with Eq (15) of the S method, which estimates \( I(t) \approx 0 \) if \( Z(t + \Delta t - [t_{\text{gen}}]_\Delta t) = 0 \), leading to division by zero in Eq (13).

Discretizing Eq (9) using forward and backward Euler, we obtain two approximations of \( Z(t + \Delta t) \):

\[
Z(t + \Delta t) = Q(t + \Delta t) - Q(t) \approx \begin{cases} 
\beta(t)S(t)I(t)\Delta t & \text{from forward Euler,} \\
\beta(t + \Delta t)S(t + \Delta t)I(t + \Delta t)\Delta t & \text{from backward Euler.}
\end{cases}
\] (18)

Rearranging Eq (18) yields two estimates of \( \beta(t) \),

\[
\beta(t) \approx \begin{cases} 
\frac{Z(t + \Delta t)}{S(t)I(t)\Delta t} & \text{from forward Euler,} \\
\frac{Z(t + \Delta t)}{S(t + \Delta t)I(t + \Delta t)\Delta t} & \text{from backward Euler,}
\end{cases}
\] (19)

whose average supplies a more accurate estimate (see §S9 of S1 Text), given by

\[
\beta(t) \approx \frac{Z(t) + Z(t + \Delta t)}{2S(t)I(t)\Delta t}.
\] (20)

Thus, the SI method is defined by Eq (20), coupled with Eqs (16) and (17) for the reconstruction of \( S \) and \( I \). Compared to the S method, the SI method, in principle, requires one
additional input parameter, namely the initial number of infecteds \( I_0 = I(t_0) \). In §3.6, we show that, in practice, this additional information is not necessary.

2.2.4 Estimating true incidence from reported incidence. Let \( C(t) \) be the number of infections reported during the time interval \([t − Δt, t]\). We estimate true incidence \( Z \) from reported incidence \( C \) via

\[
Z(t) \approx \frac{1}{p_{\text{rep}}} C(t + [t_{\text{rep}}]_{Δt}) ,
\]

where \( p_{\text{rep}} \) is the probability that an infection is reported and \([t_{\text{rep}}]_{Δt}\) is the mean time between infection and reporting, rounded to the nearest integer multiple of the observation interval \( Δt \).

Eq (21) has the limitation that multiplying by \( p_{\text{rep}}^{-1} \) does not correct for under-reporting if, by chance, \( C(t + [t_{\text{rep}}]_{Δt}) = 0 \). In this situation, not only is the result \( Z(t) \approx 0 \) incorrect, but we divide by zero in the FC and S methods when we substitute Eqs (5) and (15) in Eqs (7) and (13), respectively. If \( C(t + [t_{\text{rep}}]_{Δt}) = C(t + [t_{\text{rep}}]_{Δt} + Δt) = 0 \), then the SI method also suffers: Eq (20) gives \( β(t) \approx 0 \). To circumvent these issues, we replace zeros in reported incidence time series by linearly interpolating between nonzero values prior to estimating true incidence using Eq (21). We do not replace leading and trailing zeros.

If what we observe is deaths from disease, rather than infections, then we have the complication that only a fraction of infections end in death. In this situation, we can still use Eq (21) to estimate \( Z \), provided we interpret (i) \( C \) as reported disease mortality, (ii) \( p_{\text{rep}} \) as the case fatality ratio times the probability that a death from disease is reported, and (iii) \( t_{\text{rep}} \) as the mean time between infection and reporting of disease-induced death.

A more sophisticated method of inferring true incidence from reported data is described in [34].

2.2.5 Averaging raw estimates of \( β(t) \). Given fixed time series data and input parameters, the FC, S, and SI methods return estimates of \( β(t) \) that are entirely determined (not random). In the absence of additional data observed from the same population, it is difficult to assign confidence to the output.

However, if an estimate \( \hat{β}(t) \) is approximately periodic (with apparent period \( T \)) and contains \( m \) complete cycles, and if we assume \( β(t) \) is truly periodic, then we can view \( \hat{β}(t) \) as containing a sample of \( m \) estimates of the true cycle, with some variance, and use its mean as an estimator instead of any one of the \( m \) cycles. For such an estimate \( \hat{β}(t) \) defined on the interval \([t_0, t_0 + mT]\), the mean and variance are given by

\[
\bar{\hat{x}}(t) = \frac{1}{m} \sum_{i=0}^{m-1} \hat{β}(t + iT) , \quad t \in [t_0, t_0 + T) ,
\]

\[
s^2(t) = \frac{1}{m-1} \sum_{i=0}^{m-1} [\hat{β}(t + iT) − \bar{\hat{x}}(t)]^2 , \quad t \in [t_0, t_0 + T) .
\]

In §3.3, we apply the S and SI methods to simulated data to estimate an underlying, seasonally forced \( β(t) \) (Eq (27)), which has a period of 1 year. We linearly interpolate the raw time series estimate \( \hat{β}_s \) and compute the average 1-year cycle in the interpolant \( \hat{β}_\text{int}(t) \) using Eq (22a).

Comparing this average to the true 1-year cycle, we are able to assess bias in the two methods.

Note that \( \bar{\hat{x}}(t) \) and \( s^2(t) \) can be used to obtain a formal, likelihood-based measure of confidence in estimates \( \hat{β}(t) \) (see §2.3.4 in [35]).
2.2.6 Smoothing raw estimates of $\beta(t)$. Process and observation error introduce random fluctuations in reported incidence on top of longer-term (e.g., seasonal) variation. In §3.2, we show that noise in reported incidence is spuriously amplified in $\beta_k$, the raw time series estimate of $\beta(t)$.

To distill temporal patterns of interest from the noise, we fit a smooth loess (short for local regression; see Ch. 8.1 in [36]) curve $\beta_{\text{loess}}(t; q)$ to the points $\{(t_i, \beta_i)\}_{i=0}^{n}$, and use $\beta_{\text{loess}}(t; q)$ as our final estimate of $\beta(t)$. Here, $q \in \{5, \ldots, n+1\}$ is an integer-valued parameter controlling the degree of smoothing. At times $t \in [t_0, t_n]$, the fitted value $\beta_{\text{loess}}(t; q)$ is obtained as follows:

1. Order the distances $d_k = |t_k - t|$ of the time points $t_k$ (Eq (3)) from $t$, letting $d_k$ denote the $k$th smallest distance (for $i = 1, \ldots, n+1$).

2. Fit a quadratic polynomial $p_2(t)$ to the points $\{(t_k, \beta_k)\}_{k=0}^{n}$. This is done by weighted least squares using tricube weights

$$w_k = \begin{cases} 
1 - \left(\frac{d_k}{d_q}\right)^3 & \text{if } 0 \leq d_k < d_q, \\
0 & \text{if } d_k \geq d_q.
\end{cases}$$

(23)

Hence only time points $t_k$ nearer to $t$ than the $q$th nearest time point are weighted in the fit.

3. Define $\beta_{\text{loess}}(t; q) = p_2(t)$.

Typically, smoother fits are obtained with greater $q$ [36, 37].

The optimal value of $q$ for a given time series $\beta_k$, denoted by $q_{\text{opt}}$, is that which minimizes error in $\beta_{\text{loess}}(t; q)$ relative to $\beta(t)$. In §3.4, we estimate $\beta(t)$ from simulated data, smooth $\beta_k$ using each value of $q$ on a grid, and use our knowledge of $\beta(t)$ to determine $q_{\text{opt}}$. We show that it is possible for smoothing to eliminate much of the error in $\beta_k$ attributable to process and observation error. Thus, in §2.2.7, we explicitly define the FC, S, and SI methods with loess smoothing as a final step.

In practice, $\beta(t)$ is not known, so we cannot determine $q_{\text{opt}}$. In this case, $q_{\text{opt}}$ can be estimated using statistical methods, such as time series cross-validation [38]. However, reasonable results can be obtained much more quickly by inspecting $\beta_{\text{loess}}(t; q)$ directly and increasing $q$ from 4 until a desirable degree of smoothing is achieved (e.g., until noise on the time scale of weeks is visibly reduced, and patterns on the time scale of months are easier to discern).

2.2.7 Summary. In Boxes 1–3 below, we summarize the three methods derived in §§2.2.1–2.2.6 for estimating time-varying transmission rates $\beta(t)$ from time series data with observation times $t_k$ (Eq (3)). We use the notation $\chi_k$ to refer to the value supplied or computed for $x(t_k)$ within the estimation algorithms ($x = C, B, \mu, Z, S, I, \beta$).

In Box 4, we provide instructions for input specification based on our analysis of the methods.

---

Box 1. FC method (Fine & Clarkson 1982 [6])

$$Z_k \leftarrow \frac{1}{P_{rep}} C_{k+r}, \quad \text{where } r = \frac{[t_{rep}]_\Delta t}{\Delta t},$$

(24a)

$$S_k \leftarrow S_{k-1} + B_k - Z_k,$$

(24b)
\[ I_k \leftarrow Z_k, \quad (24c) \]

\[ \beta_k \leftarrow \frac{Z_{k+1}}{S_k I_k \Delta t}, \quad (24d) \]

where \( \Delta t \) is assumed to be roughly equal to \( t_{\text{gen}} \), and natural mortality is assumed to be negligible. Users must specify:

- a time series \( \{(t_k, C_k)\}_{k=0}^{n} \) of reported incidence or reported disease mortality, with zeros replaced via linear interpolation between nonzero values;
- a time series \( \{(t_k, B_k)\}_{k=0}^{n} \) of births;
- input parameters \( S_0, t_{\text{gen}}, p_{\text{rep}}, \) and \( t_{\text{rep}} \).

### Box 2. S method (adapted from Krylova 2011 [24])

\[ Z_k \leftarrow \frac{1}{p_{\text{rep}}} C_{k+r}, \quad \text{where } r = \frac{[t_{\text{rep}}]_{\Delta t}}{\Delta t}, \quad (25a) \]

\[ S_k \leftarrow S_{k-1} + B_k - Z_k - \mu_{k-1} S_{k-1} \Delta t, \quad (25b) \]

\[ I_k \leftarrow \frac{Z_{k+1-g}}{(\gamma + \mu_k) \Delta t}, \quad \text{where } g = [t_{\text{gen}}]_{\Delta t} \Delta t, \quad (25c) \]

\[ \beta_k \leftarrow \frac{Z_{k+1}}{S_k I_k \Delta t}, \quad (25d) \]

\[ \beta_{\text{loess}}(t; q) \leftarrow \text{loess curve fit to time series } \{(t_k, \beta_k)\}_{k=0}^{n}. \quad (25e) \]

Users must specify:

- a time series \( \{(t_k, C_k)\}_{k=0}^{n} \) of reported incidence or reported disease mortality, with zeros replaced via linear interpolation between nonzero values;
- a time series \( \{(t_k, B_k)\}_{k=0}^{n} \) of births;
- a time series \( \{(t_k, \mu_k)\}_{k=0}^{n} \) of the \emph{per capita} natural mortality rate;
- input parameters \( S_0, t_{\text{gen}} = \gamma^{-1}, p_{\text{rep}}, \) and \( t_{\text{rep}} \);
- loess smoothing parameter \( q \).
Box 3. SI method (adapted from deJonge 2014 [25])

\[
Z_k \leftarrow \frac{1}{p_{\text{rep}}} C_{k+r}, \quad \text{where } r = \frac{[t_{\text{rep}}] \Delta t}{\Delta t}, \tag{26a}
\]

\[
S_k \leftarrow \frac{[1 - \frac{1}{2} \mu_{k-1} \Delta t] S_{k-1} + B_k - Z_k}{1 + \frac{1}{2} \mu_k \Delta t}, \tag{26b}
\]

\[
I_k \leftarrow \frac{[1 - \frac{1}{2} (\gamma + \mu_{k-1}) \Delta t] I_{k-1} + Z_k}{1 + \frac{1}{2} (\gamma + \mu_k) \Delta t}, \tag{26c}
\]

\[
\beta_k \leftarrow \frac{Z_k + Z_{k+1}}{2 S_k I_k \Delta t}, \tag{26d}
\]

\[
\beta_{\text{loess}}(t; q) \leftarrow \text{loess curve fit to time series } \{(t_k, \beta_k)\}_{k=0}^n. \tag{26e}
\]

Users must specify:

- a time series \(\{(t_k, C_k)\}_{k=0}^n\) of reported incidence or reported disease mortality, with zeros replaced via linear interpolation between nonzero values;
- a time series \(\{(t_k, B_k)\}_{k=0}^n\) of births;
- a time series \(\{(t_k, \mu_k)\}_{k=0}^n\) of the per capita natural mortality rate;
- input parameters \(S_0, I_0, t_{\text{gen}} = \gamma^{-1}, p_{\text{rep}}, \text{ and } t_{\text{rep}};
- \text{loess smoothing parameter } q.

Box 4. Instructions for input specification

- \(\beta_k\) is sensitive to mis-specification of \(S_0\), but not \(I_0\) (cf. §3.6.1). If the user’s estimate of \(S_0\) is uncertain, and if the incidence time series \(Z_k\) is roughly periodic, then a more accurate estimate of \(S_0\) may be obtained via peak-to-peak iteration (PTPI; cf. §3.7).

- If \(S_k\) is negative for any \(k\), then it is likely that the case reporting probability \(p_{\text{rep}}\) was underestimated or that births were systematically under-reported by \(B_k\). This can be resolved by correcting the estimate of \(p_{\text{rep}}\) or correcting \(B_k\), then restarting the algorithm. Users should apply close to the minimal correction necessary to prevent negative \(S_k\).

- \(q\) must be tuned to the \(\beta_k\) time series. An estimate of \(q_{\text{opt}}\) can be obtained using statistical methods, such as time series cross-validation [38]. However, \(q\) can be tuned quickly through visual inspection of \(\beta_{\text{loess}}(t; q)\): one can increase \(q\) from 5 until a desirable degree of smoothing is achieved (e.g., until noise on the time scale of weeks is visibly reduced, and patterns on the time scale of months are easier to discern).
2.3 Simulating reported incidence data

In order to compare the performance of the FC, S, and SI methods in estimating $\beta(t)$, we apply the methods to simulated reported incidence data with known underlying $\beta(t)$. Here, we outline our methods for simulating these data using the SIR model (1).

2.3.1 Seasonal forcing of $\beta(t)$ with environmental stochasticity. We reproduce seasonal fluctuation in the transmission rate by modeling $\beta(t)$ in Eq (1) as a sinusoidal forcing function with period equal to one year:

$$\beta(t) = \langle \beta \rangle \left(1 + \alpha \cos \left(\frac{2\pi t}{\text{year}}\right)\right).$$

Here, $\alpha \in [0, 1]$ is the amplitude of seasonal forcing relative to the mean $\langle \beta \rangle$. We introduce stochastic fluctuation by adding a randomly generated phase shift:

$$\beta\phi(t) = \langle \beta \rangle \left(1 + \alpha \cos \left(\frac{2\pi t}{\text{year}} + \phi(t; \epsilon)\right)\right).$$

$\phi$ is a realization of a continuous-time stochastic process consisting of independent, Normal(0, $\epsilon^2$)-distributed random variables. It models environmental stochasticity leading to random noise in the transmission rate. Modeling environmental stochasticity with a random phase shift rather than additive noise conveniently avoids negative $\beta\phi(t)$:

$\beta\phi(t)$ oscillates between $\langle \beta \rangle(1 - \alpha)$ and $\langle \beta \rangle(1 + \alpha)$ regardless of the distribution of the noise. In practice, we take the values of $\phi$ at times $t_k$ (Eq (3)) and linearly interpolate to obtain values in between. This helps to make simulations of Eqs (1) and (9) with adaptive time steps (cf. §2.3.2) reproducible.

2.3.2 Generating incidence time series with demographic stochasticity. We supplement Eq (1) with Eq (9), so that trajectories of the resulting system record changes in cumulative incidence $Q$. In this system, we employ the noisy transmission rate $\beta\phi(t)$ (Eq (28)) and constant vital rates $\nu_c$ and $\mu_c$. We then either (i) numerically integrate the differential equations to approximate their solution, or (ii) treat the system more realistically as an event-driven, continuous-time Markov process (with event rates specified by terms in the differential equations) and use the adaptive tau-leaping algorithm for stochastic simulation [39, 40]. The latter approach accounts for demographic stochasticity in disease dynamics. We prevent disease fadeout in simulations with demographic stochasticity by setting the rates of infected recovery and death to zero whenever $I = 1$.

In both methods of simulation, we record the state of the system at times $t_k$ (Eq (3)), choosing initial state

$$\begin{pmatrix}
S(t_0) \\
I(t_0) \\
R(t_0) \\
Q(t_0)
\end{pmatrix} = \begin{pmatrix}
S_0 \\
I_0 \\
R_0 \\
0
\end{pmatrix},$$

where $S_0 + I_0 + R_0 = N_0 = N(t_0)$. Finally, we derive incidence $Z$ from $Q$ via first differences:

$$Z(t) = Q(t) - Q(t - \Delta t).$$

2.3.3 Introducing observation error. Observation error due to under-reporting ($p_{rep} < 1$) and reporting delays ($t_{rep} > 0$ weeks) creates discrepancies between true incidence $Z$ and reported incidence $C$. We introduce random observation error to simulated incidence time
series with delayed binomial sampling:

\[ C(t + [t_{\text{rep}}]) \sim \text{Binomial}(Z(t), p_{\text{rep}}). \]  

(31)

For simulations without observation error, we set \(p_{\text{rep}} < 1\) and \(t_{\text{rep}} > 0\) weeks.

2.3.4 Parametrization. The simulation method outlined in §§2.3.1–2.3.3 is parametrized by

\[
\begin{align*}
\text{disease parameters} & : \langle \beta \rangle, \quad \alpha, \quad \epsilon, \quad t_{\text{gen}}; \\
\text{population parameters} & : \tilde{N}_0 = N(0), \quad S_0 = S(t_0), \quad I_0 = I(t_0), \quad v_c, \quad \mu_c; \quad \text{and} \\
\text{reporting parameters} & : p_{\text{rep}}, \quad t_{\text{rep}}, \quad t_0, \quad \Delta t, \quad n.
\end{align*}
\]

For most simulations, we assign parameters the reference values listed in Table 1. We consider different values when we investigate the sensitivity of \(\hat{\beta}(t)\) estimates to data-generating parameters (cf. §2.6.1).

We bypass transient dynamics by choosing \(t_0 = 2000\) years and numerically integrating system (1) between 0 years and \(t_0\) in order to obtain a point \((S', I', R')\) very near the attractor. For this step, we exclude environmental noise, defining \(\beta(t)\) as in Eq (27), and take the initial state to be the endemic equilibrium of the unforced system (system (1) with \(\beta \equiv \langle \beta \rangle\) and \(v \equiv \mu \equiv \mu_c\)):

\[
\begin{pmatrix}
S(0) \\
I(0) \\
R(0)
\end{pmatrix} =
\begin{pmatrix}
\dot{S} \\
\dot{I} \\
\dot{R}
\end{pmatrix} =
\begin{pmatrix}
\frac{\dot{N}_0}{N_0} \\
\frac{\dot{I}_0}{I_0} \\
\frac{\dot{R}_0}{R_0}
\end{pmatrix}
\left(\tilde{N}_0 \left(1 - \frac{1}{N_0}\right) \left(\frac{\dot{N}_0}{N_0} \right)\right) .
\]

(32)

2.4 Creating mock birth and natural mortality time series

In addition to reported incidence data, the FC, S, and SI methods require time series of births and the \textit{per capita} natural mortality rate. For simplicity, we create mock time series by (i) choosing constant vital rates \(v'_c\) and \(\mu'_c\), then (ii) setting \(B_k = v'_c N_0 \Delta t\) and \(\mu_k = \mu'_c\) for all \(k\). Note that \(v'_c N_0 \Delta t\) is the result of integrating the net birth rate in the SIR model (1), given by \(v(t)N_0\), between successive observation times using \(v \equiv v'_c\).

We specify \(v'_c = v_c\) and \(\mu'_c = \mu_c\), where \(v_c\) and \(\mu_c\) are the data-generating vital rates (cf. §2.3.4), except when we investigate the sensitivity of \(\hat{\beta}(t)\) estimates to incorrect vital data (cf. §2.6.2). For example, to model under-reporting of births, we simply set \(v'_c < v_c\).

2.5 Measuring \(\hat{\beta}(t)\) estimation error

When we simulate reported incidence data, the underlying transmission rate \(\beta(t)\) is defined beforehand via Eq (27) and known for all \(t\). We use this knowledge to quantify the error in estimates of \(\beta(t)\) obtained from the data. Specifically, given an estimate \(\hat{\beta}(t)\) defined at time points \(t_k\) (Eq (3)), we compute the relative root mean square error (RMMSE), defined as

\[
\text{RRMSE} (\beta, \hat{\beta}) := \sqrt{\frac{1}{n + 1} \sum_{k=0}^{n} \left( \frac{\beta(t_k) - \hat{\beta}(t_k)}{\beta} \right)^2} ,
\]

(33)
where 

$$\tilde{\beta} := \frac{1}{n + 1} \sum_{k=0}^{n} \beta(t_k).$$

(34)

Note that by “underlying transmission rate” we mean the transmission rate excluding environmental noise. Although we simulate data using the noisy $\beta_n(t)$, defined in Eq (28), our aim is to reconstruct the noiseless $\beta(t)$, defined in Eq (27).

### 2.6 Sensitivity analysis

Error in $\beta(t)$ estimation from reported incidence data depends on how the data were generated. The number of cases reported over time is influenced by features of the disease (e.g., the natural history of infection), population (e.g., contact patterns), and case reporting (e.g., the frequency and accuracy of reports). In our simulations of reported incidence, there are 14 data-generating parameters (cf. §2.3.4), whose values are summarized in the vector

$$\theta = (\langle \beta \rangle, \alpha, \epsilon, \tilde{N}_0, S_0, I_0, v_c, \mu_c, t_{\text{gen}}, p_{\text{rep}}, t_{\text{rep}}, t_0, \Delta t, n).$$

(35)

Estimation error also depends on how accurately certain data-generating parameters are specified by users of the FC, S, and SI methods. The initial observation time $t_0$, observation interval $\Delta t$, and time series length $n$ are always known exactly. Other parameters ($\langle \beta \rangle, \alpha, \epsilon, \tilde{N}_0, v_c, \mu_c$) influence our simulations of reported incidence, but in practice are not parameters of the FC, S, and SI methods. In practice, users are required to specify only $S_0, t_{\text{gen}}, p_{\text{rep}}, t_{\text{rep}}$, and (with the SI method) $I_0$. However, when we test the methods here, we do specify vital rates $v_c$ and $\mu_c$ in order to create mock (constant) birth and natural mortality time series (cf. §2.4). The specified values of these 7 input parameters are summarized in the vector

$$\theta' = (S_0', v_0', v_c', \mu_c', t_{\text{gen}}', p_{\text{rep}}', t_{\text{rep}}').$$

(36)

First, we investigate the sensitivity of the methods to the data-generating parameter values $\theta$. Then, we examine their sensitivity to error in the user’s specification $\theta'$ of the input parameters. Here, we describe our analysis using the notation $\hat{\beta}(t; \theta, \theta')$ to refer to transmission rate estimates constructed with user input $\theta$, from data generated by parameter values $\theta$.

#### 2.6.1 Sensitivity to data-generating parameters

In §3.5, we consider the ideal situation in which the input $\theta'$ corresponds exactly to the data-generating $\theta$. In this case, how sensitive is error in $\hat{\beta}(t; \theta, \theta')$ to $\theta'$? For example, is $\beta(t)$ estimated more accurately for diseases with longer mean generation interval $t_{\text{gen}}$, etc.? To answer these questions, we perform the following steps on a grid of data-generating parameter values $\theta$:

1. Simulate 1000 reported incidence time series using $\theta$.
2. Create corresponding mock (constant) birth and natural mortality time series (cf. §2.4), specifying $v_c' = v_c$ and $\mu_c' = \mu_c$ in the input $\theta'$.
3. Estimate $\beta(t)$ from the simulated data, specifying $S_0 = S_0, I_0 = I_0, t_{\text{gen}}' = t_{\text{gen}}, p_{\text{rep}}' = p_{\text{rep}}$, and $t_{\text{rep}}' = t_{\text{rep}}$ in the input $\theta'$.
4. Compute the median RRMSE in the estimates $\hat{\beta}(t; \theta, \theta')$ (1000 estimates corresponding to 1000 simulations).

We repeat this analysis 6 times, corresponding to 2 methods of $\beta(t)$ estimation (S or SI) and 3 methods of data simulation:
without demographic stochasticity and without observation error (fixing \( p_{\text{rep}} = 1, t_{\text{rep}} = 0 \) weeks),

• with demographic stochasticity but without observation error (fixing \( p_{\text{rep}} = 1, t_{\text{rep}} = 0 \) weeks), or

• with demographic stochasticity and with observation error (fixing \( p_{\text{rep}} = 0.25 \) unless sensitivity to \( p_{\text{rep}} \) is considered, \( t_{\text{rep}} = 2 \) weeks).

Environmental stochasticity (\( \epsilon = 0.5 \)) is included in all simulations.

### 2.6.2 Sensitivity to mis-specification of input parameters.

In §3.6, we fix the data-generating \( \theta \) and consider the more realistic situation in which components of the input \( \theta' \) differ from the corresponding components of \( \theta \) by a potentially large factor. In this case, how sensitive is error in \( \tilde{\beta}(t; \theta, \theta') \) to error in \( \theta' \)? For example, how important is having an accurate estimate of \( t_{\text{gen}}, \text{etc.} \)? To answer these questions, we perform the following steps:

1. Simulate 1000 reported incidence time series using fixed data-generating parameter values \( \theta \). (We assign the reference values listed in Table 1.)

2. For each point on a grid of input parameter values \( \theta' \):
   a. Create mock (constant) birth and natural mortality time series, taking \( \nu_c' \) and \( \mu_c' \) from the input \( \theta' \).
   b. Estimate \( \beta(t) \) from the simulated data, taking \( S_0, I_0, t_{\text{gen}}, p_{\text{rep}}, \) and \( t_{\text{rep}} \) from the input \( \theta' \).
   c. Compute the median RRMSE in the estimates \( \tilde{\beta}(t; \theta, \theta') \) (1000 estimates corresponding to 1000 simulations).

We repeat this analysis 6 times, as outlined at the end of §2.6.1.

### 2.7 Asymptotic analysis

Here, we examine analytically the propagation of input error to the output of the SI method. (Similar expressions for propagated errors are obtained by analyzing the S method.) Our analysis here supports numerical results presented in §3.6 concerning the sensitivity of \( \beta(t) \) estimation error to mis-specification of input parameters.

#### 2.7.1 Explicit solutions of the (\( S_k, I_k \)) difference equations.

The SI method uses Eq (26a) to Eq (26c) to recursively reconstruct \( S(t) \) and \( I(t) \) from time series of reported incidence, births, and natural mortality. After substitution of Eq (26a), Eq (26b) and Eq (26c) can be written as

\[
S_{k+1} = \frac{1 - \frac{1}{2} \mu_k \Delta t}{1 + \frac{1}{2} \mu_k \Delta t} S_k + \frac{B_{k+1} - \frac{1}{2} \rho_{\text{rep}} C_{k+1+r}}{1 + \frac{1}{2} \mu_{k+1} \Delta t}, \quad k = 0, 1, \ldots, (37a)
\]

\[
I_{k+1} = \frac{1 - \frac{1}{2}(\gamma + \mu_k) \Delta t}{1 + \frac{1}{2}(\gamma + \mu_k) \Delta t} I_k + \frac{\rho_{\text{rep}} C_{k+1+r}}{1 + \frac{1}{2}(\gamma + \mu_{k+1}) \Delta t}, \quad k = 0, 1, \ldots, (37b)
\]

where \( r = [t_{\text{rep}}] \Delta t \) is the mean case reporting delay in units of the observation interval, rounded to the nearest integer. Eq (37) are linear, first order difference equations, whose explicit solutions are obtained using standard algebraic techniques (see Eq 1.2.4 in [41]) and
given by

\[
S_k = \sum_{j=0}^{k-1} \prod_{i=j}^{k-1} \left( 1 - \frac{1}{2} \mu_i \Delta t \right) + \sum_{i=0}^{k-1} \left( B_i + \frac{1}{2} \mu_{i+1} \Delta t \right) \prod_{i=j+1}^{k-1} \left( 1 + \frac{1}{2} \mu_i \Delta t \right), \quad k = 0, 1, \ldots, (38a)
\]

\[
I_k = I_0 \prod_{j=0}^{k-1} \left( 1 - \frac{1}{2} (\gamma_j + \mu_j) \Delta t \right) + \sum_{j=0}^{k-1} \frac{C_{j+1,e}}{1 + \frac{1}{2} (\gamma_j + \mu_j) \Delta t} \prod_{i=j+1}^{k-1} \left( 1 + \frac{1}{2} (\gamma_i + \mu_i) \Delta t \right), \quad k = 0, 1, \ldots, (38b)
\]

with the conventions \( \sum_{i=b}^{a} x_i = 0 \) and \( \prod_{i=b}^{a} x_i = 1 \) if \( a < b \). As we show in \S 2.7.2, explicit solutions of Eq (37) facilitate asymptotic analysis.

2.7.2 Propagation of input error to \((S_k, I_k)\). We consider the special case in which the vital rates are constant and set \( B_k = v^f N_0 \Delta t \) and \( \mu_k = \mu_c \) for all \( k \) (cf. \S 2.4). Then Eq (38) simplify to

\[
S_k(S_0, v_c, \mu_c, p_{rep}) = S_0 \left( 1 - \frac{1}{2} \mu_c \Delta t \right)^k + \sum_{i=0}^{k-1} v^f N_0 \Delta t - \frac{1}{p_{rep}} C_{i+1,e} \left( 1 - \frac{1}{2} \mu_c \Delta t \right)^{k-1-i} \quad (39a)
\]

\[
I_k(I_0, \mu_c, f_{gen}, p_{rep}) = I_0 \left( 1 - \frac{1}{2} (\gamma_c + \mu_c) \Delta t \right)^k + \sum_{i=0}^{k-1} \frac{1}{p_{rep}} C_{i+1,e} \left( 1 - \frac{1}{2} (\gamma_c + \mu_c) \Delta t \right)^{k-1-i} \quad (39b)
\]

where we have made explicit the dependence of \( S_k \) and \( I_k \) on input parameters \( S_0, I_0, v_c, \mu_c, f_{gen} = \gamma^{-1} \), and \( p_{rep} \). Using Eq (39), we can derive exact expressions for the error propagated to \( S_k \) and \( I_k \) in the SI method as a result of assigning an incorrect value to an input parameter.

If the initial number of susceptibles is truly \( S_0 \) but we specify \( S_0 = \omega S_0 \), where \( \omega > 0 \), then the error propagated to \( S_0 \) is

\[
\text{Err}(S_k, S_0 \rightarrow \omega S_0) = S_k(\omega S_0, v_c, \mu_c, p_{rep}) - S_k(S_0, v_c, \mu_c, p_{rep}) = (\omega - 1) S_0 \left( 1 - \frac{1}{2} \mu_c \Delta t \right)^k \quad (40)
\]

\[
= (\omega - 1) S_0 \left( 1 - \frac{\Delta t}{t_{life}} \right)^k \rightarrow 0, \quad k \rightarrow \infty,
\]

where \( t_{life} = \mu_c^{-1} \) is the life expectancy in the population. Similarly, specifying \( I_0' = \omega I_0 \) for \( I_0 \) yields an error.
in $I_k$, where $t_{inf} = (\gamma + \mu_c)^{-1}$ is the mean time between infection and removal from the infected compartment, accounting for the possibility of natural death during infection. Eqs (40) and (41) show that the errors propagated to $S_k$ and $I_k$ vanish as $k \to \infty$; we exploit this fact to improve susceptible reconstruction (cf. §2.8).

Mis-specifying $\nu_c$ by assigning a value $\nu'_c = \omega \nu_c$ creates an error in $S_k$ that increases in magnitude over time and converges to a limit:

$$\text{Err}(S_k, \nu_c \leftrightarrow \omega \nu_c) = S_k(S_0, \omega \nu_c, \mu_c, p_{\text{rep}}) - S_k(S_0, \nu_c, \mu_c, p_{\text{rep}})$$

$$= \sum_{i=0}^{k-1} \frac{(\omega - 1)\nu_c \tilde{N}_0 \Delta t}{1 + \frac{1}{2} \mu_c \Delta t} \left(1 - \frac{1}{2} \frac{\mu_c \Delta t}{1 + \frac{1}{2} \mu_c \Delta t}\right)^i$$

$$= \frac{(\omega - 1)\nu_c \tilde{N}_0 \Delta t}{1 + \frac{1}{2} \mu_c \Delta t} \sum_{i=0}^{k-1} \left(1 - \frac{1}{2} \frac{\mu_c \Delta t}{1 + \frac{1}{2} \mu_c \Delta t}\right)^i$$

$$= \frac{(\omega - 1)\nu_c \tilde{N}_0}{\mu_c} \left[1 - \left(1 - \frac{1}{2} \frac{\mu_c \Delta t}{1 + \frac{1}{2} \mu_c \Delta t}\right)^k\right] \quad \xrightarrow{k \to \infty} \quad (\omega - 1)\nu_c \tilde{N}_0 t_{\text{life}}.$$

Unlike Eq (42), the exact expression for $\text{Err}(S_k, \mu_c \leftrightarrow \omega \mu_c)$ is not readily simplified and is difficult to interpret:

$$\text{Err}(S_k, \mu_c \leftrightarrow \omega \mu_c)$$

$$= S_k(S_0, \nu_c, \omega \mu_c, p_{\text{rep}}) - S_k(S_0, \nu_c, \mu_c, p_{\text{rep}})$$

$$= S_0 \left(1 - \frac{1}{2} \frac{\omega \mu_c \Delta t}{1 + \frac{1}{2} \omega \mu_c \Delta t}\right)^k + \sum_{i=0}^{k-1} \frac{\nu_c \tilde{N}_0 \Delta t - \frac{1}{2} \omega \mu_c \Delta t}{1 + \frac{1}{2} \omega \mu_c \Delta t} \left(1 - \frac{1}{2} \frac{\omega \mu_c \Delta t}{1 + \frac{1}{2} \omega \mu_c \Delta t}\right)^{k-1-i}$$

$$- S_0 \left(1 - \frac{1}{2} \frac{\omega \mu_c \Delta t}{1 + \frac{1}{2} \omega \mu_c \Delta t}\right)^k - \sum_{i=0}^{k-1} \frac{\nu_c \tilde{N}_0 \Delta t - \frac{1}{2} \omega \mu_c \Delta t}{1 + \frac{1}{2} \mu_c \Delta t} \left(1 - \frac{1}{2} \frac{\mu_c \Delta t}{1 + \frac{1}{2} \mu_c \Delta t}\right)^{k-1-i}.$$

However, if $C_k$ has a well-defined long-term average $\langle C \rangle$ (this will be true if, for instance, $C_k$ is periodic), then $\text{Err}(S_k, \mu_c \leftrightarrow \omega \mu_c)$ has a well-defined long-term average $\langle \text{Err}(S_k, \mu_c \leftrightarrow \omega \mu_c) \rangle$ with a simple form. Replacing $C_{i+1,x}$ in Eq (43) with $\langle C \rangle$, simplifying the resulting expression,
then taking the limit as \( k \to \infty \), we obtain

\[
\langle \text{Err}(\hat{S}_k, \mu_c \leftarrow \omega \mu_c) \rangle = \lim_{k \to \infty} \left\{ \sum_e \left[ S_e \left(1 - \frac{1}{2} \omega \mu_c \Delta t \right)^k + \frac{v_e \hat{N}_0 \Delta t - \frac{1}{\rho_{\text{rep}}} \langle C \rangle}{\omega \mu_c \Delta t} \left[ 1 - \left( 1 - \frac{1}{2} \omega \mu_c \Delta t \right)^k \right] \right] - \sum_e \left[ S_e \left(1 - \frac{1}{2} \omega \mu_c \Delta t \right)^k - \frac{v_e \hat{N}_0 \Delta t - \frac{1}{\rho_{\text{rep}}} \langle C \rangle}{\mu_c \Delta t} \left[ 1 - \left( 1 - \frac{1}{2} \omega \mu_c \Delta t \right)^k \right] \right] \right\}^{k=1}
\]

\[
= \left( \frac{1}{\omega} - 1 \right) \left( v_e \hat{N}_0 \Delta t - \frac{1}{\rho_{\text{rep}}} \langle C \rangle \right) \frac{\tau_{\text{life}}}{\Delta t},
\]

We can similarly show the following, still assuming that \( \langle C \rangle \) is well-defined:

\[
\langle \text{Err}(I_k, \mu_c \leftarrow \omega \mu_c) \rangle = \frac{\langle C \rangle (t'_{\text{int}} - t_{\text{int}})}{\rho_{\text{rep}} \Delta t}, \quad t'_{\text{int}} = (\gamma + \omega \mu_c)^{-1},
\]

\[
\langle \text{Err}(I_k, t'_{\text{gen}} \leftarrow \omega t_{\text{gen}}) \rangle = \frac{\langle C \rangle (t'_{\text{gen}} - t_{\text{gen}})}{\rho_{\text{rep}} \Delta t}, \quad t'_{\text{gen}} = \left( \frac{\gamma}{\omega} \right)^{-1},
\]

\[
\langle \text{Err}(S_k, \rho_{\text{rep}} \leftarrow \omega \rho_{\text{rep}}) \rangle = \left( 1 - \frac{1}{\omega} \right) \frac{\langle C \rangle \tau_{\text{life}}}{\rho_{\text{rep}} \Delta t},
\]

\[
\langle \text{Err}(I_k, \rho_{\text{rep}} \leftarrow \omega \rho_{\text{rep}}) \rangle = \left( \frac{1}{\omega} - 1 \right) \frac{\langle C \rangle \tau_{\text{life}}}{\rho_{\text{rep}} \Delta t}.
\]

Here, \( t'_{\text{int}} \) is the (incorrect) mean time spent infected that results when \( \omega \mu_c \) is incorrectly specified for \( \mu_c \) (Eq (45)) or \( \omega t_{\text{gen}} \) is incorrectly specified for \( t_{\text{gen}} \) (Eq (46)).

**2.7.3 Propagation of error in \( (S_k, I_k) \) to \( \beta_k \).** Let \( \beta_k(Z_k, Z_{k+1}, S_k, I_k) \) be the raw SI method estimate of \( \beta(t_k) \), given by the right hand side of Eq (26d). Suppose that, due to propagated error (cf. §2.7.2), the arguments are incorrect by a factor, so that

\[
Z_k = \omega_2 Z(t_k), \quad Z_{k+1} = \omega_2 Z(t_{k+1}), \quad S_k = \omega_2 S(t_k), \quad I_k = \omega_2 I(t_k),
\]

where \( \omega_2, \omega_2, \omega_1 > 0 \). Then the computed \( \beta_k \) will have relative error

\[
\frac{\beta_k(Z_k, Z_{k+1}, S_k, I_k) - \beta_k(Z(t_k), Z(t_{k+1}), S(t_k), I(t_k))}{\beta_k(Z(t_k), Z(t_{k+1}), S(t_k), I(t_k))} = \frac{\omega_2}{\omega_1} - 1.
\]

Hence severe underestimation of \( S_k \) or \( I_k \) \( \omega_2 \ll 1 \) or \( \omega_1 \ll 1 \) causes the relative error in \( \beta_k \) to blow up.

**2.8 Estimating \( S_0 \) via peak-to-peak iteration**

Reconstruction of susceptibles \( S(t) \) is a necessary step in the reconstruction of \( \beta(t) \) using the FC, S, and SI methods. In §3.6, we show that susceptible reconstruction requires accurate specification of the initial number of susceptibles \( S_0 = S(t_0) \). However, reliable estimates of \( S_0 \) have, to this point, been difficult to obtain in practice.

We propose a technique for iteratively improving estimates of \( S_0 \), requiring only incidence, birth, and natural mortality data at times \( t_k \) (Eq (3)). Crucially, our technique, which we call
"peak-to-peak iteration" (PTPI), enables accurate susceptible reconstruction without direct observation of the susceptible population size at the initial time.

Our approach is motivated by application of the SI method to simulated data. When we incorrectly guessed the value of $S_0$ and attempted to reconstruct $S(t)$ via Eq (26b), the absolute error in the reconstruction $\{(t_k, S_k)\}_{k=0}^n$ decreased monotonically over time ($k$). (Eq (40) shows that the error propagated from $S_0$ to $S_n$ vanishes as $k \to \infty$.) Consequently, if the underlying dynamics are at least approximately periodic, and if $t_0$ and $t_n$ occur at the same phase of the cycle, then $S_n$ is actually a better estimate of $S_0$ than our initial guess. In this situation, instead of reconstructing $\beta(t)$ directly, we can use $S_n$ as an updated estimate of $S_0$, and reconstruct $S(t)$ more accurately. This procedure can be repeated any number of times, and, with simulated data, we observe rapid convergence to an accurate estimate of $S_0$ (cf. §3.7).

The key point is that the reconstructed final state can be used as an improved estimate of the initial state only if the initial and final states occur at the same phase of the cycle. This will not be true unless the observation period (the time between the first and last observations in time series data) is an integer multiple of the period of the underlying dynamics. We can ensure this by choosing appropriate times at which to start and stop $S(t)$ reconstruction. In noisy periodic data, the points in a cycle that are easiest to identify robustly are the peaks. Consequently, we ignore observations (i) prior to the time $t_0$ of the first peak in the incidence time series and (ii) after the time $t_n$ of the last peak that occurs near an integer multiple of the apparent period after the first peak. For the truncated time series, the iterations converge to an accurate estimate of $S(t_n)$ starting from an initial guess, and we recover the corresponding accurate estimate of $S_0$ by solving Eq (26b) backwards in time, from $t_n$ to $t_0$:

$$S_{k-1} \approx \left[1 + \frac{1}{2} \mu_1 \Delta t\right] S_k - B_k + Z_k \frac{1}{1 - \frac{1}{2} \mu_{k-1} \Delta t}.$$  (51)

The complete PTPI algorithm, which consists of finding $t_a$ and $t_b$ (truncation step) and estimating $S_0$ (iteration step), is outlined in Boxes 5 and 6 below. In §3.7, we assess the performance of PTPI by applying the technique to simulated data with known underlying $S_0$, starting from an incorrect initial estimate of $S_0$.

**Box 5. Peak-to-peak iteration: Truncation step**

**Goal:** Given a roughly periodic time series $\{(t_k, Z_k)\}_{k=1}^n$ of incidence, we want to find the time $t_a$ of the first peak and the time $t_b$ of the last peak occurring at the same phase of the cycle. These times are necessary for the iteration step (Box 6).

**Algorithm:**

i. Smooth the raw incidence time series $Z_k$ by applying a $(2\ell_1 + 1)$-point central moving average, computed via

$$\bar{Z}_k = \frac{1}{2\ell_1 + 1} \sum_{i=-\ell_1}^{\ell_1} Z_{k+i}, \quad k = \ell_1, \ldots, n - \ell_1.$$  (52)

Choose minimal $\ell_1$ large enough to remove spurious peaks in $Z_k$ caused by noise, while retaining true peaks.

ii. Identify the period $T$ of the smoothed incidence time series $\{(t_k, \bar{Z}_k)\}_{k=1}^{n-\ell_1}$ from its power spectrum, and calculate the number of embedded cycles, given by $m = \left\lfloor \frac{n-\ell_1}{T} \right\rfloor$. 


iii. Construct the set $\mathcal{I}$ indexing peaks in $\{(t_i, \tilde{Z}_k)\}_{k=t_i}^{n-t_i}$:

$$\mathcal{I} = \{k \in \{\ell_1 + \ell_2, \ldots, n - \ell_1 - \ell_2\} : \tilde{Z}_k > \tilde{Z}_{k+i} \text{ for all } i = 1, \ldots, \ell_2 \}. \quad (53)$$

Choose minimal $\ell_2$ large enough to ensure that $\mathcal{I}$ indexes true peaks in $\tilde{Z}_k$, but not spurious peaks caused by noise (any that remain after smoothing).

iv. Define $T = \{t_k : k \in \mathcal{I}\}$, the set of times of peaks in $\tilde{Z}_k$, and record the time of the first peak, given by $t_a = \min(T)$.

v. For $i = 0, \ldots, m$, define $\tau_i = t_a + iT$ and find the element of $T$ nearest $\tau_i$, namely $\arg \min_{t \in T} |t - \tau|$. The resulting subset $T_{\text{phase}}$ should contain successive time points that are roughly one period apart, i.e., the corresponding peaks in $\tilde{Z}_k$ should occur at the same phase of the cycle.

vi. Record the time of the last such peak, given by $t_b = \max(T_{\text{phase}})$.

---

Box 6. Peak-to-peak iteration: Iteration step

**Goal:** We want to produce an accurate estimate of the initial number of susceptibles $S_0 = S(t_0)$, given

- a roughly periodic time series $\{(t_i, Z_k)\}_{k=0}^{n}$ of incidence,
- a time series $\{(t_i, B_k)\}_{k=0}^{n}$ of births,
- a time series $\{(t_i, \mu_k)\}_{k=0}^{n}$ of the per capita natural mortality rate,
- times $t_a$ and $t_b$ as defined in the truncation step (Box 5), and
- an initial estimate of $S_0$.

**Algorithm:**

i. Define an initial estimate of $S(t_a)$. (We use the initial estimate of $S_0$)

ii. Reconstruct $S(t)$ between times $t_a$ and $t_b$ using Eq (26b), starting with the current estimate of $S(t_a)$.

iii. Update the estimate of $S(t_a)$ with the estimate of $S(t_b)$ obtained in (ii).

iv. Repeat (ii) and (iii) until the sequence of estimates of $S(t_a)$ converges (to within a desirable tolerance).

v. Reconstruct $S(t)$ between times $t_0$ and $t_a$ using Eq (51), starting with the final estimate of $S(t_a)$ obtained in (iv). The reconstruction is performed backwards in time, from $t_a$ to $t_0$.

vi. Record the estimate of $S_0 = S(t_0)$ computed in (v). This value can be passed back to Eq (26b), allowing for reconstruction of $S(t)$ between times $t_0$ and $t_m$, as usual.
3 Results

In §3.1, we compare the performance of the FC, S, and SI methods in estimating $\beta(t)$ from an idealized reported incidence time series. In §3.2, we show how process and observation error create spurious noise in estimates of $\beta(t)$. In §§3.3 and 3.4, we examine averaging and smoothing as ways to distill temporal patterns of interest from noisy estimates of $\beta(t)$. In §§3.5 and 3.6, we summarize our systematic analysis of the sensitivity of $\beta(t)$ estimation error to data-generating parameters and to mis-specification of input parameters by the user. In §3.7, addressing apparent sensitivity to mis-specification of the initial number of susceptibles $S_0$, we assess the performance of PTPI as a method of estimating $S_0$. Finally, in §3.8, we report the run times of the S and SI methods and PTPI.

The results reported here are entirely reproducible using the annotated R code available in SI File.

3.1 Example of $\beta(t)$ estimation using the FC, S, and SI methods

We applied the FC, S, and SI methods without input error to four reported incidence time series, simulated without process or observation error. The time series estimates $S_k$ and $\beta_k$ are shown in Fig 1. The S and SI methods estimated $S(t)$ and $\beta(t)$ accurately at every time point, whereas the FC method captured seasonality but failed otherwise. In the FC method, $S_k$ neglects natural mortality (Eq (24b)), so it increases without bound while $\beta_k$ decays to zero due to division by $S_k$ (Eq (24d)).

Fig 1A confirms that the absolute error in the FC method estimate of $S(t)$ increases linearly as $\mu_c (S) t$, where $\mu_c$ is the constant per capita natural mortality rate and $(S)$ is the continuous-time average of $S(t)$. In practice, the FC method fails whenever natural mortality in the underlying population is non-negligible. Since the S and SI methods address this limitation at effectively no computational cost, we do not present further analysis of the FC method.

In Fig 1B, the SI method estimate of $\beta(t)$ was very accurate (RRMSE $\approx 0.2\%$), whereas the S method estimate peaked too early and too high (RRMSE $\approx 2.4\%$).

3.2 Effects of process and observation error

We applied the S and SI methods without input error to four reported incidence time series $C_k$, simulated using the same parameter values but with different levels of process and observation noise. The first simulation was purely deterministic, while the remaining three included (i) environmental stochasticity [ES], (ii) ES and demographic stochasticity [ES+DS], or (iii) ES, DS, and observation error [ES+DS+OE]. Fig 2 shows the resulting estimates $Z_k$, $I_k$, and $\beta_k$ of true incidence $Z(t)$, prevalence $I(t)$, and the seasonally forced transmission rate $\beta(t)$.

Noise of any type introduces random fluctuations in $C_k$ on top of longer-term (e.g., seasonal) variation. Noise in $C_k$ is propagated to $Z_k$ (Fig 2A) and $I_k$ (Fig 2B), because (i) in both the S and SI methods, we scale $C_{k+r}$ by a constant factor of $p_{rep}^{-1}$ $\geq 1$ to compute $Z_k$ (Eqs (25a) and (26a)); (ii) in the S method, we scale $C_{k+1-r}$ by a constant factor of $[p_{rep}(1 + \mu_c)1]^{-1}$ to compute $I_k$ (Eq (25c) after substitution of Eq (25a)); and (iii) in the SI method, $I_k$ contains a weighted sum of $C_r$ terms (Eq (38b)).

Noise in $Z_k$ and $I_k$ is amplified in $\beta_k$ (Fig 2C), distorting the correct temporal pattern, for the following reason. When $Z$ and $I$ are close to zero, small absolute changes in either yield large relative changes in the ratio $Z/I$ and in turn $\beta_k$, which contains a factor of $Z_k/1$ in the S method (Eq (25d)) and $(Z_k + Z_{k+1})/(2I_k)$ in the SI method (Eq (26d)). Hence low amplitude
noise in $Z_k$ and $I_k$ appears as spurious, higher amplitude noise in $\beta_k$. This is an important issue in practice, because the incidence of endemic diseases is typically very small relative to the population size, and periodic fluctuations bringing incidence even closer to zero are common for many diseases [4, 14, 42].

Fig 2 shows that the SI method is much better than the S method at resisting noise propagation. One reason is the effective smoothing of incidence in the SI method, which replaces $Z_{k+1}$ with $(Z_k + Z_{k+1})/2$ in the computation of $\beta_k$ (compare Eqs (25d) and (26d)). We expose a second reason in §3.2.1 below by comparing the variance in $I_k$ induced by observation error, between the two methods. (We expect similar results for process error.)
Consider the S and SI method estimates of prevalence $I(t_k)$,

$$i^{[S]}_k = \left( \frac{C_{k+1}}{\gamma + \mu_k} \right)^k, \quad t^{[S]}_k = I_0 \left( \frac{1 - \left( \gamma + \mu_k \right) \Delta t}{1 + \frac{1}{2} \left( \gamma + \mu_k \right) \Delta t} \right)^k + \sum_{i=0}^{k-1} \frac{C_{i+1}}{\gamma + \mu_k} \Delta t \left( \frac{1 - \left( \gamma + \mu_k \right) \Delta t}{1 + \frac{1}{2} \left( \gamma + \mu_k \right) \Delta t} \right)^{k-1-i}. \tag{54b}$$
Here, \( g = [t_{\text{gen}}]_{\Delta t}/\Delta t \) and \( r = [t_{\text{rep}}]_{\Delta t}/\Delta t \) are the mean generation interval and case reporting delay in units of the observation interval, rounded to the nearest integer. These estimates are obtained from Eq (25c) (after substitution of Eqs (25a)) and (38b) when we assume a constant natural mortality rate \( \mu_c \). Following §2.3.3, suppose reported incidence is generated from true incidence \( Z(t_k) \) via 

\[
C_{k+r}^{\text{ind}} \sim \text{Binomial}(Z(t_k), p_{\text{rep}}).
\]

Then the variance of \( C_{k+r} \) is

\[
\text{Var}(C_{k+r}) = Z(t_k)p_{\text{rep}}(1 - p_{\text{rep}}).
\]  

(55)

It follows from Eqs (54) and (55) and the identity \( \text{Var}(aX) = a^2 \text{Var}(X) \) that

\[
\text{Var}(I_k^{(S)}) = \frac{(1 - p_{\text{rep}})Z(t_{k+1})}{p_{\text{rep}}((\gamma + \mu_c)\Delta t)^2},
\]

(56a)

\[
\text{Var}(I_k^{(SI)}) = \sum_{i=0}^{k-1} \frac{(1 - p_{\text{rep}})Z(t_{i+1})}{p_{\text{rep}}[1 + \frac{1}{2}((\gamma + \mu_c)\Delta t)]^2} \left( \frac{1 - \frac{1}{2}((\gamma + \mu_c)\Delta t)}{1 + \frac{1}{2}((\gamma + \mu_c)\Delta t)} \right)^{2(k-i)}.
\]

(56b)

If \( Z(t) \) has a well-defined average \( \langle Z \rangle \), then replacing instances of \( Z \) in Eq (56) with \( \langle Z \rangle \) and taking the limit as \( k \to \infty \), we obtain the average variances

\[
\langle \text{Var}(I_k^{(S)}) \rangle = \frac{(1 - p_{\text{rep}})\langle Z \rangle}{2p_{\text{rep}}((\gamma + \mu_c)\Delta t)},
\]

(57a)

\[
\langle \text{Var}(I_k^{(SI)}) \rangle = \lim_{k \to \infty} \left\{ \frac{(1 - p_{\text{rep}})\langle Z \rangle}{2p_{\text{rep}}((\gamma + \mu_c)\Delta t)} \left[ 1 - \frac{1 - \frac{1}{2}((\gamma + \mu_c)\Delta t)}{1 + \frac{1}{2}((\gamma + \mu_c)\Delta t)} \right]^{2k} \right\}
\]

(57b)

Comparing these with \( \langle \text{Var}(C_k) \rangle = \langle Z \rangle p_{\text{rep}}(1 - p_{\text{rep}}) \) using reference parameter values \( t_{\text{gen}} = \gamma^{-1} = 13 \) days, \( \mu_c = 0.04 \) year\(^{-1} \), and \( \Delta t = 1 \) week, we obtain

\[
\frac{\langle \text{Var}(I_k^{(S)}) \rangle}{\langle \text{Var}(C_k) \rangle} = \frac{1}{p_{\text{rep}}^2((\gamma + \mu_c)\Delta t)^2} \approx \frac{3.44}{p_{\text{rep}}^2},
\]

(58a)

\[
\frac{\langle \text{Var}(I_k^{(SI)}) \rangle}{\langle \text{Var}(C_k) \rangle} = \frac{1}{2p_{\text{rep}}^2((\gamma + \mu_c)\Delta t)^2} \approx \frac{0.93}{p_{\text{rep}}^2},
\]

(58b)

where \( t_{\text{inf}} = (\gamma + \mu_c)^{-1} \) is the mean time spent infected. Hence, while both the S and SI methods suffer from propagation of noise from reported incidence \( C_k \) to estimated prevalence \( I_k \), particularly for \( p_{\text{rep}} \ll 1 \), the S method tends to be much worse (by a factor of 3.44/0.93 ≈ 3.7 in this example). Comparative resistance to noise propagation is a distinct advantage of the SI method over the S method.

### 3.3 Averaging the raw estimate of \( \beta(t) \)

Fig 3A displays two raw estimates \( \beta_k \) (S and SI methods, applied without input error) of a seasonally forced \( \beta(t) \), each spanning 1000 years (only the first 10 years are shown). The estimates
embed 1000 1-year cycles, which are displayed in Fig 3B and 3C together with their 1-year average (cf. §2.2.5).

Both estimates suffered from spurious noise distorting the correct seasonal pattern, caused by process and observation error in the data-generating process (cf. §3.2). As in Fig 2C, the variance was markedly smaller with the SI method. Averaging the embedded 1-year cycles recovered the true 1-year cycle from the noise. In the absence of input error, the S method appears to carry a slight bias (peaking early and too high, as in Fig 1), whereas the SI method is nearly unbiased.

While some existing infectious disease time series span several centuries [15], in practice, averaging as in Fig 3B and 3C is sensible only over time intervals during which the underlying seasonal pattern in transmission is roughly stationary.
3.4 Smoothing the raw estimate of $\beta(t)$

Regardless of whether averaging is employed, comparison of Fig 3C and 3D shows that it is helpful to smooth the $\beta_k$ time series by fitting a loess curve $\beta_{loess}(t; q)$ (cf. §2.2.6). An appropriate degree of smoothing (i.e., choice of loess smoothing parameter $q$) eliminated spurious noise without significantly increasing bias.

Fig 4A quantifies the effect of smoothing $\beta_k$ using the optimal value $q_{opt}$ for parameter $q$ (cf. §2.2.6). It plots RRMSE before and after smoothing as a function of the amount of noise in the simulated reported incidence data, which was modulated by varying the case reporting probability $p_{rep}$ between 0.01 and 1 (more noise for smaller $p_{rep}$; see Eq (31)).

![Fig 4A](https://doi.org/10.1371/journal.pcbi.1008124.g004)

**Fig 4. Reduction in $\beta(t)$ estimation error with optimal loess smoothing.** The horizontal axis measures the case reporting probability $p_{rep}$ for which 41 values equally spaced on a logarithmic scale between 0.01 and 1 were considered. Using each value of $p_{rep}$ and reference values (Table 1) for all other parameters, 100 reported incidence time series ($\Delta t = 1$ week, $n = 1042$) were simulated accounting for environmental noise in transmission ($\epsilon = 0.5$), demographic stochasticity, and random under-reporting of cases (measured by $p_{rep}$). The underlying seasonally forced $\beta(t)$ (Eq (27)) was estimated from reported incidence using the S and SI methods, both applied without input error, yielding two raw estimates $\beta_k$ per simulation. Smooth loess curves $\beta_{loess}(t; q)$ ($q = 10, \ldots, 110$; cf. §2.2.6) were fit to each $\beta_k$ time series. The optimal $q$ for a given time series, denoted by $q_{opt}$, was defined as the value that minimized RRMSE ($\beta_{loess}$) in $\beta_{loess}(t; q)$. Overall, for each value of $p_{rep}$ and each $\beta(t)$ estimation method (S and SI), 100 values of $q_{opt}$ were obtained corresponding to 100 $\beta_k$ time series. Plotted on the vertical axis as functions of $p_{rep}$ are the median and 5th and 95th percentiles of [Panel A] RRMSE in the raw estimates $\beta_k$ [dashed lines] and optimal loess estimates $\beta_{loess}(t; q_{opt})$ [solid lines] and [Panel B] $q_{opt}$. Lines and bands indicate the median and 5th–95th percentile range, respectively. Results for the S and SI methods are shown in green and red, respectively.

https://doi.org/10.1371/journal.pcbi.1008124.g004
Using the optimal loess estimate $\beta_{\text{loess}}(t_k; q_{\text{opt}})$ instead of the raw estimate $\beta_k$ significantly reduced RRMSE—by at least 46% for the S method and 17% for the SI method across all simulations. Although raw estimates generated by the SI method were consistently more accurate (expected in light of Fig 3B and 3C), optimal loess estimates were comparable between the S and SI methods for $p_{\text{rep}} > 0.2$ (RRMSE $\approx 3\%$). For $p_{\text{rep}} < 0.2$ (severe under-reporting of cases), optimal smoothing failed to an increasing extent to recover the underlying $\beta(t)$ from noise in $\beta_k$. In this setting, the S method was greatly outperformed by the SI method, which is more resilient to noise in reported incidence (cf. §3.2).

Fig 4B shows that median $q_{\text{opt}}$ was roughly constant for $p_{\text{rep}} > 0.1$, with

$$\text{median } q_{\text{opt}} \approx \begin{cases} 65 & \text{for the S method,} \\ 53 & \text{for the SI method.} \end{cases}$$

(59)

More smoothing (greater $q$) was required to minimize RRMSE for $p_{\text{rep}} > 0.1$. More generally, Fig 4 indicates that the S and SI methods should always include a smoothing step. Hence, in the remaining analysis, we always smooth $\beta_k$.

### 3.5 Sensitivity to data-generating parameters

Here, we characterize the sensitivity of $\beta(t)$ estimation error to parameters of the data-generating process. As in §§3.1–3.4, we consider the ideal case in which the user-specified values of all input parameters are equal to the true (data-generating) values. The details of our analysis are outlined in §2.6.1.

Fig 5 plots the median RRMSE in estimates of a seasonally forced $\beta(t)$ ($\text{Eq (27)}$) from 1000 realizations of a reported incidence time series, as a bivariate function of the mean $\langle \beta \rangle$ and amplitude $\alpha$ of seasonal forcing. To aid interpretation, the $\langle \beta \rangle$ axis was scaled to measure the basic reproduction number $R_0$ ($\text{Eq (2)}$).

Fig 6 plots median RRMSE as a univariate function of each of 6 additional parameters—the initial states $S_0$ and $I_0$, vital rates $\nu_c$ and $\mu_c$, mean generation interval $t_{\text{gen}}$, and case reporting probability $p_{\text{rep}}$—with the focal parameter assigned values between $\frac{1}{4}$ and 4 times its reference value (Table 1). The horizontal axis measures the ratio of the focal parameter’s data-generating value to its reference value, so that commensurate deviations from the reference case can be compared across the 6 parameters.

In order to produce Figs 5 and 6, we assigned reference values (Table 1) to all but the focal data-generating parameter(s) (e.g., all except $\langle \beta \rangle$ and $\alpha$ in Fig 5). We fit loess curves $\beta_{\text{loess}}(t; q)$ to all raw estimates $\beta_k$ of $\beta(t)$, and recorded the RRMSE in $\beta_{\text{loess}}(t_k; q)$. Motivated by Fig 4B and Eq (59), we fixed $q = q^*$, taking $q^* = 65$ with the S method and $q^* = 53$ with the SI method.

A pattern in our interpretation of Figs 5 and 6 below is that error in $\beta(t)$ estimation is sensitive to a parameter if changes in that parameter (i) cause incidence $Z(t)$ or prevalence $I(t)$ to approach zero more frequently or more closely, or (ii) increase noise in estimated incidence $Z_k$ or estimated prevalence $I_k$. Both outcomes incorrectly increase noise in $\beta_k$ (cf. §3.2).

When the noise in $\beta_k$ is extreme, setting $q = q^*$ can undersmooth the time series ($q^* < q_{\text{opt}}$). In this case, smaller RRMSE is attainable by determining $q_{\text{opt}}$ and setting $q = q_{\text{opt}}$. Nevertheless, we did not find $q_{\text{opt}}$ for each of the $5\times10^6$ time series considered by Figs 5 and 6, which would have increased the total computation time by a factor of 100. Consequently, Figs 5 and 6 may overestimate the sensitivity of $\beta(t)$ estimation error to data-generating parameters. (In §S5.3 of S1 Text, we show that the quantitative effect of choosing $q^*$ over $q_{\text{opt}}$ is likely to be small.)

#### 3.5.1 Sensitivity to the basic reproduction number $R_0$, and seasonal amplitude $\alpha$ (Fig 5)

For fixed $\alpha$, median RRMSE was a non-monotonic function of $R_0$. The reason is that
Fig 5. Sensitivity of $\beta(t)$ estimation error to the mean $\beta$ and amplitude $\alpha$ of seasonal forcing. Contained in each panel are heatmaps of median RRMSE (Eq (33)) in estimates of a seasonally forced $\beta(t)$ (Eq (27)) from simulated reported incidence time series, as a bivariate function of the mean $\beta$ and amplitude $\alpha$ of seasonal forcing. The $\beta$ axis has been scaled to measure the basic reproduction number $R_0$ (Eq (2)). When simulating reported incidence, reference values (Table 1) were assigned to all data-generating parameters except $\beta$ and $\alpha$. A grid of $(R_0, \alpha)$ pairs with levels $R_0 = 2, 3, \ldots, 32$ and $\alpha = 0, 0.01, \ldots, 0.2$ was considered, with $\beta$ defined for each value of $R_0$ via Eq (2). For each parametrization, 1000 simulations were performed with environmental stochasticity [ES] ($\epsilon = 0.5$) and with or without demographic stochasticity [DS] and observation error [OE], as indicated by row: [Row A] without DS or OE ($p_{rep} = 1, t_{rep} = 0$ weeks), [Row B] with DS but without OE ($p_{rep} = 1, t_{rep} = 0$ weeks), [Row C] with DS and OE ($p_{rep} = 0.25, t_{rep} = 2$ weeks). Corresponding mock birth and natural mortality time series were created, then $\beta(t)$ was estimated from the data using [Left] the S method and [Right] the SI method, all without input error. For each set of estimates of $\beta(t)$ (1000 estimates per parametrization, per simulation method, per estimation method), the median RRMSE was calculated (after smoothing with fixed $q$; see Eq (59)) and displayed as one point in the appropriate heatmap, coloured according to the logarithmic scale on the right. The darkest blue indicates median RRMSE less than 0.01. https://doi.org/10.1371/journal.pcbi.1008124.g005
changes in (effective) $R_0$ are responsible for dynamical transitions that alter the structure of solutions of the SIR model \cite{28, 42, 43}. Specifically, as $R_0$ is increased from 2 to 32, minimum incidence $Z_{\text{min}}$ and minimum prevalence $I_{\text{min}}$ on the attractor varies non-monotonically (see Fig 2 in \cite{28}). Smaller $Z_{\text{min}}$ and $I_{\text{min}}$ yield more noise in $\beta_k$, and correspondingly greater
RRMSE. For fixed $R_0$, $I_{min}$ decreases monotonically as $\alpha$ is increased from 0 to 1 (see Fig 11 in [43]), so we expect median RRMSE to increase monotonically with $\alpha$, as observed in Fig 5.

### 3.5.2 Sensitivity to the initial state ($S_0$, $I_0$) (Fig 6)

RRMSE is sensitive to the data-generating $S_0$, but not $I_0$. The reference values of $S_0$ and $I_0$ are taken from a point ($S^*$, $I^*$, $R^*$) on the attractor of the SIR model (1) with seasonally forced $\beta(t)$ and constant vital rates $\nu_c$ and $\mu_c$ (cf. §2.3.4). When $S_0$ is far from $S^*$, the solution of system (1) undergoes extreme fluctuation before relaxing to the attractor, and both $Z$ and $I$ approach zero during the transient, generating spurious noise at the start of the $\beta_k$ time series.

Note that $I_0$ differing from $I^*$ has a much smaller effect on dynamics than $S_0$ differing from $S^*$ by the same factor. Since $I^* \ll S^*$, the perturbation of ($S_0$, $I_0$, $R_0$) from the attractor is much smaller.

### 3.5.3 Sensitivity to vital rates $\nu_c$ and $\mu_c$ (Fig 6)

Median RRMSE was a non-monotonic function of the data-generating birth rate $\nu_c$. This behaviour arises because scaling $\nu_c$ is dynamically equivalent to scaling $R_0$ by the same factor [2, 28], and median RRMSE is a non-monotonic function of $R_0$ (cf. §3.5.1 above).

Changing the data-generating natural mortality rate $\mu_c$ had a negligible effect on RRMSE. This is unsurprising, because natural death is dominated by recovery and disease-induced death in governing the rate of infected decrease. That is, $\gamma \gg \mu(t)$ in Eq (1b), so changes in $\mu_c$ by up to a factor of 4 have little effect on dynamics.

### 3.5.4 Sensitivity to the mean generation interval $t_{gen}$ (Fig 6)

Median RRMSE increased rapidly as the data-generating $t_{gen}$ was made smaller than $2^{-4/5}$ (roughly 0.57) times its reference value of 13 days. A period-doubling bifurcation occurs near this value of $t_{gen}$, and the attractor of the SIR model (1) acquires a 2-year cycle with much smaller $Z_{min}$ and $I_{min}$ (see §S5.3.1 of S1 Text). Propagation of noise to $\beta_k$ intensifies, resulting in greater RRMSE.

The performance of the S method fluctuates more as a function of $t_{gen}$ than that of the SI method. This occurs because the S method rounds $t_{gen}$ in the numerator of Eq (25c) to the nearest integer multiple of $\Delta t$, and the rounding error oscillates as a function of $t_{gen}$. The SI method does not require rounding, so these fluctuations are not observed.

### 3.5.5 Sensitivity to the case reporting probability $p_{rep}$ (Fig 6)

When the reported incidence data contain observation error (Fig 6C), RRMSE is additionally sensitive to the case reporting probability $p_{rep}$. Decreasing $p_{rep}$ increases noise in reported incidence $C_k$ (Eq (31)), which is propagated to estimated incidence $Z_k$, estimated prevalence $I_k$, and in turn $\beta_k$ (cf. §3.2).

Fig 6 suggests weak sensitivity to $p_{rep}$. However, noise in $Z_k$ and $I_k$ is amplified in $\beta_k$ to the extent that $Z$ and $I$ are close to zero (cf. §3.2). Hence, for example, if the data-generating $t_{gen}$ were assigned a value smaller than half its reference value of 13 days, then we would have observed more acute sensitivity to $p_{rep}$ as a result of closer approaches to zero by $Z$ and $I$ (cf. §3.5.4 above).

### 3.5.6 S method versus SI method (Figs 5 and 6)

Both the S and SI methods performed well, estimating $\beta(t)$ with median RRMSE less than 10% across most parametrizations. However, by resisting noise propagation (cf. §3.2), the SI method was significantly less sensitive to the data-generating parameters and to the addition of demographic stochasticity and observation error.

### 3.6 Sensitivity to mis-specification of input parameters

In §3.5, we considered the ideal situation in which the user knows the true (data-generating) values of the input parameters. Here, we examine the more realistic situation in which the
user specifies input parameters with some error. The effect of mis-specification is particularly important for parameters that are difficult to estimate accurately, such as the case reporting probability \( p_{\text{rep}} \). The details of our analysis are outlined in §2.6.2.

We restrict our attention to application of the SI method to reported incidence data simulated with process and observation error. Differences in RRMSE between methods of data simulation and \( \beta(t) \) estimation are dominated (by an order of magnitude) by the increase in RRMSE resulting from mis-specified input parameters.

Fig 7A plots the median RRMSE in estimates of \( \beta(t) \) from 1000 realizations of a reported incidence time series, as a univariate function of the factor by which an input parameter—one of the initial states \( S_0 \) and \( I_0 \), mean generation interval \( t_{\text{gen}} \), vital rates \( v_c \) and \( \mu_c \), and case reporting parameters \( P_{\text{rep}} \) and \( t_{\text{rep}} \)—was mis-specified. The specified value of the focal parameter was varied between \( \frac{1}{2} \) and 4 times its true (data-generating) value, and the remaining parameters were specified without error.

### 3.6.1 Sensitivity to error in the specified initial state \( (S_0, I_0) \)

Fig 7 shows that error in the specified value of \( S_0 \) is propagated non-negligibly to estimates of \( \beta(t) \), while mis-specification of \( I_0 \) has practically no effect on \( \beta(t) \) estimation error. Eqs (40) and (41) show that specifying incorrect values \( S'_0 \) and \( I'_0 \) for \( S_0 \) and \( I_0 \) creates errors in \( S_k \) and \( I_k \) that vanish geometrically as \( k \to \infty \). However, since \( t_{\text{inf}} \gg t_{\text{inf}} \), the decay is significantly slower in \( S_0 \). Indeed, with reference values \( \mu_c = 0.04 \text{ year}^{-1}, t_{\text{gen}} = 3002\text{ days} \) and \( \Delta t = 1 \text{ week} \), we find that a factor of 10 reduction in error between times \( t_k \) and \( t_{k+1} \) requires just \( i = 5 \) in the infected time series, compared to \( i = 5002 \) in the susceptible time series. Hence, in practice, accurate reconstruction of \( S(t) \), \( I(t) \), and in turn \( \beta(t) \) relies on accurate specification of \( S_0 \) but not \( I_0 \). We address sensitivity to mis-specification of \( S_0 \) in §3.7 below.

### 3.6.2 Sensitivity to error in the specified birth rate \( v_c \) and case reporting probability \( p_{\text{rep}} \)

Mis-specifying \( v_c \) or \( p_{\text{rep}} \) by a factor of \( 2^{1/10} \) (7.2%) yielded median RRMSE greater than 30%. Mis-specifying by a factor of \( 2^{-1/10} \) (6.7%) led to even worse estimates of \( \beta(t) \), with median

Fig 7. Sensitivity of \( \beta(t) \) estimation error to the user-specified values of input parameters. [Panel A] Median RRMSE (Eq (33)) in estimates of \( \beta(t) \) from simulated reported incidence time series (\( \Delta t = 1 \text{ week} \), \( n = 1042 \)), as a univariate function of the factor by which an input parameter was mis-specified. One thousand simulations were performed using fixed values (Table 1) for all data-generating parameters. The simulations accounted for environmental stochasticity [ES] (\( \epsilon = 0.5 \)), demographic stochasticity [DS], and observation error [OE] (\( p_{\text{rep}} = 0.25, t_{\text{rep}} = 2 \text{ weeks} \)). For each simulation, corresponding mock birth and natural mortality time series were created, and \( \beta(t) \) was estimated from the data using the SI method. True (data-generating) values were specified for all input parameters except the focal parameter (indicated by the legend), for which 41 values logarithmically spaced between \( \frac{1}{2} \) and 4 times the true value were specified in turn. Each input parameterization yielded 1000 estimates of \( \beta(t) \), whose median RRMSE was calculated (after smoothing with fixed \( q \); see Eq (59)) and displayed as one point in the appropriate graph. [Panel B] Result of repeating the analysis from Panel A in which \( S_0 \) was specified with varying amounts of error, but with the initially erroneous value of \( S_0 \) updated using the method of peak-to-peak iteration (PTPI; 25 iterations) prior to \( \beta(t) \) estimation. The original result, obtained without PTPI, is presented for comparison.

https://doi.org/10.1371/journal.pcbi.1008124.g007
RRMSE exceeding 100% (not visible in Fig 7A). Eqs (42) and (47) show that specifying incorrect values $n_c$ and $p_{rep}$ for $n_c$ and $p_{rep}$ generates absolute errors in $S_k$ that tend to increase over time ($k$) to a limit. In practice, systematic underestimation of births by the $B_k$ time series (modeled here by $n_c < n_c^*$) and overestimation of incidence by the $Z_k$ time series ($p_{rep} < p_{rep}^*$) can cause $S_k$ to eventually take negative values. Once this happens, attempts by the S and SI methods to reconstruct $\beta(t)$ fail completely.

While this failure may seem concerning, it should be viewed as a tool for diagnosing incorrect birth and case reporting rates: if the S or SI method yields negative $S_k$ for any $k$, then one should speculate that births were underestimated or that incidence was overestimated, and retry the algorithm with a scaled up $B_k$ time series and/or with greater $p_{rep}$ (as $Z_k$ is computed by scaling reported incidence by a factor of $p_{rep}^*$; see Eqs (25a) and (26a)). Of course, overcorrection is also undesirable (cf. right half of Fig 7A). In our work, we have found that a brief exploration of possible adjustments—factors by which to increase $B_k$ and/or $p_{rep}$—suffices to identify ones that prevent both negative $S_k$ and pronounced transient dynamics at the start of the susceptible time series (indicating under- or overcorrection).

### 3.7 Solution of the $S_0$ estimation problem using PTPI

In §3.6, we showed that the performance of the S and SI methods is highly sensitive to misspecification of the initial number of susceptibles $S_0$. Here, we assess PTPI as a way to iteratively improve initially poor estimates of $S_0$ prior to reconstruction of $S(t)$ and $\beta(t)$.

Fig 8 demonstrates PTPI for an example in which $S_0$ was overestimated by a factor of 4 by a user of the SI method. PTPI yielded increasingly accurate estimates of $S_0$ and correspondingly more accurate reconstructions of $S(t)$ (Fig 8B) and $\beta(t)$ (Fig 8C). Fig 7B repeats our analysis from §3.6, except using PTPI (25 iterations) to update the incorrect estimate of $S_0$ prior to reconstructing $\beta(t)$. We see that application of PTPI in conjunction with the SI method enables accurate $\beta(t)$ reconstruction independently of errors in the initial estimate of $S_0$. This result is unsurprising in light of Fig 9, which shows that PTPI converges rapidly (in fewer than 10 iterations) to an accurate estimate of $S_0$ independently of the initial guess. Due to process error in the underlying dynamics, the relative error in the limiting estimate of $S_0$ varied between the 1000 realizations of reported incidence considered (5th–95th percentile range $[-11.9, 12.5]\%$, median 0.9%). Process error creates variance in the time between peaks in incidence (see Fig 8A), violating the periodicity assumption of PTPI (the theoretical basis of the technique; cf. §2.8). Nevertheless, Figs 7–9 demonstrate that PTPI can significantly improve $S(t)$ and $\beta(t)$ reconstruction from roughly periodic incidence data.

### 3.8 Run time

We implemented the S and SI methods and PTPI in R and ran them on a MacBook Pro with a 2.4 GHz Quad-Core Intel Core i5 chip. The S and SI methods are both extremely fast, requiring a total of 0.124 and 0.376 seconds, respectively, to generate a reconstruction of $\beta(t)$ from 1000 years of weekly reported incidence ($\Delta t = 1$ week, $n = 52142$). Application of PTPI in conjunction with either method increases the run time with each iteration, but the total run time remains inconsequential due to the rate of convergence of the iterations to a limiting estimate of $S_0$. For example, when we applied PTPI to the same simulated data, the truncation step (Box 5) added 0.094 seconds to the total run time, while the iteration step (Box 5) added 1.01 seconds per iteration on average.
4 Discussion

We have compared three fast methods of estimating the time-varying transmission rate $\beta(t)$ from reported incidence time series, all based on discretizations of the SIR model (1). Fine and Clarkson’s method [6], referred to here as the FC method, fails rapidly in practice, because it treats natural mortality in the susceptible population as negligible. Although Krylova’s method [24], adapted here as the S method, corrects this limitation of the FC method and is accurate.
for certain simulated data, her method suffers from extreme sensitivity to process and observation error. Specifically, noise in reported incidence is spuriously propagated to its estimates of $\beta(t)$. Our algorithm for transmission rate estimation, referred to here as the SI method and based on deJonge’s method [25], is much more resilient to noise in reported incidence and therefore superior to the S method.

Like its predecessors, the SI method is sensitive to (i) certain input parameters: the initial number of susceptible individuals $S_0$, the case reporting probability $p_{\text{rep}}$, and the mean generation interval $t_{\text{gen}}$; as well as (ii) vital data: times series of births and natural mortality without substantial systematic errors.

The requirement of a good estimate of $S_0$ has been a major barrier to use of existing fast methods of $\beta(t)$ estimation (including those presented in [6, 24, 25]). We have proposed and demonstrated PTPI as a valid and fast technique for obtaining accurate estimates of $S_0$ from poor initial guesses, conditional on periodic dynamics (epidemic recurrence with a fixed period). Use of the SI method in conjunction with PTPI represents a major advance over the existing fast methods.

Estimation of the case reporting probability $p_{\text{rep}}$ is possible using maximum likelihood approaches, including trajectory matching. However, a fast way to obtain a crude estimate of $p_{\text{rep}}$ is to divide cumulative reported incidence over the time interval $[t_0, t_n]$, by the cumulative

\[ \frac{\text{Cumulative reported incidence}}{t_n - t_0} \]

For a more accurate estimate, one could use maximum likelihood approaches, such as trajectory matching.
incidence that is expected from the unforced SIR model (system (1) with $\beta \equiv \langle \beta \rangle$, $\nu \equiv \nu_c$, and $\mu \equiv \mu_c$) at equilibrium:

$$p_{\text{rep}} \approx \frac{\sum_{k=1}^{n} C_k}{\nu_c N_0 (1 - \frac{1}{R_0}) (t_n - t_0)}.$$

(60)

This approximation can be made in temporal subintervals to obtain a time-varying reporting rate, which would replace the constant $p_{\text{rep}}$ in Eq (26a). Sensitivity of the SI method to mis-specification of the mean generation interval ($t_{\text{gen}}$) may be of greater concern, though if the distribution of the incubation period (time from infection to onset of symptoms) is narrow, then $t_{\text{gen}}$ will be well approximated by the (observable) mean serial interval [44].

Overall, the SI method, in conjunction with PTPI, represents a highly tractable approach to reconstructing susceptibles and $\beta(t)$ from infectious disease time series that span decades or centuries. It makes fewer assumptions about the disease and population of interest than the regression-based tSIR method [7, 23] (i.e., it does not require an infectious period equal to the observation interval, ignore susceptible mortality, or assume that cumulative incidence approximates cumulative births). Moreover, it is significantly less complex and much less computationally demanding than simulation-based methods of inference, such as iterated filtering [8, 19, 20] and generalized profiling [21, 22].

Even when the observed infectious disease time series is short enough that simulation-based methods are tractable, the approach to transmission rate reconstruction that we promote here can be usefully employed to provide better starting conditions at negligible computational cost.

Supporting information

S1 Text. Text supplement. A .pdf document containing annotated R code, making the results reported here completely reproducible by the reader.

(PDF)

S1 File. Source files. A .zip archive containing all of the source files needed to compile S1 Text.

(ZIP)

Acknowledgments

We thank Ben Bolker, Jonathan Dushoff, and Sang Woo Park for helpful comments and discussion.

Author Contributions

Conceptualization: Mikael Jagan, Michelle S. deJonge, Olga Krylova, David J. D. Earn.

Formal analysis: Mikael Jagan, Michelle S. deJonge, David J. D. Earn.

Funding acquisition: Mikael Jagan, Olga Krylova, David J. D. Earn.

Investigation: Mikael Jagan, Michelle S. deJonge, David J. D. Earn.

Methodology: Mikael Jagan, Michelle S. deJonge, Olga Krylova, David J. D. Earn.

Project administration: Mikael Jagan, Michelle S. deJonge, David J. D. Earn.
Resources: David J. D. Earn.
Software: Mikael Jagan.
Supervision: David J. D. Earn.
Validation: Mikael Jagan.
Visualization: Mikael Jagan.
Writing – original draft: Mikael Jagan, Michelle S. deJonge.
Writing – review & editing: Mikael Jagan, David J. D. Earn.

References


Supporting Information

Fast estimation of time-varying infectious disease transmission rates

PLOS Computational Biology

Mikael Jagan, Michelle S. deJonge, Olga Krylova, David J. D. Earn

September 23, 2020
## Contents

S0 Preliminaries .................................................. 1

S1 Example of $\beta(t)$ estimation using the FC, S, and SI methods ........................................ 2
   S1.1 Creating a list of parameter values ............................................ 2
   S1.2 Simulating time series data ..................................................... 4
   S1.3 Estimating the time-varying transmission rate .................................. 5
   S1.4 Measuring estimation error .......................................................... 9

S2 Effect of process and observation error .......................................... 10

S3 Averaging the raw estimate of $\beta(t)$ ................................................................. 13

S4 Smoothing the raw estimate of $\beta(t)$ ............................................................... 16

S5 Sensitivity to data-generating parameters ............................................... 22
   S5.1 Sensitivity to $R_0$ and $\alpha$ ...................................................... 23
   S5.2 Sensitivity to $S_0$, $I_0$, $\nu_c$, $\mu_c$, $t_{gen}$, and $p_{rep}$ ................. 27
   S5.3 A note on smoothing ........................................................................ 30
      S5.3.1 $C_k$ and $\beta_k$ for $t_{gen} = (2^{-1.5}, 1, 2^{1.5}) \cdot 13$ days .......... 30
      S5.3.2 $\beta_{loess}(t; q^*)$ and $\beta_{loess}(t; q_{opt})$ for $t_{gen} = (2^{-1.5}, 1, 2^{1.5}) \cdot 13$ days .. 32
      S5.3.3 Discussion ................................................................................. 35

S6 Sensitivity to error in input parameters .................................................. 36
   S6.1 A note on smoothing ........................................................................ 39

S7 Estimating $S_0$ via PTPI: Example ......................................................... 39
   S7.1 Truncation step ................................................................................. 40
   S7.2 Iteration step ................................................................................... 43

S8 Estimating $S_0$ via PTPI: Convergence ................................................... 47

S9 Appendix: Choice of discretization in the SI method .................................... 50
   S9.1 Nine discretization schemes .............................................................. 50
   S9.2 Comparison of RRMSE, bias, and variance ...................................... 51
      S9.2.1 RRMSE .................................................................................... 51
      S9.2.2 Bias and variance ..................................................................... 55
S0 Preliminaries

This supplement to the manuscript is compiled from the source file S1_Text.Rnw using R version 4.0.2 (2020-06-22) [1] and these R package versions:

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>knitr</td>
<td>1.29</td>
</tr>
<tr>
<td>tikzDevice</td>
<td>0.12.3.1</td>
</tr>
<tr>
<td>colorRamps</td>
<td>2.3</td>
</tr>
<tr>
<td>RColorBrewer</td>
<td>1.1.2-2</td>
</tr>
<tr>
<td>scales</td>
<td>1.1.1</td>
</tr>
<tr>
<td>deSolve</td>
<td>1.28</td>
</tr>
<tr>
<td>adaptivetau</td>
<td>2.2-3</td>
</tr>
</tbody>
</table>

Our primary aim here is to make our results entirely reproducible by the reader. Our secondary aim is to make our methods available to potential users. To this end, we introduce our R package fastbeta, which implements

- the FC, S, and SI methods for estimating time-varying transmission rates $\beta(t)$ from time series data; and
- peak-to-peak iteration (PTPI) for estimating the initial number of susceptible individuals $S_0$ from time series data.

All methods are based on the SIR model with time-varying rates of birth, death, and transmission:

\[
\begin{align*}
\frac{dS}{dt} &= \nu(t)N_0 - \beta(t)SI - \mu(t)S, \quad (1a) \\
\frac{dI}{dt} &= \beta(t)SI - \gamma I - \mu(t)I, \quad (1b) \\
\frac{dR}{dt} &= \gamma I - \mu(t)R, \quad (1c)
\end{align*}
\]

where $\gamma = 1/t_{\text{gen}}$.

The most recent version of fastbeta is located in our GitHub repository and can be installed using `install_github()` from the remotes package.

```r
if (!require(remotes)) install.packages("remotes")
remotes::install_github("davidearn/fastbeta")
```

However, readers attempting to compile this document from source must install fastbeta from the plos branch of the repository, which houses the version used at the time of this writing.

```r
if (!require(remotes)) install.packages("remotes")
remotes::install_github("davidearn/fastbeta", ref = "plos")
```
For complete compilation instructions, refer to README.md.

Here is a list of functions implemented in the plos version of fastbeta:

```r
library(fastbeta)
ls("package:fastbeta")
## [1] "compute_rrmse"  "estimate_beta_FC"  "estimate_beta_S"
## [4] "estimate_beta_SI"  "get_peak_times"  "make_data"
## [7] "make_par_list"  "ptpi"  "test_s2dgbeta"
## [10] "test_s2dgpars"  "test_s2inpars"
```

We will introduce these on the fly. More complete documentation can be accessed by running ?fastbeta.

In the sections to follow, we annotate the R code needed to reproduce results reported in the manuscript. Plotting commands have been suppressed, but are preserved in S1_Text.Rnw. Since our work involves millions of simulations, we have retained certain output in the directory RData/ to significantly reduce compilation time.

### S1 Example of $\beta(t)$ estimation using the FC, S, and SI methods

Fig 1 in the manuscript compares the output of the FC, S, and SI methods for a simulated time series of reported incidence. To reproduce Fig 1, we simulate time series data using system (1) with constant vital rates $\nu_c$ and $\mu_c$ and a seasonally forced transmission rate that includes environmental noise:

$$\beta_{\phi}(t) = \langle \beta \rangle \left[ 1 + \alpha \cos \left( \frac{2\pi t}{1\text{year}} + \phi(t; \epsilon) \right) \right].$$ (2)

Doing so, we obtain observations of reported incidence at equally spaced time points

$$t_k = t_0 + k\Delta t, \quad k = 0, \ldots, n.$$ (3)

We then apply each algorithm (FC, S, SI) to reconstruct the seasonally forced transmission rate from the data.

As this simulate-estimate routine is our main investigative approach going forward, we describe each step in detail here (and reproduce Fig 1 in the process), but we do not repeat these details in later sections.

### S1.1 Creating a list of parameter values

Simulating a reported incidence time series from system (1) requires a list of values for all data-generating parameters. Our function make_par_list() simplifies the task of
creating such a list. It does this by “filling in the blanks” when we want \( R_0 \), \( \langle \beta \rangle \), \( N_0 \), \( S_0 \), or \( I_0 \) to depend in a complicated way on other parameters’ values.

Below, we call `make_par_list()`, defining all arguments explicitly. Except for `dt_weeks`, which is the observation interval \( \Delta t \) in weeks, the values of time and rate parameters must be supplied in units \( \Delta t \) and per unit \( \Delta t \). In this call to `make_par_list()`, we indicate \( \Delta t = 1 \) week, \( t_0 = 2000 \) years, \( t_{\text{gen}} = 13 \) days, and \( \nu_c = \mu_c = 0.04 \text{ year}^{-1} \).

```r
## List of parameter values
par_list <- make_par_list(
  dt_weeks = 1,                # observation interval
  t0 = 2000 * (365 / 7) * 1,   # time of first observation
  prep = 1,                    # case reporting probability
  trep = 0,                    # case reporting delay
  hatN0 = 1e06,                # population size at 0 years
  N0 = NA,                     # population size at `t0`
  S0 = NA,                     # number susceptibles at `t0`
  I0 = NA,                     # number infecteds at `t0`
  nu = 0.04 * (7 / 365) * 1,  # birth rate (relative to `hatN0`) 
  mu = 0.04 * (7 / 365) * 1,  # natural mortality rate (per capita) 
  tgen = 13 * (1 / 7) / 1,    # mean generation interval
  Rnaught = 20,                # basic reproduction number
  beta_mean = NA,              # mean of seasonal forcing
  alpha = 0.08,                # amplitude of seasonal forcing
  epsilon = 0                  # s.d. of environmental noise
)
unlist(par_list) # printed as a named vector
```

`make_par_list()` requires that exactly one of \( R_0 \) and \( \langle \beta \rangle \) (arguments `Rnaught` and `beta_mean`) is defined in the function call. It sets the undefined parameter internally by enforcing the identity

\[
R_0 = \frac{\nu_c \hat{N}_0}{\mu_c} \cdot \frac{\langle \beta \rangle}{\gamma + \mu_c}.
\]  

(4)

Above, we “omitted” `beta_mean` from the function call (by supplying a value `NA`) and obtained the desired value in the output.
Values for $N_0$, $S_0$, and $I_0$ (arguments $N_0$, $S_0$, and $I_0$) were also set internally. This was done via numerical integration of system (1) with constant vital rates $\nu_c$ and $\mu_c$ and a seasonally forced transmission rate that excludes environmental noise:

$$\beta(t) = \langle \beta \rangle \left[ 1 + \alpha \cos \left( \frac{2\pi t}{1 \text{ year}} \right) \right]. \quad (5)$$

After integrating between times $t = 0$ years and $t = t_0$ (in this case 2000 years) `make_par_list()` chooses for $N_0$, $S_0$ and $I_0$ the values of $N^* = N(t_0)$, $S^* = S(t_0)$, and $I^* = I(t_0)$. This ensures that the initial state of any simulation using `par_list` is a point very near the attractor of the system being simulated.

In the above call to `make_par_list()`, we indicated the default values of all arguments. In future calls to this function, we will not specify arguments explicitly except for clarity or to request a value different from the default.

### S1.2 Simulating time series data

To reproduce the simulation considered in Fig 1, we pass `par_list` to our function `make_data()`, which returns the simulated time series data in a data frame.

```r
## Data frame containing time series data
def <- make_data(
  par_list = par_list,       # parametrization
  n = 20 * 365 / 7,         # time series length: ~20 years
  with_dem_stoch = FALSE    # no demographic stochasticity
)
head(df)
```

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>t_years</th>
<th>beta</th>
<th>beta_phi</th>
<th>N</th>
<th>S</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>104285.7</td>
<td>2000.000</td>
<td>1.164734e-05</td>
<td>1.164734e-05</td>
<td>1e+06</td>
<td>54052.00</td>
<td>1319.000</td>
</tr>
<tr>
<td>2</td>
<td>104286.7</td>
<td>2000.019</td>
<td>1.164108e-05</td>
<td>1.164108e-05</td>
<td>1e+06</td>
<td>53910.11</td>
<td>1442.447</td>
</tr>
<tr>
<td>3</td>
<td>104287.7</td>
<td>2000.038</td>
<td>1.162241e-05</td>
<td>1.162241e-05</td>
<td>1e+06</td>
<td>53692.54</td>
<td>1573.095</td>
</tr>
<tr>
<td>4</td>
<td>104288.7</td>
<td>2000.058</td>
<td>1.159158e-05</td>
<td>1.159158e-05</td>
<td>1e+06</td>
<td>53398.91</td>
<td>1708.214</td>
</tr>
<tr>
<td>5</td>
<td>104289.7</td>
<td>2000.077</td>
<td>1.154904e-05</td>
<td>1.154904e-05</td>
<td>1e+06</td>
<td>53031.36</td>
<td>1844.220</td>
</tr>
<tr>
<td>6</td>
<td>104290.7</td>
<td>2000.096</td>
<td>1.149543e-05</td>
<td>1.149543e-05</td>
<td>1e+06</td>
<td>52594.95</td>
<td>1976.772</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>Q</td>
<td>Z</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>944629.0</td>
<td>0.0000</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>944647.4</td>
<td>867.5997</td>
<td>867.5997</td>
<td>868</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>944734.4</td>
<td>1811.0116</td>
<td>943.4119</td>
<td>943</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>944892.9</td>
<td>2830.6905</td>
<td>1019.6789</td>
<td>1020</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>945124.4</td>
<td>3924.5379</td>
<td>1093.8474</td>
<td>1094</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>945428.3</td>
<td>5087.5473</td>
<td>1163.0094</td>
<td>1163</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The output contains the following information:

| \( t \) | \( \frac{t_k}{\Delta t} \) | Time in units of the observation interval \( \Delta t \). Equal to \( \frac{t_0}{\Delta t} + (0, 1, \ldots, n) \).
| \( t\_years \) | \( t_k \) | Time in years. Equal to \( t_0 + (0, 1, \ldots, n)\Delta t \).
| \( \beta \) | \( \beta(t_k)\Delta t \) | Seasonally forced transmission rate without environmental noise (Eq (5)), expressed per unit \( \Delta t \) per susceptible per infected.
| \( \beta\phi \) | \( \beta\phi(t_k)\Delta t \) | Seasonally forced transmission rate with environmental noise (Eq (2)), expressed per unit \( \Delta t \) per susceptible per infected.
| \( N \) | \( N(t_k) \) | Population size.
| \( S \) | \( S(t_k) \) | Number of susceptible individuals.
| \( I \) | \( I(t_k) \) | Number of infected individuals.
| \( R \) | \( R(t_k) \) | Number of removed individuals.
| \( Q \) | \( Q(t_k) \) | Cumulative incidence.
| \( Z \) | \( Z(t_k) \) | Incidence.
| \( C \) | \( C(t_k) \) | Reported incidence.

In this example, there is no environmental noise, simply because \( \text{par\_list} \) specified \( \epsilon = 0 \). Hence \( \beta \) and \( \beta\phi \) are identical in the returned data frame.

### S1.3 Estimating the time-varying transmission rate

We apply the FC, S, and SI methods to estimate incidence \( Z \), susceptibles \( S \), infecteds \( I \), and the seasonally forced transmission rate \( \beta \) from reported incidence and vital data. We have implemented these methods in our functions \( \text{estimate\_beta\_FC()} \), \( \text{estimate\_beta\_S()} \), and \( \text{estimate\_beta\_SI()} \). We will refer to these collectively as \( \text{estimate\_beta()} \), but note that there is no function by that name.

The first argument of \( \text{estimate\_beta()} \) expects a data frame \( \text{df} \) with columns \( t \), \( C \), \( B \), and (for the S and SI methods) \( \mu \). These specify equally spaced observation times \( t_k \) (Eq (3)) in units of the observation interval \( \Delta t \) (i.e., \( t_k/\Delta t \)) and, at those times, reported incidence \( C_k \), observed births \( B_k \), and the observed per capita natural mortality rate \( \mu_k \) expressed per unit \( \Delta t \).

The second argument expects a list \( \text{par\_list} \) with elements \( \text{prep}, \text{trep}, \text{tgen}, S_0 \), and (for the SI method) \( I_0 \). These specify values for input parameters \( p_{\text{rep}}, t_{\text{rep}}, t_{\text{gen}}, S_0 \), and \( I_0 \).
Here, we supply the data frame `df` obtained earlier using `make_data()`. Note that this data frame does not have columns `B` and `mu`. When these columns are missing in the function call, `estimate_beta()` creates mock (constant) time series as follows: (i) it looks in `par_list` for additional elements `hatN0`, `nu`, and `mu`, specifying a population size $\hat{N}_0$ and constant vital rates $\nu'_c$ and $\mu'_c$, then (ii) it sets $B_k = \nu'_c \hat{N}_0 \Delta t$ and $\mu_k = \mu'_c$ for all $k$.

In practice, the argument `par_list` contains the user’s potentially incorrect estimates of the input parameters, such as the initial number of susceptibles individuals $S_0$. For this example, there is no input error: we assign each input parameter its true (data-generating) value. That is, the `par_list` that we pass to `estimate_beta()` is precisely the `par_list` that we passed to `make_data()` earlier.

```r
## List of functions
estimate_beta <- list(
  FC = estimate_beta_FC,
  S = estimate_beta_S,
  SI = estimate_beta_SI
)

## List of returned data frames
df_est <- lapply(estimate_beta, function(f) f(df, par_list))
lapply(df_est, head, n = 10)

## $FC
##     t     C  Z Z_agg  B B_agg S   I  beta
## 1 104285.7 NA NA 767.1233 NA   767.1233 NA NA
## 2 104286.7 868 868 767.1233 NA   767.1233 NA NA
## 3 104287.7 943 943 1811.767.1233 1534.247 53775.06 1811 1.085364e-05
## 4 104288.7 1020 1020 767.1233 NA   767.1233 NA NA
## 5 104289.7 1094 1094 2114.767.1233 1534.247 52342.56 2114 1.034483e-05
## 6 104290.7 1163 1163 2387.767.1233 1534.247 51291.80 2387 1.006868e-05
## 7 104291.7 1224 1224 2585.767.1233 1534.247 51291.80 2585 1.006868e-05
## 8 104292.7 1274 1274 2585.767.1233 1534.247 51291.80 2585 1.006868e-05
## 9 104293.7 1311 1311 2585.767.1233 1534.247 51291.80 2585 1.006868e-05
## 10 104294.7 1332 1332 767.1233 NA   767.1233 NA NA

## $S
##     t     C  Z  B  mu S   I
## 1 104285.7 NA NA 0.0007671233 54051.82  NA
## 2 104286.7 868 868 767.1233 53909.48  NA
## 3 104287.7 943 943 767.1233 53692.25 1609.707
## 4 104288.7 1020 1020 767.1233 53398.18 1748.794
```
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>104289.7</td>
<td>1094</td>
<td>1094</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>53030.34</td>
<td>1891.591</td>
</tr>
<tr>
<td>6</td>
<td>104290.7</td>
<td>1163</td>
<td>1163</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>52593.78</td>
<td>2028.824</td>
</tr>
<tr>
<td>7</td>
<td>104291.7</td>
<td>1224</td>
<td>1224</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>52096.56</td>
<td>2156.784</td>
</tr>
<tr>
<td>8</td>
<td>104292.7</td>
<td>1274</td>
<td>1274</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>51549.72</td>
<td>2269.909</td>
</tr>
<tr>
<td>9</td>
<td>104293.7</td>
<td>1311</td>
<td>1311</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>50966.30</td>
<td>2362.634</td>
</tr>
<tr>
<td>10</td>
<td>104294.7</td>
<td>1332</td>
<td>1332</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>50362.32</td>
<td>2431.251</td>
</tr>
</tbody>
</table>

## beta

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

## $SI$

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>C</th>
<th>Z</th>
<th>B</th>
<th>mu</th>
<th>S</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>104285.7</td>
<td>NA</td>
<td>NA</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>54051.82</td>
<td>1318.586</td>
</tr>
<tr>
<td>2</td>
<td>104286.7</td>
<td>868</td>
<td>868</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>53909.53</td>
<td>1442.230</td>
</tr>
<tr>
<td>3</td>
<td>104287.7</td>
<td>943</td>
<td>943</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>53692.38</td>
<td>1572.434</td>
</tr>
<tr>
<td>4</td>
<td>104288.7</td>
<td>1020</td>
<td>1020</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>53398.43</td>
<td>1707.986</td>
</tr>
<tr>
<td>5</td>
<td>104289.7</td>
<td>1094</td>
<td>1094</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>53030.73</td>
<td>1844.252</td>
</tr>
<tr>
<td>6</td>
<td>104290.7</td>
<td>1163</td>
<td>1163</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>52593.78</td>
<td>1976.990</td>
</tr>
<tr>
<td>7</td>
<td>104291.7</td>
<td>1224</td>
<td>1224</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>52096.56</td>
<td>2101.398</td>
</tr>
<tr>
<td>8</td>
<td>104292.7</td>
<td>1274</td>
<td>1274</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>51549.72</td>
<td>2212.350</td>
</tr>
<tr>
<td>9</td>
<td>104293.7</td>
<td>1311</td>
<td>1311</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>50966.30</td>
<td>2305.321</td>
</tr>
<tr>
<td>10</td>
<td>104294.7</td>
<td>1332</td>
<td>1332</td>
<td>767.1233</td>
<td>0.0007671233</td>
<td>50362.32</td>
<td>2375.346</td>
</tr>
</tbody>
</table>

## beta

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>
The output contains the following information:

| t   | \frac{t_k}{\Delta t} | Time in units of the observation interval \Delta t. Equal to \frac{t}{\Delta t} + (0,1,\ldots,n). Identical to input df$t$.
| C   | C_k | Reported incidence. Identical to input df$C$, except that missing values and zeros (treated like missing values) have been imputed by linear interpolation.
| B   | B_k | Observed births. Identical to input df$B$ if supplied. Otherwise, every element is \texttt{with(par_list, hatN0 * nu * 1)}.
| mu | \mu_k \Delta t | Observed per capita natural mortality rate, expressed per unit \Delta t. Identical to input df$mu$ if supplied. Otherwise, every element is \texttt{with(par_list, mu)}.
| Z   | Z_k | Estimate of incidence \(Z(t_k)\).
| S   | S_k | Estimate of susceptibles \(S(t_k)\).
| I   | I_k | Estimate of infecteds \(I(t_k)\).
| beta | \beta_k \Delta t | Raw estimate of the time-varying transmission rate \(\beta(t_k)\) expressed per unit \(\Delta t\) per susceptible per infected.

Note that the FC method’s output has missing values in alternating rows. This is not a mistake: the FC method aggregates incidence and births over the mean generation interval (roughly 2 weeks in this example), instead of the observation interval (1 week). As a result, estimation of \(S(t_k)\) and \(\beta(t_k)\) is only possible at every other observation time. Aggregation is not a requirement of the S and SI methods.

Be warned that \texttt{estimate_beta()} returns raw estimates of \(\beta(t)\). Smoothing the \(\beta_k\) time series by fitting a loess curve \(\beta_{\text{loess}}(t; q)\) is recommended in the event that it displays unwanted noise. Determining what degree of smoothing is optimal \(i.e.,\) choosing a good value for the loess smoothing parameter \(q\) \(s\) is non-trivial, and therefore not undertaken by \texttt{estimate_beta()}. We discuss this issue in more detail in §S4.

Plotting the \(S_k\) and \(\beta_k\) time series returned by \texttt{estimate_beta()}, we reproduce Fig 1. Note that we are simply plotting (with a scaling) \(\text{df_est}$FC$S\), \(\text{df_est}$S$S\), and \(\text{df_est}$SI$S\), and separately \(\text{df_est}$FC$beta\), \(\text{df_est}$S$beta\), and \(\text{df_est}$SI$beta\).
S1.4 Measuring estimation error

In the manuscript, we report the relative root mean square error (RRMSE) in each $\beta_k$ time series. We compute RRMSE using `compute_rrmse()`, which takes as arguments the two vectors we wish to compare, with the estimate passed second.

```r
sapply(df_est, function(x) compute_rrmse(df$beta, x$beta))
```

## FC  S  SI
## 0.335487234 0.024010753 0.002139994
S2 Effect of process and observation error

Fig 1 in the manuscript shows the output of the S and SI methods for idealized reported incidence data. Those data were simulated deterministically, i.e.,

- without environmental stochasticity [ES] (par_list with epsilon = 0 passed to make_data()),
- without demographic stochasticity [DS] (with_dem_stoch = FALSE passed to make_data()), and
- without observation error [OE] (par_list with prep = 1 passed to make_data()).

In Fig 2, we consider the effect of adding ES, DS, and OE in turn to the original deterministic simulation, on the $\beta_k$ time series generated by the S and SI methods. We produce these four simulations as follows:

```r
## List of lists of parameter values
par_list <- list(
  xxxxxx = make_par_list(epsilon = 0, prep = 1),  # deterministic
  esxxxx = make_par_list(epsilon = 0.5, prep = 1),  # ES
  esdsxx = make_par_list(epsilon = 0.5, prep = 1),  # ES+DE
  esdsoe = make_par_list(epsilon = 0.5, prep = 0.25)  # ES+DE+OE
)

## List of data frames containing time series data
df <- mapply(make_data,
  par_list = par_list,
  n = 3 * 365 / 7,
  with_dem_stoch = c(FALSE, FALSE, TRUE, TRUE),
  seed = 1305,
  SIMPLIFY = FALSE
)

names(df)
## [1] "xxxxxxx" "esxxxx" "esdsxx" "esdsoe"

names(df$esdsoe)
## [1] "t" "t_years" "beta" "beta_phi" "N" "S"
## [7] "I" "R" "Q" "Z" "C"
```
Here, \texttt{par_list} is a \textit{list of lists}, containing the parameter values desired for each simulation, and \texttt{df} is a list of data frames, containing the corresponding simulated data.

The next code chunk applies the S and SI methods, without input error, to each simulated reported incidence time series.

```r
## List of lists of data frames containing estimation output
df_est <- list(  
  ## List of data frames returned by S method
  S = mapply(estimate_beta_S,  
    df = df,  
    par_list = par_list,  
    SIMPLIFY = FALSE  
  ),  
  ## List of data frames returned by SI method
  SI = mapply(estimate_beta_SI,  
    df = df,  
    par_list = par_list,  
    SIMPLIFY = FALSE  
  )  
)
names(df_est)
## [1] "S" "SI"

names(df_est$SI)
## [1] "xxxxxx" "esxxxx" "esdxxx" "esdsoe"

names(df_est$SI$esdsoe)
## [1] "t" "C" "Z" "B" "mu" "S" "I" "beta"
```

Hence each element of \texttt{df_est} contains the output of one of the S and SI methods for all 4 reported incidence time series. Plotting the resulting 8 $Z_k$, $I_k$, and $\beta_k$ time series (estimates of incidence, prevalence, and the seasonally forced transmission rate) yields Fig 2.
We obtain the RRMSE in each $\beta_k$ time series with the following line of code.

```r
lapply(df_est, function(x) {
  mapply(function(y, z) compute_rrmse(y$beta, z$beta), y = df, z = x)
})
```

## $S$

<table>
<thead>
<tr>
<th>xxxxxx</th>
<th>esxxxx</th>
<th>esdsxx</th>
<th>esdsoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02393485</td>
<td>0.03352897</td>
<td>0.12261568</td>
<td>0.15458400</td>
</tr>
</tbody>
</table>

## $SI$

<table>
<thead>
<tr>
<th>xxxxxx</th>
<th>esxxxx</th>
<th>esdsxx</th>
<th>esdsoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.002145247</td>
<td>0.015053954</td>
<td>0.053576843</td>
<td>0.070091436</td>
</tr>
</tbody>
</table>
Fig 3 in the manuscript considers the seasonally forced $\beta(t)$ (Eq (5)) and three estimates, each spanning 1000 years. It overlays the 1000 1-year cycles embedded in each estimate and plots their 1-year average. To reproduce Fig 3, we simulate 1000 years of weekly reported incidence, including in the simulation environmental noise in transmission ($\epsilon = 0.5$), demographic stochasticity, and random under-reporting of cases ($p_{\text{rep}} = 0.25$).

```r
## List of parameter values
par_list <- make_par_list(epsilon = 0.5, prep = 0.25)

## Data frame containing time series data
df <- make_data(par_list = par_list, n = 1000 * 365 / 7 + 1, with_dem_stoch = TRUE, seed = 1217)
```

We estimate the seasonally forced $\beta(t)$ using the S and SI methods, without input error.

```r
## List of functions
estimate_beta <- list(S = estimate_beta_S, SI = estimate_beta_SI)

## List of returned data frames. Column \texttt{\beta} in \texttt{df_est[[i]]} stores the raw estimate generated by \texttt{estimate_beta[[i]]}.
df_est <- lapply(estimate_beta, function(f) f(df, par_list))
```

The raw time series estimates $\beta_k$ contain spurious noise due to process and observation error. Hence, for comparison, we fit a loess curve $\beta_{\text{loess}}(t; q)$ to the time series returned by the SI method, where $q$ is the loess smoothing parameter indicating (roughly) the number of nearest neighbours weighted in local regression. It turns out that $q = 53$ is a good choice for this parameter in this example (see §S4). To do the fitting, we call `loess()` with argument `span` indicating $q$ as a proportion of time series length. Additional arguments are fully explained in the documentation, accessible by running `?loess` and `?loess.control`.

S3 Averaging the raw estimate of $\beta(t)$
## Object of class `loess` defining the fitted loess curve

```r
SI_loess <- stats::loess(
  formula = beta ~ t,
  data = df_est$SI,
  span = 53 / nrow(df_est$SI),
  degree = 2,
  na.action = "na.exclude",
  control = loess.control(surface = "direct")
)
```

We will calculate the average 1-year cycle in the linear interpolant $\beta_{\text{int}}(t)$ of $\beta_k$ (S and SI methods), and in the loess curve $\beta_{\text{loess}}(t; q)$ fit to the same time series (SI method only).

## List of interpolants. `fits[[i]]` is the interpolant of column `beta` in `df_est[[i]]`.

```r
fits <- lapply(df_est, function(x) {
  approxfun(x$t, x$beta, method = "linear", rule = 1)
})
```

## Appending the `loess` object from earlier

```r
fits$SI_loess <- SI_loess
```

Before proceeding, we should verify that the $\beta_k$ time series contain 1000 1-year cycles.

```r
## First and last time points
t0 <- df_est$SI$t[1]
tn <- df_est$SI$t[nrow(df_est$SI)]
## 1-year period in units of the observation interval
period <- with(par_list, (365 / 7) / dt_weeks)
## Number of cycles
m <- floor((tn - t0) / period)
m
## [1] 1000
```

We must also specify the true 1-year cycle that each average 1-year cycle should estimate. We specify the true cycle with a reference time indicating the initial phase. For simplicity, we consider the cycle between times $t_0$ and $t_0 + \text{period}$ to be the true cycle. Phases of this cycle are specified by times $s$ between 0 and $\text{period}$. To estimate the value of the true cycle at phase $s$, we evaluate each fit in `fits` (2 linear interpolants and 1 loess
curve) at times \( t_0 + s + (0:(m-1)) \times \text{period} \) and average the resulting \( m \) values.

\[ \text{get_phase_average() \ computes this estimate for any \( s \), for a given fit \( f \).} \]

```
get_phase_average <- function(s, f) {
  times <- t0 + (s %% period) + (0:(m-1)) \times \text{period}
  x <- \text{if (inherits(f, "loess")) predict(f, times) else f(times)}
  mean(x, na.rm = TRUE)
}
```

Note that, whereas linear interpolants are just functions that we can evaluate at desired times, loess objects must be passed to \texttt{predict()} to obtain fitted values. Note also that the modulo operation \( s \ %\% \text{period} \) makes \texttt{get_phase_average()} a periodic function of \( s \).

We construct the average 1-year cycle in each fit in \texttt{fits} by applying \texttt{get_phase_average()} on a desired grid of \( s \) values.

```
s_grid <- seq(0, period, length.out = 150)
average_one_year <- data.frame(
  s_grid,
  lapply(fits, function(f) sapply(s_grid, get_phase_average, f = f))
)
head(average_one_year)

## s_grid S SI SI_loess
## 1 0.0000000 1.193014e-05 1.153366e-05 1.148309e-05
## 2 0.3499521 1.194339e-05 1.154372e-05 1.148619e-05
## 3 0.6999041 1.193943e-05 1.155337e-05 1.148787e-05
## 4 1.0498562 1.191978e-05 1.155581e-05 1.148843e-05
## 5 1.3998082 1.190497e-05 1.155318e-05 1.148760e-05
## 6 1.7497603 1.189256e-05 1.154800e-05 1.148552e-05
```

We reproduce Fig 3 by plotting, for each 1000-year estimate of \( \beta(t) \), the 1000 embedded 1-year cycles and their 1-year average, all on the same 1-year axis.
Fig 3. Bias and variance in 1-year cycles embedded in three estimates of a seasonally forced $\beta(t)$, in the absence of input error.

S4 Smoothing the raw estimate of $\beta(t)$

Figs 2C, 3B, and 3C demonstrate that, in the absence of input error, much of the error in raw estimates of $\beta(t)$ is attributable to noise in the time series. The S and SI methods produce the correct temporal pattern, but it is distorted by noise. Comparison of Figs 3C and 3D shows that fitting a smooth loess curve $\beta_{\text{loess}}(t; q)$ to the raw time series estimate $\beta_k$ can lead to substantial improvement in both accuracy and interpretability.

In practice, we would like to choose the value $q_{\text{opt}}$ of the loess smoothing parameter $q$ that minimizes the error in $\beta_{\text{loess}}(t; q)$ relative to $\beta(t)$. However, we cannot calculate (and therefore cannot minimize) the error in $\beta_{\text{loess}}(t; q)$ when $\beta(t)$ is not known. In this situation, we can still estimate $q_{\text{opt}}$ using statistical methods (most notably time series cross-validation [2]) or by direct inspection of $\beta_{\text{loess}}(t; q)$ for each value of $q$ on a grid.

In our simulated data setting, we do know $\beta(t)$ and can therefore determine $q_{\text{opt}}$ exactly. In this setting, it is instructive to quantify the reduction in error that is achieved
when the optimal loess estimate $\beta_{\text{loess}}(t; q_{\text{opt}})$ is chosen over the raw time series estimate $\beta_k$. Fig 4 in the manuscript addresses this issue.

To reproduce Fig 4, we consider 41 values of the case reporting probability $p_{\text{rep}}$ and fix all other data-generating parameters. (Smaller values of $p_{\text{rep}}$ generate noisier reported incidence time series, resulting in noisier $\beta_k$.) Using each value of $p_{\text{rep}}$, we perform 100 simulations of reported incidence.

```r
prep <- 10^seq(-2, 0, length.out = 41)
par_list <- make_par_list(epsilon = 0.5)
nsim <- 100
q <- 10:110
```

For each of these $41 \times 100$ simulations, we carry out the following steps. We estimate the seasonally forced $\beta(t)$ (Eq (5)) from the simulated reported incidence time series, without input error. (The underlying $\beta(t)$ was the same across all simulations.) For each $q$ between 10 and 110, we fit a loess curve $\beta_{\text{loess}}(t; q)$ to the raw time series estimate $\beta_k$. Finally, we record

$$\text{RRMSE}_{\text{raw}} = \text{RRMSE in } \{(t_k, \beta_k)\}_{k=0}^n,$$
$$\text{RRMSE}_{\text{loess,min}} = \min_{q \in \{10, \ldots, 110\}} \left[ \text{RRMSE in } \{(t_k, \beta_{\text{loess}}(t_k; q))\}_{k=0}^n \right],$$
$$q_{\text{opt}} = \arg \min_{q \in \{10, \ldots, 110\}} \left[ \text{RRMSE in } \{(t_k, \beta_{\text{loess}}(t_k; q))\}_{k=0}^n \right].$$

Hence, for each value of $p_{\text{rep}}$, we obtain 100 values for each of $\text{RRMSE}_{\text{raw}}$, $\text{RRMSE}_{\text{loess,min}}$, and $q_{\text{opt}}$. We can preallocate space for this output.

```r
out <- array(NA, 
  dim = c(length(prep), nsim, 3, 2),
  dimnames = list(NULL, NULL, 
  c("rrmse_raw", "rrmse_loess_min", "qopt"),
  c("S", "SI")
)
```

The fourth dimension of the output array enables our comparison of the distributions of $\text{RRMSE}_{\text{raw}}$, $\text{RRMSE}_{\text{loess,min}}$, and $q_{\text{opt}}$ for different methods of $\beta(t)$ estimation—in this case, the S and SI methods. Our expectation based on Figs 2C, 3B, and 3C is that the S method requires more smoothing.

The next code chunk does all of the hard work. Simulations are saved in the directory RData/loess/. Main results are saved in the file RData/loess.RData.
for (i in seq_along(prep)) {

## Update `par_list` with current value of `prep`
par_list$prep <- prep[i]

## Create a directory for this loop's `.RData`
dirname <- paste0(
  "RData/loess/",
  ## log10 current value of `prep`
  "prep_log10v-", sprintf("%+05.0f", log(prep[i], 10) * 1000), "/"
)
if (!dir.exists(dirname)) {
  dir.create(dirname, recursive = TRUE)
}

for (j in seq_len(nsim)) {

  message(
    "prep value ", i, " of ", length(prep), "",
    "sim ", j, " of ", nsim
  )

  ## File name for simulation
  filename <- paste0(dirname, "sim", sprintf("%04.0f", j), ".RData")

  ## Simulate reported incidence data, if you haven't already
  if (file.exists(filename)) {
    load(filename)
  } else {
    df <- make_data(
      par_list = par_list,
      n = 20 * 365 / 7,
      with_dem_stoch = TRUE,
      seed = j
    )
    save(df, file = filename)
  }

  ## Estimate the seasonally forced transmission rate from
  ## reported incidence
  estimate_beta <- list(

S = estimate_beta_S,
SI = estimate_beta_SI
)
df_est <- lapply(estimate_beta, function(f) f(df, par_list))

## Compute the error in each raw estimate
out[i, j, "rrmse_raw", ] <- sapply(df_est, function(x) {
  compute_rrmse(df$beta, x$beta)
})

## Preallocate memory for the error in each loess estimate
## (one for each value of `q`, for each raw estimate)
rrmse_after_loess <- array(NA, dim = c(length(q), 2),
                           dimnames = list(NULL, c("S", "SI")))

## Standardize missing values in the raw estimates. `loess()`
## handles `NA` but complains about `NaN` and `Inf`.
df_est <- lapply(df_est, function(x) {
  x$beta[!is.finite(x$beta)] <- NA
  x
})

for (k in seq_along(q)) {

  ## Fit a smooth loess curve to each raw estimate
  loess_fit <- lapply(df_est, function(x) {
    stats::loess(
      formula = beta ~ t,
      data = x,
      span = q[k] / nrow(x),
      degree = 2,
      na.action = "na.exclude",
      control = loess.control(
        surface = "direct",
        statistics = "none"
      )
    )
  })
}
## Compute the error in each loess estimate

```r
rrmse_after_loess[k, ] <- sapply(loess_fit, function(x) {
    compute_rrmse(df$beta, predict(x))
})
```

## Find the minimum error over all loess estimates.

## Also retrieve the value of `q` that minimized error.

```r
out[i, j, "rrmse_loess_min", ] <- apply(rrmse_after_loess, 2, min)
out[i, j, "qopt", ] <- apply(rrmse_after_loess, 2, function(x) {
    q[which.min(x)]
})
```

```r
attr(out, "arg_list") <- list(
    prep = prep,
    par_list = par_list,
    q = q
)
save(out, file = "RData/loess.RData")
```

For each value of \( p_{\text{rep}} \), we desire the median and 5th and 95th percentiles of \( \text{RRMSE}_{\text{raw}} \), \( \text{RRMSE}_{\text{loess.min}} \), and \( q_{\text{opt}} \).  

```r
pct <- apply(out, c(1, 3, 4), quantile, probs = c(0.05, 0.5, 0.95))
```

Plotting these as functions of \( p_{\text{rep}} \), we reproduce Fig 3.
In the manuscript, we report the minimum percentage reduction in RRMSE across all simulations. Borrowing our earlier notation, this is the minimum value of

$$100 \times \left(1 - \frac{\text{RRMSE}_{\text{loess,min}}}{\text{RRMSE}_{\text{raw}}}\right).$$

We also report the median value of $q_{\text{opt}}$ across all simulations for which $p_{\text{rep}} \geq 0.1$. 

```r
apply(out, 4, function(x) {
  min(100 * (1 - x[, , "rrmse_loess_min"] / x[, , "rrmse_raw"]), na.rm = TRUE)
})
n
# S  SI
# 46.38507 17.47202
```
In our remaining analysis, we set $q = q^*$ when smoothing any $\beta_k$ time series, taking

$$q^* = \begin{cases} 65 & \text{with the S method,} \\ 53 & \text{with the SI method.} \end{cases}$$  

(7)

For a given time series, this setting may not be optimal ($q^* \neq q_{opt}$), but can be justified, as we explain in the sections to follow.

### S5 Sensitivity to data-generating parameters

Error in estimates of the seasonally forced $\beta(t)$ (Eq (5)) from simulated reported incidence data is a function of data-generating parameters, given by

$$\theta = (\langle \beta \rangle, \alpha, \epsilon, \hat{N}_0, S_0, I_0, \nu_c, \mu_c, t_{gen}, p_{rep}, t_{rep}, t_0, \Delta t, n).$$

(8)

In order to measure the sensitivity of $\beta(t)$ estimation error to $\theta$, we must define grids of parameter values. For this task, it is helpful to associate with each parameter a reference value:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle \beta \rangle$</td>
<td>$\beta^*$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.08</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\hat{N}_0$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>$S_0$</td>
<td>$S^*$</td>
</tr>
<tr>
<td>$I_0$</td>
<td>$I^*$</td>
</tr>
<tr>
<td>$\nu_c$</td>
<td>0.04 year$^{-1}$</td>
</tr>
<tr>
<td>$\mu_c$</td>
<td>0.04 year$^{-1}$</td>
</tr>
<tr>
<td>$t_{gen}$</td>
<td>13 days</td>
</tr>
<tr>
<td>$p_{rep}$</td>
<td>$p^*_{rep}$</td>
</tr>
<tr>
<td>$t_{rep}$</td>
<td>$t^*_{rep}$</td>
</tr>
<tr>
<td>$t_0$</td>
<td>2000 years</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>1 week</td>
</tr>
<tr>
<td>$n$</td>
<td>1042</td>
</tr>
</tbody>
</table>

Here, $S^*$ and $I^*$ are the values of $S$ and $I$ at a point very near the attractor of system (1) with constant vital rates $\nu_c$ and $\mu_c$ and a seasonally forced transmission rate (Eq (5)). It follows that $S^*$ and $I^*$ vary with parameters of the system (specifically, $\langle \beta \rangle$, $\alpha$, $\nu_c$, $\mu_c$, and $t_{gen}$). See §S1.1 for details on how $S^*$ and $I^*$ are obtained using `make_par_list()` given values for other parameters.

$\beta^*$ is the value of $\langle \beta \rangle$ that satisfies Eq (4) with $R_0 = 20$, $\hat{N}_0 = 10^6$, $\nu_c = \mu_c = 0.04$ year$^{-1}$, and $t_{gen} = \gamma^{-1} = 13$ days:

$$\beta^* = 20 \cdot \frac{0.04 \text{year}^{-1}}{10^6 \cdot 0.04 \text{year}^{-1}} \cdot \left( \frac{1}{13} \text{day}^{-1} + 0.04 \text{year}^{-1} \right)$$

$$\approx 5.6234 \times 10^{-4} \text{year}^{-1}.$$
Finally, \( p_{\text{rep}}^* = \begin{cases} 1 & \text{for analysis without OE} \\ 0.25 & \text{for analysis with OE} \end{cases} \) and \( t_{\text{rep}}^* = \begin{cases} 0 \text{ weeks for analysis without OE} \\ 2 \text{ weeks for analysis with OE} \end{cases} \) where an “analysis with OE” is one in which we desire simulations of reported incidence with observation error.

### S5.1 Sensitivity to \( \mathcal{R}_0 \) and \( \alpha \)

Fig 5 in the manuscript describes how \( \beta(t) \) estimation error depends on features of \( \beta(t) \) itself. For the seasonally forced \( \beta(t) \) (Eq (5)), these features are the mean \( \langle \beta \rangle \) and amplitude \( \alpha \). Fig 5 casts error as a function of \( \mathcal{R}_0 \) and \( \alpha \), rather than \( \langle \beta \rangle \) and \( \alpha \), but this formulation is equivalent, because \( \mathcal{R}_0 \) is proportional to \( \langle \beta \rangle \) (Eq (4)). It is also more interpretable: unlike \( \langle \beta \rangle \), \( \mathcal{R}_0 \) has a natural epidemiological meaning and is dimensionless (its numerical value does not depend on the chosen units of time).

To reproduce Fig 5, we set all data-generating parameters other than \( \langle \beta \rangle \) and \( \alpha \) equal to their reference value in (9). We consider the grid of \((\mathcal{R}_0, \alpha)\) pairs with levels \( \mathcal{R}_0 = 2, 3, \ldots, 32 \) and \( \alpha = 0, 0.01, \ldots, 0.2 \)—defining \( \langle \beta \rangle \) for each \( \mathcal{R}_0 \) using Eq (4)—and simulate 1000 reported incidence time series using each of these \( 31 \times 21 \) parametrizations.

```r
Rnaught <- seq(2, 32, by = 1)
alpha <- seq(0, 0.2, by = 0.01)
nsim <- 1000
```

We estimate \( \beta(t) \) from each simulated reported incidence time series, without input error, fit a loess curve \( \beta_{\text{loess}}(t; q) \) to the raw estimate \( \beta_k \), and record the RRMSE in \( \beta_{\text{loess}}(t_k; q) \). For comparison, this is done using both the S and SI methods. We fix \( q = q^* \) (Eq (7)) independently of the \( \beta_k \) time series being smoothed. (See §S5.3 for discussion of the consequences of using fixed \( q \) in this analysis.)

This algorithm is implemented in our function `test_s2dgbeta()` (“sensitivity to data-generating \( \beta(t) \)”), which takes as arguments

- `par_list_ref`, a list containing the reference values of all data-generating parameters;
- `Rnaught` and `alpha`, numeric vectors specifying the desired grid of \((\mathcal{R}_0, \alpha)\) pairs;
- `with_dem_stoch`, a logical scalar indicating whether simulations should account for demographic stochasticity;
- `nsim`, the number of simulations to perform using each \((\mathcal{R}_0, \alpha)\) pair;
loess_par, a numeric vector of length 2 specifying the value of the loess smoothing parameter \( q \) used when fitting loess curves to raw transmission rate estimates \( \beta_k \). \( \text{loess\_par[1]} \) is used with the S method. \( \text{loess\_par[2]} \) is used with the SI method.

\[ s2dgbeta() \] returns a 4-dimensional array, whose \([i, j, k, m]\) th entry is the RRMSE in an estimate of \( \beta(t) \) (S method if \( m = 1 \), SI method if \( m = 2 \)) from simulation \( k \) of \( \text{nsim} \) using the \((i,j)\)th \((R_0, \alpha)\) grid point.

First, we consider simulations with environmental stochasticity \((\epsilon = 0.5)\), without demographic stochasticity, and without observation error \((p_{\text{rep}} = 1, t_{\text{rep}} = 0)\).

```r
rmse_esxxxx <- test_s2dgbeta(
  par_list_ref = make_par_list(epsilon = 0.5, prep = 1, trep = 0),
  Rnaught = Rnaught,
  alpha = alpha,
  with_dem_stoch = FALSE,
  nsim = nsim,
  loess_par = qstar
)
save(rmse_esxxxx, file = "RData/s2dgbeta_esxxxx.RData")
```

Second, we consider simulations with environmental stochasticity \((\epsilon = 0.5)\), with demographic stochasticity, and without observation error \((p_{\text{rep}} = 1, t_{\text{rep}} = 0)\).

```r
rmse_esdsxx <- test_s2dgbeta(
  par_list_ref = make_par_list(epsilon = 0.5, prep = 1, trep = 0),
  Rnaught = Rnaught,
  alpha = alpha,
  with_dem_stoch = TRUE,
  nsim = nsim,
  loess_par = qstar
)
save(rmse_esdsxx, file = "RData/s2dgbeta_esdsxx.RData")
```

Third, we consider simulations with environmental stochasticity \((\epsilon = 0.5)\), with demographic stochasticity, and with observation error \((p_{\text{rep}} = 0.25, t_{\text{rep}} = 2 \text{ weeks})\).

```r
rmse_esdsoe <- test_s2dgbeta(
  par_list_ref = make_par_list(epsilon = 0.5, prep = 0.25, trep = 2),
  Rnaught = Rnaught,
  alpha = alpha,
  with_dem_stoch = TRUE,
)
nsim = nsim,
loess_par = qstar
)
save(rrmse_esdsoe, file = "RData/s2dgbeta_esdsoe.RData")

We apply `get_rrmse_50pct()` to the output of `test_s2dgbeta()` in order to compute the median RRMSE in each set of `nsim` estimates of $\beta(t)$.

`get_rrmse_50pct()` takes as arguments `rrmse` (any of the three arrays defined earlier) and `method` ("S" or "SI"). It returns an array whose $[i, j]$th entry is the median RRMSE for the $(i, j)$th $(R_0, \alpha)$ grid point.

Fig 5 displays heatmaps of median RRMSE obtained from the output of `get_rrmse_50pct()`. There is one heatmap for each choice of the arguments `rrmse` and `method` ($3 \times 2$ heatmaps in total). Navy fill indicates median RRMSE less than the minimum shown on the colour scale.
Fig 5. Sensitivity of $\beta(t)$ estimation error to the mean $\langle \beta \rangle$ and amplitude $\alpha$ of seasonal forcing. The $\langle \beta \rangle$ axis has been scaled to measure the basic reproduction number $R_0$ (Eq (4)).
S5.2 Sensitivity to $S_0$, $I_0$, $\nu_c$, $\mu_c$, $t_{\text{gen}}$, and $p_{\text{rep}}$

Fig 6 describes how $\beta(t)$ estimation error varies as a function of data-generating parameters other than $\langle \beta \rangle$ and $\alpha$: the initial states $S_0$ and $I_0$, vital rates $\nu_c$ and $\mu_c$, mean generation interval $t_{\text{gen}}$, and case reporting probability $p_{\text{rep}}$.

To reproduce Fig 6, we explore lines in the data-generating parameter space by assigning all parameters their reference value in (9), except a focal parameter (one of $S_0$, $I_0$, $\nu_c$, $\mu_c$, $t_{\text{gen}}$, and $p_{\text{rep}}$), which we assign each of 41 values logarithmically spaced between $\frac{1}{4}$ and 4 times its reference value. Using each of these $5 \times 41$ or $6 \times 41$ parametrizations (we fix $p_{\text{rep}} = 1$ when we desire simulations without observation error), we simulate 1000 reported incidence time series.

```r
scale_factors <- 2^seq(-2, 2, length.out = 41)
nsim <- 1000
```

We estimate $\beta(t)$ from each simulated reported incidence time series, without input error, fit a loess curve $\beta_{\text{loess}}(t; q)$ to the raw estimate $\beta_k$, and record the RRMSE in $\beta_{\text{loess}}(t_k; q)$. For comparison, this is done using both the S and SI methods. We fix $q = q^*$ (Eq (7)) independently of the $\beta_k$ time series being smoothed. (See §S5.3 for discussion of the consequences of using fixed $q$ in this analysis.)

This algorithm is implemented in our function `test_s2dgpars()` ("sensitivity to data-generating parameters"). Its arguments are identical to those of `test_s2dgpars()`, except, instead of $R_{0\text{aught}}$ and $\alpha$, `test_s2dgpars()` has arguments

- `pars_to_vary`, a character vector listing the data-generating parameters to be treated as a focal parameter;
- `scale_factors`, a numeric vector listing the factors by which the reference value of each focal parameter is scaled to obtain the grid of values considered for that parameter.

`test_s2dgpars()` returns a 4-dimensional array, whose $[i, j, k, m]$ th entry is the RRMSE in an estimate of $\beta(t)$ (S method if $m = 1$, SI method if $m = 2$) from simulation $k$ of $nsim$ in which $\text{pars_to_vary}[j]$ is assigned its reference value times $\text{scale_factors}[i]$.

First, we consider simulations with environmental stochasticity ($\epsilon = 0.5$), without demographic stochasticity, and without observation error ($p_{\text{rep}} = 1$, $t_{\text{rep}} = 0$ weeks).

```r
rrmse_esxxxx <- test_s2dgpars(
pars_to_vary = c("S0", "I0", "nu", "mu", "tgen"),
par_list_ref = make_par_list(epsilon = 0.5, prep = 1, trep = 0),
scale_factors = scale_factors,
```
Second, we consider simulations with environmental stochasticity ($\epsilon = 0.5$), with demographic stochasticity, and without observation error ($p_{rep} = 1$, $t_{rep} = 0$ weeks).

```r
rrmse_esdsxx <- test_s2dgpars(
  pars_to_vary = c("S0", "I0", "nu", "mu", "tgen"),
  par_list_ref = make_par_list(epsilon = 0.5, prep = 1, trep = 0),
  scale_factors = scale_factors,
  with_dem_stoch = TRUE,
  nsim = nsim,
  loess_par = qstar
)
save(rrmse_esdsxx, file = "RData/s2dgpars_esdsxx.RData")
```

Third, we consider simulations with environmental stochasticity ($\epsilon = 0.5$), with demographic stochasticity, and with observation error ($p_{rep} = 0.25$, $t_{rep} = 2$ weeks).

```r
rrmse_esdsoe <- test_s2dgpars(
  pars_to_vary = c("S0", "I0", "nu", "mu", "tgen", "prep"),
  par_list_ref = make_par_list(prep = 0.25, trep = 2),
  scale_factors = scale_factors,
  with_dem_stoch = TRUE,
  nsim = nsim,
  loess_par = qstar
)
save(rrmse_esdsoe, file = "RData/s2dgpars_esdsoe.RData")
```

In this third analysis, when $S_0$, $I_0$, $\nu_c$, $\mu_c$, or $t_{gen}$ is varied, $p_{rep}$ is fixed and assigned its reference value, 0.25. When $p_{rep}$ itself is varied, we consider for $p_{rep}$ each value in the vector `scale_factors * 0.25`.

As with `test_s2dgbeta()`, we apply `get_rrmse_50pct()` to the output of `test_s2dgpars()` in order to compute the median RRMSE in each set of $\text{nsim}$ estimates of $\beta(t)$.

Fig 6 displays the output of `get_rrmse_50pct()`, plotting median RRMSE as a function of each data-generating parameter. To be precise, the horizontal axis measures the ratio of the data-generating and reference values of the focal parameter, which ranges from 28.
$\frac{1}{4}$ to 4 regardless of the focal parameter. This allows results for different parameters to be compared in one panel.

Fig 6. Sensitivity of $\beta(t)$ estimation error to data-generating parameters other than $\langle \beta \rangle$ and $\alpha$. 
S5.3 A note on smoothing

To generate Figs 5 and 6, we fixed \( q = q^* \) (Eq (7)) when fitting loess curves \( \beta_{\text{loess}}(t; q) \) to raw transmission rate estimates \( \beta_k \). For a given \( \beta_k \) time series, this setting may not have been optimal (\( q^* \neq q_{\text{opt}} \)), meaning that the RRMSE calculated for \( \beta_{\text{loess}}(t_k; q) \) was greater with \( q = q^* \) than it would have been had we found \( q_{\text{opt}} \) and used \( q = q_{\text{opt}} \).

This is potentially problematic, because sensitivity to data-generating parameters is mediated by propagation of noise from the simulated reported incidence data to \( \beta_k \). We may have observed less sensitivity to a parameter (for example, \( t_{\text{gen}} \) in Fig 6) had we smoothed more when there was extreme noise in \( \beta_k \) (i.e., had we set \( q = q_{\text{opt}} \) when \( q_{\text{opt}} > q^* \)). We did not do this, because finding \( q_{\text{opt}} \) for each of the \( 5 \times 10^6 \) time series considered by Figs 5 and 6 would have increased the total computation time by a factor of 100. Hence Figs 5 and 6 may overestimate the sensitivity of \( \beta(t) \) estimation error to data-generating parameters.

Nevertheless, we expect the quantitative effect of choosing \( q^* \) over \( q_{\text{opt}} \) to be relatively small. Consider the graph corresponding to \( t_{\text{gen}} \) in the right panel of Fig 6C, which displays median RRMSE close to \((0.12, 0.03, 0.045)\) when \( t_{\text{gen}} \) is \((2^{-1.5}, 1, 2^{1.5}) \cdot 13 \) days, respectively (13 days being the reference value). For these values of \( t_{\text{gen}} \), it is instructive to compare (i) simulated reported incidence time series \( C_k \), (ii) raw transmission rate estimates \( \beta_k \) from \( C_k \), and (iii) the corresponding loess estimates \( \beta_{\text{loess}}(t; q^*) \) and \( \beta_{\text{loess}}(t; q_{\text{opt}}) \).

S5.3.1 \( C_k \) and \( \beta_k \) for \( t_{\text{gen}} = (2^{-1.5}, 1, 2^{1.5}) \cdot 13 \) days

First, we simulate a reported incidence time series \( C_k \) using each of \( t_{\text{gen}} = (2^{-1.5}, 1, 2^{1.5}) \cdot 13 \) days. All three simulations account for environmental stochasticity (\( \epsilon = 0.5 \)), demographic stochasticity, and observation error (\( p_{\text{rep}} = 0.25 \)).

```r
## List of reference parameter values
par_list_ref <- make_par_list(epsilon = 0.5, prep = 0.25)

## List of lists of data-generating parameter values
par_list <- mapply(make_par_list,
    tgen = 2^c(-1.5, 0, 1.5) * par_list_ref$tgen,
    beta_mean = par_list_ref$beta_mean,
    Rnaught = NA,
    epsilon = 0.5,
    prep = 0.25,
    SIMPLIFY = FALSE)

## List of data frames containing time series data
df <- mapply(make_data,
    par_list)
```
par_list = par_list,
  n = 20 * 365 / 7,
  with_dem_stoch = TRUE,
  seed = c(1836, 6183, 3618),
  SIMPLIFY = FALSE)

Plotting each $C_k$ time series yields the following result. Note that we are simply plotting $\text{df[[i]]$C}$ for $i = 1, 2, 3$.

![Graph](image)

We see that a period-doubling bifurcation occurs between $t_{gen} = 13$ days and $t_{gen} = 2^{-1.5} \cdot 13$ days, with $C_k$ attaining a much smaller minimum in the time series with a 2-year cycle (generated by $t_{gen} = 2^{-1.5} \cdot 13$ days).

Due to much closer approaches to zero by incidence and prevalence with $t_{gen} = 2^{-1.5} \cdot 13$ days, noise in $C_k$ is amplified to a much greater extent in the raw transmission rate estimate $\beta_k$. We show this by applying the SI method without input error to estimate the underlying, seasonally forced transmission rate $\beta(t)$ (Eq (5))—which was the same across simulations—from each $C_k$ time series.

```r
## List of data frames containing estimation output
df_est <- mapply(estimate_beta_SI,
  df = df,
  par_list = par_list,
  SIMPLIFY = FALSE)
```

Plotting $\beta_k$ shows that, indeed, propagation of noise from $C_k$ to $\beta_k$ is much more severe when $t_{gen} = 2^{-1.5} \cdot 13$ days. Note that we are simply plotting $\text{df_est[[i]]$beta}$, scaled by $\text{with(par_list, 1/beta_mean)}$, for $i = 1, 2, 3$. 

31
We calculate the RRMSE in each of these estimates as follows.

```r
rrmse_raw <- mapply(
    function(x, y) compute_rrmse(x$beta, y$beta),
    x = df,
    y = df_est
)
rrmse_raw
## [1] 0.26917391 0.06715921 0.05893578
```

### S5.3.2 $\beta_{\text{loess}}(t; q^*)$ and $\beta_{\text{loess}}(t; q_{\text{opt}})$ for $t_{\text{gen}} = (2^{-1.5}, 1, 2^{1.5}) \cdot 13$ days

All three of these estimates of $\beta(t)$ are greatly improved with loess smoothing. First, we consider smoothing all three $\beta_k$ time series with $q = q^*$.

```r
loess_fit <- lapply(df_est,
    function(x) {
        loess(
            formula = beta ~ t,
            data = x,
            span = qstar["SI"] / nrow(x),
            degree = 2,
            na.action = "na.exclude",
            control = loess.control(surface = "direct")
        )
    })
```

32
Plotting these loess estimates $\beta_{\text{loess}}(t; q^*)$ yields the following result. Note that we are plotting (with a scaling) \texttt{predict(loess.fit[[i]])} for $i = 1, 2, 3$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{plot}
\caption{Plot of $\beta(t)/\langle \beta \rangle$ for different $t_{\text{gen}}$.}
\end{figure}

We calculate the RRMSE in each of these estimates as follows.

```r
rrmse_loess_qstar <- mapply(
  function(x, y) compute_rrmse(x$beta, predict(y)),
  x = df, y = loess_fit)
rrmse_loess_qstar
## [1] 0.10502016 0.03356297 0.03478337
```

Next, we consider smoothing each $\beta_k$ time series with $q = q_{\text{opt}}$. Of course, we must first find $q_{\text{opt}}$.

```r
q <- 10:150
## Array of values of RRMSE. Entry `[i, j]` contains the RRMSE
## in the loess estimate obtained from `df_est[[j]]$beta` using
## `q[i]` for the loess smoothing parameter.
rrmse_loess <- mapply(
  function(x, y) {
    sapply(q, function(z) {
      loess_fit <- loess(
        formula = beta ~ t,
        data = y,
        span = z / nrow(y),
        degree = 2,
  
```
As expected, the $\beta_k$ time series corresponding to $t_{\text{gen}} = 2^{-1.5} \cdot 13$ days requires the most smoothing (greatest $q_{\text{opt}}$). Plotting these optimal loess estimates $\beta_{\text{loess}}(t; q_{\text{opt}})$ yields the following result. Once again, we are plotting (with a scaling) $\text{predict(loess_fit[[i]]})$ for $i = 1, 2, 3$. 

---

34
The RRMSE in each of these estimates is calculated as before.

```r
rrmse_loess_qopt <- mapply(
  function(x, y) compute_rrmse(x$beta, predict(y)),
  x = df,
  y = loess_fit
)
rrmse_loess_qopt
```

```
## [1] 0.09848373 0.03346263 0.03475891
```

### S5.3.3 Discussion

Comparing $\beta_k$, $\beta_{loess}(t; q^*)$, and $\beta_{loess}(t; q_{opt})$ for each value of $t_{gen}$, we find that when noise in $\beta_k$ is severe (in this example, when $t_{gen} = 2^{-1.5} \cdot 13$ days), even an optimal degree of smoothing cannot recover the true $\beta(t)$ from the noise, due to underlying bias. No amount of variance reduction can correct the error due to bias. For this reason, smoothing $\beta_k$ using $q^*$ for the loess smoothing parameter $q$ was never much worse than smoothing using the optimal value $q_{opt}$, even when $q^* \ll q_{opt}$ (as was the case with $t_{gen} = 2^{-1.5} \cdot 13$ days):

```r
## Summary of results
data.frame(
  rrmse_raw,
  qstar = as.numeric(qstar[“SI”]),
  rrmse_loess_qstar,
  qopt,
  rrmse_loess_qopt
)
```
This suggests that the decision to fix \( q = q^* \) when generating Figs 5 and 6 does not greatly mischaracterize the effect of parameters like \( t_{\text{gen}} \) on \( \beta(t) \) estimation error. Had we found \( q_{\text{opt}} \) for each raw estimate \( \beta_k \), we would have calculated quantitatively similar RRMSE.

### S6 Sensitivity to error in input parameters

Error in estimates of the seasonally forced transmission rate (Eq (5)) from simulated reported incidence data is also a function of the user-specified values of input parameters, given by

\[
\theta' = (S'_0, I'_0, \nu'_c, \mu'_c, t'_{\text{gen}}, p'_{\text{rep}}, t'_{\text{rep}}).
\]  

(12)

Input error arises when the user’s input mischaracterizes the data-generating process. In our simulated data setting, this occurs when the specified value of an input parameter differs from the value used to simulate data (e.g., when \( S'_0 \neq S_0 \), and so on).

Fig 7A in the manuscript describes how \( \beta(t) \) estimation error varies as a function of input error. To reproduce Fig 7A, we simulate 1000 reported incidence time series using the reference values in (9) for all data-generating parameters. From each reported incidence time series, we estimate the underlying \( \beta_k \) using the S and SI methods with different errors in the input. Specifically, we explore lines in the input parameter space by assigning all input parameters their true (data-generating) value, except a focal parameter (one of \( S_0, I_0, \nu_c, \mu_c, t_{\text{gen}}, p_{\text{rep}}, \) and \( t_{\text{rep}} \)), which we assign each of 41 values logarithmically spaced between \( \frac{1}{4} \) and 4 times its true value. Hence, in total, we consider \( 7 \times 41 \) parametrizations of the S and SI methods. We fit loess curves \( \beta_{\text{loess}}(t; q) \) to each raw transmission rate estimate \( \beta_k \) generated in this process, fixing \( q = q^* \) (Eq (7)), and record the RRMSE in \( \beta_{\text{loess}}(t_k; q^*) \) (See §S6.1 for discussion of the consequences of using fixed \( q \) in this analysis.)

The above algorithm is implemented in our function `test_s2inpars()` (“sensitivity to input parameters”), which takes as arguments

- `par_list_ref`, a list containing values for all data-generating parameters, used in all simulations;
- `pars_to_vary`, a character vector listing the input parameters to be treated as a focal parameter;
- `scale_factors`, a numeric vector listing the factors by which the data-generating value of each focal parameter is scaled to obtain the grid of input values considered for that parameter;
- `with_dem_stoch`, a logical scalar indicating whether simulations should account for demographic stochasticity;
- `nsim`, the number of simulations to perform;
- `loess_par`, a numeric vector of length 2 specifying the value of the loess smoothing parameter \( q \) used when fitting loess curves to raw transmission rate estimates \( \beta_k \).

\( \text{loess\_par}[1] \) is used with the S method. \( \text{loess\_par}[2] \) is used with the SI method.

\texttt{test\_s2inpars()} returns a 4-dimensional array, whose \([i, j, k, m]\)th entry is the RRMSE in the estimate of \( \beta(t) \) from simulation \( k \) of \( \text{nsim} \), generated by assigning \( \text{pars\_to\_vary}[j] \) its true (data-generating) value times \( \text{scale\_factors}[i] \) in the input to the S (\( m = 1 \)) or SI (\( m = 2 \)) method.

We reproduce Fig 7A starting with the following call to \texttt{test\_s2inpars()}.  

```r
rmse_esdsoe <- test_s2inpars(
  pars_to_vary = c("S0", "I0", "nu", "mu", "tgen", "prep", "trep"),
  par_list_ref = make_par_list(epsilon = 0.5, prep = 0.25, trep = 2),
  scale_factors = 2^seq(-2, 2, length.out = 41),
  with_dem_stoch = TRUE,
  nsim = 1000,
  loess_par = qstar
)
```

We retrieve medians from \( \text{rmse\_esdsoe} \) in the next code chunk. Note that some parametrizations cause the SI method to fail. For example, modest underestimation of \( \nu_c \) by \( \nu'_c \) or of \( p_{\text{rep}} \) by \( p'_{\text{rep}} \) causes \( S_k \)—the SI method estimate of \( S(t_k) \)—to become negative. When this happens, \texttt{test\_s2inpars()} assigns RRMSE the value \texttt{NA}. Below, we calculate the median RRMSE only for those parametrizations that yield a full set of 1000 values of RRMSE (no \texttt{NA}s).

```r
## Preallocate memory for median RRMSE:
## one value for each parametrization of the SI method
rmse_50pct <- NA * rmse_esdsoe[, , 1, "SI"]
dim(rmse_50pct)
## [1] 41  7

## Define vector indexing parametrizations for which RRMSE
```
```r
## was never `NA`
ind_no_na <- which(
  apply(!is.na(rrmse_esdsoe[, , , "SI"]), c(1, 2), all)
)

## Calculate median RRMSE for the indexed parametrizations
rrmse_50pct[ind_no_na] <- sapply(ind_no_na,
  function(i) {
    ai <- arrayInd(i, dim(rrmse_50pct))
    quantile(rrmse_esdsoe[ai[1], ai[2] , , "SI"],
      probs = 0.5,
      na.rm = TRUE
    )
  })

Fig 7B repeats the analysis from Fig 7A concerning mis-specification of \(S_0\), except with initially erroneous estimates of \(S_0\) corrected using peak-to-peak iteration (PTPI; see §S7 below for an actual illustration of this technique) before being passed to the S and SI methods. We generate results with PTPI by repeating the last call to `test_s2inpars()` with the additional argument `ptpi_iter = 25`, indicating that PTPI should be employed and stopped after 25 iterations. Since PTPI only affects results for \(S_0\), we set `pars_to_vary = "S0"`.

```r
rrmse_esdsoe_ptpi <- test_s2inpars(
  pars_to_vary = "S0",
  par_list_ref = make_par_list(epsilon = 0.5, prep = 0.25, tresp = 2),
  scale_factors = 2^seq(-2, 2, length.out = 41),
  with_dem_stoch = TRUE,
  nsim = 1000,
  loess_par = qstar,
  ptpi_iter = 25
)

save(rrmse_esdsoe_ptpi, file = "RData/s2inpars_esdsoe_ptpi.RData")
```

There are no issues with RRMSE being assigned `NA` in this analysis, so calculating median RRMSE is more straightforward.

```r
rrmse_ptpi_50pct <- apply(
  rrmse_esdsoe_ptpi[, "S0", , "SI"], 1, quantile,
  probs = 0.5,
  names = FALSE
)
```

38
We reproduce Fig 7 by plotting median RRMSE—saved in `rrmse_50pct` and `rrmse_ptpi_50pct`—as a function of the ratio of the specified value of the focal parameter to the true (data-generating) value.

Fig 7. Sensitivity of $\beta(t)$ estimation error to the user-specified values of input parameters.

### S6.1 A note on smoothing

The exact choice of the loess smoothing parameter $q$ in this analysis is not critical, because error in the raw transmission rate estimate $\beta_k$ is primarily due to bias caused by mis-specified input parameters. Moderate oversmoothing or undersmoothing of $\beta_k$ has a negligible effect on RRMSE when $\beta_k$ is extremely biased. (Fig 8 in the manuscript shows this clearly for the case of mis-specified $S_0$.) Hence the decision to fix $q = q^*$ (Eq (7)) as done here does not have a visible quantitative effect.

### S7 Estimating $S_0$ via PTPI: Example

Fig 8 in the manuscript illustrates the use of peak-to-peak iteration (PTPI) to estimate the initial number of susceptible individuals $S_0 = S(t_0)$ from times series $Z_k$, $B_k$, and $\mu_k$ of (estimated) incidence, births, and the per capita natural mortality rate. The PTPI algorithm relies on the following:

(a) Periodicity of $Z_k$, meaning that $Z_k$ displays recurrent epidemics.

(b) Accuracy of $Z_k$, $B_k$, and $\mu_k$. Systematic errors in these time series bias the reconstruction of susceptibles by PTPI, and ultimately the estimate of $S_0$ to which the iterations converge. This makes sense, given that susceptible dynamics are the direct result of imbalance between susceptible recruitment through birth and susceptible depletion through infection and death.
To reproduce Fig 8, we simulate a reported incidence time series \( C_k \) with known underlying \( S_0 \) (to be estimated).

```r
## List of data-generating parameter values
par_list <- make_par_list(epsilon = 0.5, prep = 0.25)

## Data frame containing time series data
df <- make_data(
  par_list = par_list,
  n = 20 * 365 / 7,
  with_dem_stoch = TRUE,
  seed = 1350
)

## True value of \( 'S0' \) to be estimated
df$S[1]
## [1] 54052
```

We estimate true incidence \( Z \) from reported incidence \( C_k \) as in the SI method:

\[
Z(t_k) \approx Z_k = \frac{1}{p_{rep}} C_{k+r}, \quad r = \frac{t_{rep} \Delta t}{\Delta r},
\]

(13)

We do this using the true (data-generating) values of \( p_{rep} \) and \( t_{rep} \), so that \( Z_k \) estimates \( Z \) without systematic error. (We consider this ideal case in order to demonstrate the validity of the PTPI algorithm. The sensitivity of the method to input error is not explored here explicitly, but is likely captured by the expressions for \( \text{Err}(S_k, \xi \leftarrow \omega \xi) \) derived in §2.7.2 of the manuscript.)

```r
## Time series of estimated incidence
Z <- estimate_beta_SI(df, par_list)$Z
```

We will pass this \( Z_k \) time series to the PTPI algorithm. The complete algorithm consists of a truncation step, described in Box 4 in the manuscript, followed by an iteration step, described in Box 5 in the manuscript. Below, we explain their implementation in R, generating Fig 8 in the process.

### S7.1 Truncation step

The goal of the truncation step is to find the time \( t_a \) of the first peak in \( Z_k \) and the time \( t_b \) of the last peak occurring at the same phase of the cycle. Here, \( Z_k \) is a central moving average applied to the \( Z_k \) time series to remove unwanted noise. (Noise creates “peaks” in \( Z_k \) that we wish to ignore.)
Our function `get_peak_times()` automates the task of (i) applying a central moving average to any equally spaced, (roughly) periodic time series, then (ii) finding peak times. It takes as arguments

- `x`, a numeric vector specifying an equally spaced, (roughly) periodic time series;
- `period`, a numeric scalar specifying the period of `x` in units of the observation interval;
- `bw_mavg`, an integer scalar (a bandwidth) indicating that the central moving average applied to `x` should include `2 * bw_mavg + 1` points;
- `bw_peakid`, an integer scalar (a bandwidth) indicating that `x_mavg[i]` should be considered a peak if and only if `x_mavg[i] > x_mavg[j]` for all `j` such that `0 < |i - j| < bw_peakid`.

In the last item above, `x_mavg` is a vector of length `length(x)` containing the central moving average applied to `x`. `x_mavg[i]` is equal to

```
mean(x[(i-bw_mavg):(i+bw_mavg)])
```

for all `i` from `bw_mavg+1` to `length(x)-bw_mavg`, and equal to `NA` everywhere else (i.e., at the edges).

`get_peak_times()` returns a list containing `x_mavg` and two index vectors `all` and `phase`. `all` indexes all peaks in `x_mavg`, while `phase` indexes only those peaks in phase with the first peak (and is therefore a subset of `all`).

Before we construct a call to `get_peak_times()`, we must ascertain that our time series `Z_k` of estimated incidence is roughly periodic and determine the period. Plotting `Z_k`, it is clear that it is periodic with a 1-year cycle.

![Graph showing periodicity with a 1-year cycle](image)

In general, it may be helpful to inspect the power spectrum of `Z_k` to determine the period, but we do not do this here.

We locate the peaks in `Z_k` with the following call to `get_peak_times()`.

```r
get_peak_times()
```
## List of index vectors for peaks in incidence time series

```r
peaks <- get_peak_times(
  x = Z,
  period = with(par_list, (365 / 7) / dt_weeks),
  bw_mavg = 6,
  bw_peakid = 8
)
```

## All peaks

```r
peaks$all[1:10]
## [1]  61 118 171 218 274 322 383 429 478 539
```

## All peaks in phase with first

```r
peaks$phase[1:10]
## [1]  61 118 171 218 274 322 383 429 478 539
```

Above, we assigned `period` the value of 1 year in units of the observation interval. We chose bandwidths `bw_mavg = 6` and `bw_peakid = 8` using a simple tuning procedure. First, we chose the smallest value of `bw_mavg` that eliminated noise near peaks in $Z_k$. This was determined by visual inspection of the moving average $\overline{Z}_k$ (`peaks$x_mavg`). Next, we chose an arbitrary value of `bw_peakid` greater than 5 and less than half of `period`. This ensured that the definition of a peak was meaningful (a point greater than many of its nearest neighbours) and that peaks were not being compared against other peaks. The exact choice of `bw_peakid` tends not to be critical provided $\overline{Z}_k$ is smooth near the peaks.

Note that the two index vectors `all` and `phase` returned by `get_peak_times()` are identical. In this example, all peaks in $Z_k$ are in phase, because the time between peaks is precisely the period (1 year). This is not true in general. For example, a 2-year cycle can have major and minor peaks that are out of phase. In this case, `all` would index both major and minor peaks, but `phase` would index either minor peaks or major peaks (but not both).

Plotting true incidence $Z$ (`df$Z`), estimated incidence $Z_k$ (`Z`), and the central moving average $\overline{Z}_k$ (`peaks$x_mavg`), as well as indicators of the times of peaks in $\overline{Z}_k$ in phase with the first peak (`peaks$phase`), we reproduce Fig 8A (see below). Fig 8A verifies that all of the peaks of interest were identified by `get_peak_times()`.

We conclude the truncation step of the PTPI algorithm by retrieving the index of the first peak in $\overline{Z}_k$ and the index of the last peak occurring at the same phase of the cycle.
## Index of first peak
\[
a \leftarrow \text{with(peaks, phase}[1])
\]

## Index of last peak in phase with first peak
\[
b \leftarrow \text{with(peaks, phase[length(phase)])}
\]

The precise times \(t_a\) and \(t_b\) of these peaks are given by \(\text{df}[t[c(a, b)]]\).

### S7.2 Iteration step

The goal of the iteration step is to use times series \(Z_k\), \(B_k\), and \(\mu_k\) of (estimated) incidence, births, and the *per capita* natural mortality rate to iteratively update an initial estimate of \(S_0 = S(t_0)\). This updating procedure depends on the result of the truncation step. The iteration step is implemented in our function \(\text{ptpi()}\), which takes as arguments

- \(\text{df}\), a data frame with columns \(Z\), \(B\), and \(\mu\) specifying time series \(Z_k\), \(B_k\) and \(\mu_k\) of (estimated) incidence, births, and the *per capita* natural morality rate;
- \(\text{par_list}\), a list with elements \(\hat{N}_0\), \(\nu\), and \(\mu\), specifying a population size \(\hat{N}_0\) and constant vital rates \(\nu\) and \(\mu\), which are used to create mock vital data in the event that \(\text{df}\) does not possess columns \(B\) or \(\mu\) (\(\text{ptpi()}\) will set \(B_k = \nu \hat{N}_0 \Delta t\) and \(\mu_k = \mu\) for all \(k\));
- \(\text{a}\), an integer scalar indicating the index of the first peak in \(\text{df}[Z]\);
- \(\text{b}\), an integer scalar indicating the index of the last peak in \(\text{df}[Z]\) in phase with the first peak;
- \(\text{initial_S0_est}\), a numeric scalar indicating an initial estimate of \(S_0\);
- \(\text{iter}\), an integer scalar indicating the number of iterations to perform before stopping.

We carry out the iteration step with the following call to \(\text{ptpi()}\). For this example, we suppose that our initial guess of \(S_0\) is 4 times greater than its true value, and ask for our estimate to be updated 25 times. We provide the peak indices \(a\) and \(b\) obtained in the truncation step (see above). Finally, in the first argument, we specify our incidence time series and nothing else, and in the second argument, we specify the data-generating values of \(\hat{N}_0\), \(\nu\), and \(\mu\). This means that \(\text{ptpi()}\) will construct mock vital data without any systematic error and use it in conjunction with the supplied incidence time series.

## List containing PTPI output
\[
\text{ptpi_out} \leftarrow \text{ptpi(df = data.frame(Z),}
\]

\[
\text{df} = \text{data.frame(Z)},
\]

43
par_list = par_list,
a = a,
b = b,
initial_S0_est = df$S[1] * 4,
iter = 25
)

ptpi() returns a list with elements

- **S_mat**, a numeric matrix with \texttt{nrow(df)} rows and \texttt{iter+1} columns, containing the susceptible time series generated in each iteration;
- **S0**, a numeric vector of length \texttt{iter+1} listing the initial estimate of \( S_0 = S(t_0) \) and the estimate obtained in each iteration (equivalent to \texttt{S_mat[1, ]});
- **S0_final**, a numeric scalar indicating the final estimate of \( S_0 \) (equivalent to \texttt{S0[length(S0)]});
- **SA**, a numeric vector of length \texttt{iter+1} listing the initial estimate of \( S_a = S(t_a) \) (equal to the initial estimate of \( S_0 \)) and the estimate obtained in each iteration (equivalent to \texttt{S_mat[a, ]});
- **SA_final**, a numeric scalar indicating the final estimate of \( S_a \) (equivalent to \texttt{SA[length(SA)]}).

Examining \texttt{ptpi_out}, we find that the iterations converged to an accurate estimate of \( S_0 \).

```r
tpti_out$S0
```
```
  [1] 216208.00 138745.04  94654.54  73216.89  62793.50  57725.45  55261.27
  [8]  54063.13  53480.58  53197.33  53059.61  52992.64  52960.09  52944.26
 [15]  52936.56  52932.82  52931.00  52930.11  52929.68  52929.47  52929.37
 [22]  52929.32  52929.30  52929.29  52929.28  52929.28
```

```r
(pttpi_out$S0_final - df$S[1]) / df$S[1]
```
```
  [1] -0.02077116
```

Fig 8B (see below) displays the \texttt{iter+1} susceptible time series \( S_k \) obtained in each iteration of PTPI. To reproduce Fig 8B, we plot the columns of \texttt{pttpi_out$S_mat}, scaled by \texttt{with(par_list, 1/N0)}.
## Matrix with ordered susceptible time series as columns

\[
\text{ptpi_out}\$S\_mat[1:10, 1:5]
\]

\[
\begin{array}{ccccc}
[1,] & 216208 & 138745.0 & 94654.54 & 73216.89 \\
[2,] & NA & 138657.8 & 94601.07 & 73179.87 \\
[3,] & NA & 138394.6 & 94371.71 & 72966.93 \\
[4,] & NA & 138099.7 & 94110.54 & 72722.18 \\
[5,] & NA & 137625.1 & 93669.64 & 72297.67 \\
[6,] & NA & 137078.8 & 93157.10 & 71801.53 \\
[7,] & NA & 136581.0 & 92692.94 & 71353.74 \\
[8,] & NA & 136327.4 & 92473.04 & 71150.21 \\
[9,] & NA & 135470.3 & 91649.54 & 70343.06 \\
[10,] & NA & 134793.7 & 91006.60 & 69716.46 \\
\end{array}
\]

Note that the first column contains `NA`, but only up to index `a` (not shown), where the first iteration starts.

Fig 8C (see below) displays the SI method estimate of the transmission rate corresponding to each estimate of \( S_0 \) listed in `ptpi_out\$S0`. To reproduce Fig 8C, we pass each estimate of \( S_0 \) to `estimate_beta_SI()`, specifying the true (data-generating) value of every other input parameter. We fit a loess curve \( \hat{\beta}_{\text{loess}}(t; q^*) \) to each raw transmission rate estimate \( \hat{\beta}_k \), and record \( \hat{\beta}_{\text{loess}}(t_k; q^*) \) as a column in a matrix \( \text{beta_mat} \). Finally we plot the columns of \( \text{beta_mat} \), scaled by `with(par_list, 1/beta_mean)`.

## Matrix with ordered transmission rate time series as columns

\[
\text{beta_mat} <- \text{sapply}(\text{ptpi_out}\$S0, \\
\quad \text{function}(x) \{ \\
\quad \quad \text{par_list_with_err} \leftarrow \text{within}(\text{par_list}, S0 \leftarrow x) \\
\quad \quad \text{df_est} \leftarrow \text{estimate_beta_SI}(\text{df, par_list_with_err}) \\
\quad \quad \text{loess_fit} \leftarrow \text{loess}( \\
\quad \quad \quad \text{formula} = \beta \sim t, \\
\quad \quad \quad \text{data} = \text{df_est}, \\
\quad \quad \quad \text{span} = \text{qstar["SI"] / nrow(df_est)}, \\
\quad \quad \quad \text{degree} = 2, \\
\quad \quad \quad \text{na.action} = \text{"na.exclude"}, \\
\quad \quad \quad \text{control} = \text{loess.control(surface = "direct")}) \\
\quad \quad \text{predict(loess_fit)} \\
\quad \}) \\
\]

\[
\text{beta_mat}[1:10, 1:5]
\]
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>[2,]</td>
<td>2.978043e-06</td>
<td>4.632868e-06</td>
<td>6.774538e-06</td>
<td>8.736727e-06</td>
<td>1.016727e-05</td>
</tr>
<tr>
<td>[3,]</td>
<td>2.925529e-06</td>
<td>4.557144e-06</td>
<td>6.675664e-06</td>
<td>8.623999e-06</td>
<td>1.004922e-05</td>
</tr>
<tr>
<td>[4,]</td>
<td>2.875193e-06</td>
<td>4.484526e-06</td>
<td>6.580771e-06</td>
<td>8.515711e-06</td>
<td>9.935726e-06</td>
</tr>
<tr>
<td>[5,]</td>
<td>2.827043e-06</td>
<td>4.415025e-06</td>
<td>6.489875e-06</td>
<td>8.411877e-06</td>
<td>9.826795e-06</td>
</tr>
<tr>
<td>[6,]</td>
<td>2.781082e-06</td>
<td>4.348645e-06</td>
<td>6.402980e-06</td>
<td>8.312501e-06</td>
<td>9.722428e-06</td>
</tr>
<tr>
<td>[7,]</td>
<td>2.737311e-06</td>
<td>4.285386e-06</td>
<td>6.320081e-06</td>
<td>8.217575e-06</td>
<td>9.622613e-06</td>
</tr>
<tr>
<td>[8,]</td>
<td>2.695731e-06</td>
<td>4.225250e-06</td>
<td>6.241182e-06</td>
<td>8.127098e-06</td>
<td>9.527348e-06</td>
</tr>
<tr>
<td>[9,]</td>
<td>2.656352e-06</td>
<td>4.168251e-06</td>
<td>6.166301e-06</td>
<td>8.041095e-06</td>
<td>9.436658e-06</td>
</tr>
<tr>
<td>[10,]</td>
<td>2.619180e-06</td>
<td>4.114399e-06</td>
<td>6.095452e-06</td>
<td>7.959580e-06</td>
<td>9.350560e-06</td>
</tr>
</tbody>
</table>
**Fig 8.** Example of $S(t)$ and $\beta(t)$ reconstruction with an overestimate of $S_0$ corrected by peak-to-peak iteration.

**S8 Estimating $S_0$ via PTPI: Convergence**

Fig 9 in the manuscript displays the result of applying PTPI (25 iterations) to estimate $S_0$ from 1000 realizations of a reported incidence time series, starting from each of two initial
guesses: $\frac{1}{4}$ and 4 times the true value. The aim of this analysis is to assess the bias and variance in the limiting estimate of $S_0$.

To reproduce Fig 9, we record the estimate obtained at each iteration (for each initial guess and simulation) in an array. The following code chunk preallocates space for this output and creates a list of parameter values to be used in simulations of reported incidence.

```r
## Array with entry `[i, j, k]` equal to the `i`'th estimate of `S0` generated from the `j`'th initial guess and the `k`'th simulated reported incidence time series
out <- array(NA, dim = c(26, 2, 1000))

## List of data-generating parameter values
par_list <- make_par_list(epsilon = 0.5, prep = 0.25)
```

The next code chunk fills in the `out` array with our desired output and saves it in `RData/ptpi_convergence.RData`.

```r
for (k in 1:1000) {
  ## Data frame containing time series data
df <- make_data(par_list = par_list,
                 n = 20 * 365 / 7,
                 with_dem_stoch = TRUE,
                 seed = k)

  ## PTPI: truncation step
Z <- estimate_beta_SI(df, par_list)$Z
peaks <- get_peak_times(x = Z,
                        period = with(par_list, (365 / 7) / dt_weeks),
                        bw_mavg = 6,
                        bw_peakid = 8)

  ## PTPI: iteration step
out[, , k] <- sapply(c(0.25, 4),
                 function(x) {
                   ptpi_out <- ptpi(df = data.frame(Z),
                                    par_list = par_list,
                                    ...
We want the median and 5th and 95th percentiles of the estimate of $S_0$ obtained at each iteration (for each initial guess).

```
pct <- apply(out, c(1, 2), quantile, probs = c(0.05, 0.5, 0.95))
```

Plotting these as a functions of iteration, we reproduce Fig 9.

![Fig 9. Convergence of estimates of $S_0$ obtained using peak-to-peak iteration.](image)

In the manuscript, we report the median and 5th and 95th percentiles of the relative error in the estimate of $S_0$ obtained in the last iteration (for each initial guess). These are calculated as follows.
S9 Appendix: Choice of discretization in the SI method

The SI method modifies a method presented in [3] by deJonge. Here, we cast the SI method and deJonge’s method as two possible algorithms from a set of nine, differing according to (i) how

$$\frac{dS}{dt} = \nu(t)\hat{N}_0 - \beta(t)SI - \mu(t)S,$$  \hspace{2cm} (14a)

$$\frac{dI}{dt} = \beta(t)SI - \gamma I - \mu(t)I$$ \hspace{2cm} (14b)

are discretized (forward Euler, backward Euler, or trapezoidal method) in order to estimate susceptibles $S(t)$ and infecteds $I(t)$, and (ii) how

$$\frac{dQ}{dt} = \beta(t)SI$$ \hspace{2cm} (15)

is discretized (forward Euler, backward Euler, or both) in order to estimate the transmission rate $\beta(t)$. (“Both” means that the two estimates of $\beta(t)$ obtained by forward and backward Euler are averaged to generate a final estimate.) DeJonge’s method uses forward Euler throughout, whereas the SI method uses the trapezoidal method for Eqs (14) and both forward and backward Euler for Eq (15).

Here, we show that the SI method is more accurate than deJonge’s method and the seven other algorithms. We further show that the SI method and deJonge’s method are nearly unbiased (asymptotically) in the absence of input error.

S9.1 Nine discretization schemes

Our function `estimate_beta_SI()` takes a third argument `method`, which must be assigned a vector of length 2. `method[1]` has options “forward”, “backward”, and “trapezoid” (default), telling `estimate_beta_SI()` how to numerically integrate Eqs (14). `method[2]` has options “forward”, “backward”, and “both”, (default) telling `estimate_beta_SI()` how to numerically integrate Eq (15).

\[
Z_k \leftarrow \frac{1}{p_{\text{rep}}} C_{k+r}
\]

(16a)

\[
S_k \leftarrow \begin{cases} 
(1 - \mu_{k-1} \Delta t) S_{k-1} + B_k - Z_k & \text{if } \text{method[1]} = \text{"forward"} \\
\frac{S_{k-1} + B_k - Z_k}{1 + \frac{1}{2} \mu_k \Delta t} & \text{if } \text{method[1]} = \text{"backward"} \\
(1 - \frac{1}{2} \mu_{k-1} \Delta t) S_{k-1} + B_k - Z_k & \text{if } \text{method[1]} = \text{"trapezoid"}
\end{cases}
\]

(16b)

\[
I_k \leftarrow \begin{cases} 
[1 - (\gamma + \mu_{k-1}) \Delta t] I_{k-1} + Z_k & \text{if } \text{method[1]} = \text{"forward"} \\
\frac{I_{k-1} + Z_k}{1 + (\gamma + \mu_k) \Delta t} & \text{if } \text{method[1]} = \text{"backward"} \\
[1 - \frac{1}{2} (\gamma + \mu_{k-1}) \Delta t] I_{k-1} + Z_k & \text{if } \text{method[1]} = \text{"trapezoid"}
\end{cases}
\]

(16c)

\[
\beta_k \leftarrow \begin{cases} 
\frac{Z_{k+1}}{S_k I_k} & \text{if } \text{method[2]} = \text{"forward"} \\
\frac{Z_k}{S_k I_k} & \text{if } \text{method[2]} = \text{"backward"} \\
\frac{Z_k + Z_{k+1}}{2S_k I_k} & \text{if } \text{method[2]} = \text{"both"
\end{cases}
\]

(16d)

Hence the SI method corresponds to `method = c("trapezoid", "both")`, while deJonge’s method corresponds to `method = c("forward", "forward")`.

### S9.2 Comparison of RRMSE, bias, and variance

We will compare the nine algorithms described in Eqs (16) using two metrics. First, we consider performance as measured by the RRMSE in the raw transmission rate estimates \(\beta_k\). Second, we consider bias in the average 1-year cycle, calculated from the linear interpolant of \(\beta_k\) as in §S3.

#### S9.2.1 RRMSE

We simulate 100 reported incidence time series \(C_k\) using each of 41 values for the case reporting probability \(p_{\text{rep}}\), logarithmically spaced between 0.01 and 1. (Smaller values of \(p_{\text{rep}}\) generate noisier \(C_k\), leading to noisier \(\beta_k\).)

```r
prep <- 10^seq(-2, 0, length.out = 41)
par_list <- make_par_list(epsilon = 0.5)
nsim <- 100
```

We estimate the underlying, seasonally forced transmission rate \(\beta(t)\) (Eq (5)) from each simulated reported incidence time series using each algorithm described in Eqs (16), and record the RRMSE in each raw estimate \(\beta_k\). We can preallocate space for this output.
The next code chunk implements the steps in this routine, saving main results in the file `RData/euler.RData`. We can reuse simulations from §S4, which were saved in the directory `RData/loess/`.

```r
for (i in seq_along(prep)) {
  ## Update `par_list` with current value of `prep`
  par_list$prep <- prep[i]

  ## Create a directory for this loop's `.RData`
  dirname <- paste0("RData/loess/", "prep_log10v-", sprintf("%+05.0f", log(prep[i], 10) * 1000), "/")
  if (!dir.exists(dirname)) {
    dir.create(dirname, recursive = TRUE)
  }

  for (j in seq_len(nsim)) {

    message("prep value", i, " of ", length(prep), ", ", "sim", j, " of ", nsim

    ## File name for simulation
    filename <- paste0(dirname, "sim", sprintf("%04.0f", j), ".RData")

    ## Simulate reported incidence data, if you haven't already
    if (file.exists(filename)) {
      load(filename)
    } else {
      df <- make_data(par_list = par_list,
```
n = 20 * 365 / 7,
with_dem_stoch = TRUE,
seed = j
)

save(df, file = filename)
}

for (m1 in method1_names) {
  for (m2 in method2_names) {

    ## Estimate the seasonally forced transmission rate
    ## from reported incidence
    df_est <- estimate_beta_SI(df, par_list, method = c(m1, m2))

    ## Record the error
    out[i, j, m1, m2] <- compute_rrmse(df$beta, df_est$beta)

  }
}
}

attr(out, "arg_list") <- list(
  prep = prep,
  par_list = par_list
)

save(out, file = "RData/euler.RData")

We desire the median and 5th and 95th percentiles of RRMSE for each value of $p_{rep}$, for each of the nine algorithms used to estimate $\beta(t)$.

pct <- apply(out, c(1, 3, 4), quantile, probs = c(0.05, 0.5, 0.95))

Plotting these as functions of $p_{rep}$, and stratifying the results by method[1] (panel title) and method[2] (legend label) yields the following figure.
Fig 10. Performance of the nine discretization schemes, as measured by RRMSE in the raw transmission rate estimate $\beta_k$. Panel titles specify the discretization of Eqs (14). Legend labels specify the discretization of Eq (15).

For every choice of method[1] (panel title), the best choice of method[2] (legend label) was typically “both”. On the other hand, for a given choice of method[2], the
best choice of \texttt{method[1]} (by a small margin) was typically the one that avoided
mismatch with \texttt{method[2]}. That is, when forward and backward Euler were used to
discretize Eq (15), RRMSE was typically smallest when forward and backward Euler,
respectively, were used to discretize Eqs (14). Similarly, when both forward and backward
Euler were used to discretize Eq (15), RRMSE was typically smallest when the trapezoidal
method was used to discretize Eqs (14). This combination, with
\texttt{method = c("trapezoid", "both")}, gave the best performance overall.

S9.2.2 Bias and variance

In §S3, we looked at bias and variance in the 1-year cycles embedded in raw transmission
rate estimates $\beta_k$ spanning 1000 years. There, we compared the S and SI methods. Here,
we compare the nine algorithms described in Eqs (16).

We simulate 1000 years of weekly observations of reported incidence, including in the
simulation environmental noise in transmission ($\epsilon = 0.5$), demographic stochasticity, and
random under-reporting of cases ($\rho_{\text{rep}} = 0.25$). 

```r
par_list <- make_par_list(epsilon = 0.5, prep = 0.25)
df <- make_data(
  par_list = par_list,
  n = 1000 * 365 / 7 + 1,
  with_dem_stoch = TRUE,
  seed = 1352
)
```

We estimate the seasonally forced $\beta(t)$ using all nine discretization schemes, without input
error.

```r
df_est <- mapply(
  function(x) {
    mapply(
      function(y) {
        estimate_beta_SI(df, par_list, method = c(x, y))
      },
      y = method2_names, SIMPLIFY = FALSE
    ),
    x = method1_names, SIMPLIFY = FALSE
  }
)
```

We linearly interpolate each raw time series estimate $\beta_k$. 

55
fits <- lapply(df_est, function(x) {
    lapply(x, function(y) {
        approxfun(y$t, y$beta, method = "linear", rule = 1)
    })
})

As in §S3, we define the initial observation time \( t_0 \), period \( \text{period} \), and number of cycles \( m \), then use `get_phase_average()` to calculate the average 1-year cycle in the linear interpolants.

```r
## First and last time points, retrievable
## from any data frame in the list
t0 <- df_est[[1]][[1]]$t[1]
tn <- df_est[[1]][[1]]$t[nrow(df_est[[1]][[1]])]

## 1-year period in units of the observation interval
period <- with(par_list, (365 / 7) / dt_weeks)

## Number of cycles
m <- floor((tn - t0) / period)

greater phase average <- function(s, f) {
    x <- f(t0 + (s %% period) + (0:(m-1)) * period)
    mean(x, na.rm = TRUE)
}
s_grid <- seq(0, period, length.out = 150)
average_one_year <- lapply(fits, function(x) {
    data.frame(
        s_grid,
        lapply(x, function(f) sapply(s_grid, get_phase_average, f = f))
    )
})
```

We plot the 1000 individual cycles and their average on the same 1-year axis, yielding a 9-panel plot.
Fig 11. Bias and variance incurred by the nine discretization schemes. Row names specify the discretization of Eqs (14). Column names specify the discretization of Eq (15).

Following the pattern of Fig 10, Fig 11 shows that mismatch between method[1] and method[2] is detrimental: in the off-diagonal panels, the average 1-year cycle fails to capture the correct seasonal amplitude. In addition, use of backward Euler to discretize Eqs (14) appears ill-advised: in the panels from the second row, the average 1-year cycle lags the true cycle. Finally, it is apparent that the SI method (bottom right panel) and deJonge’s method (top left panel) are both nearly unbiased.
All methods appear prone to propagating noise from reported incidence (due to process and observation error) to $\beta_k$. However, the SI method and deJonge’s method stand out as being the least and most susceptible, respectively, to propagation of spurious noise. This likely accounts for the difference in their performance shown in Fig 10.

References

