A filtering approach to estimation of the epidemic exponential growth rate r(t) from incidence timeseries

Matthew So, Jonathan Dushoff

April 2021

Contents

1 Abstract

Outbreaks of infectious diseases such as the ongoing COVID-19 pandemic pose a threat to population health. In order to combat these outbreaks, it is important for public health authorities to understand how quickly an epidemic is spreading. One metric for studying epidemic spread is the exponential growth rate r(t). There currently exists few methods for estimating this metric using incidence data. We propose an estimation method that is conceptually simple, easily implemented, and requires few priors on infection dynamics. This method involves filtering the raw observed incidence data with a Savitzky-Golay filter to remove periodicity and reduce noise, then using another Savitzky-Golay filter on the logarithm of the filtered incidence data to perform a logarithmic differentiation. This estimation method has been evaluated on a custom modified discrete-time SEIR model incorporating a separate periodic observation process resulting in a greater number of observations on certain days of the week. The proposed estimation method is effective at estimating r(t) in simulated data, even given periodic and noisy signals similar to those found in real-world data. Usage of this estimation method on real-world data results in qualitatively stable estimates. The proposed estimation method may make r(t) estimation quicker and more accurate, which may allow for better management of infectious disease outbreaks. Additionally, this modified SEIR model may generate timeseries more representative of real-world data, which can be used to improve other epidemiological parameter estimation methods that rely on incidence timeseries.

2 Modelling

2.1 Rationale

In order to evaluate any potential r(t) estimation method, we must have a ground truth r(t); therefore, we need a model for which we can obtain a ground truth r(t) estimate. While a standard SEIR model might seem to be a good choice, it lacks some dynamical features that are found in real-world data, including noise and periodicity. Therefore, an SEIR-like model was developed which contained these dynamical features.

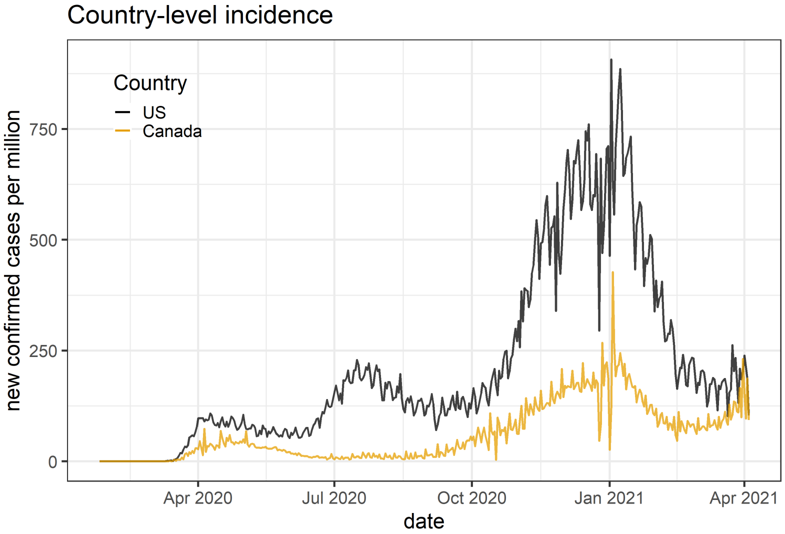

Figure 1: Real-world COVID-19 incidence data contains noise and ”spiky” periodicity. [1]

It is hypothesized in this work that this ”spikiness” primarily reflects daily changes in observations. Any trends in infection, such as changes in r(t), would be heavily smoothed by the incubation period; however, this is inconsistent with the consistent single-day spikes found in real data.

2.2 Model Explanation

2.3 About the model

The current model is a discrete-time model based on the SEIR framework, which also models weekday trends in observation data. The main idea behind the model is that the probability density function for an infectee’s observation time is modified such that observation on a Monday (for example) is more likely than the other days.

2.4 Explanations for features in the model

2.4.1 Conventions

The sum of two probability distributions X and Y is denoted as $X + Y$, and is computed as the convolution of their probability density functions. The expected value is denoted as $E[X]$. Temporary variables have self-explanatory names.

2.4.2 State variable explanations

All state variables are initialized to 0 except $S \leftarrow 10,000,000$.

- S: susceptible.
- E: exposed, but non-infectious
- I: infectious
- R: recovered (or dead)
- O: cumulative infectious individuals who have been observed
- t: time since epidemic start.

2.4.3 Parameter/Function explanations

- $\beta(t)$: Same meaning as in typical SEIR models.
- $\mathcal{R}_0$: The basic reproductive number (that is, the reproductive number not scaled by S/N. Implicitly used to define $\beta(t)$. See results for details.
- `baseObservationDist`: Probability density function of time for an infection to be observed. This PDF is modified based on the day of the week. Currently set to discrete lognormal distribution with μ=1.7, log-SD=0.5
- `incubationDist`: Probability density function of time to go from $E \rightarrow I$. Currently set to discrete lognormal distribution with μ=1.63, log-SD=0.5. [2]
- `infectiousDist`: Probability density function of time to go from $I \rightarrow R$. (time spent infectious). Currently set to exponential distribution with mean=10 days. (Note: only exponential distributions are currently supported.)
- κ: 1/(dispersion parameter) in an alternatively parameterized negative binomial distribution. [3] $negBinom(mean, 0)$ is implemented as the Poisson. I chose 0 as the value of κ, implicitly making all instances of $negBinom$ actually $Poisson$.
- $negBinom$: Negative binomial distribution parameterized by (mean, κ).
- `dayScalers`: On each weekday, the probability of observation is multiplied by this value. Set to 1.1 for Monday and Tuesday and 1 otherwise.
- `observationProb`: The total probability that an infected individual will be observed. Set to 0.8.
- t_{max}: Maximal value of t for simulation. Set to 401.

2.4.4 Derived variable explanations

- `incidence`: The number of new individuals infected today that weren't infected yesterday. At t=0, incidence is set to 10.
- μ: The reciprocal of the E[`incubationDist`], analogous to the $E \to I$ controlling parameter in SEIR model.

2.5 Explanation of model logic

2.5.1 Important assumptions

1. The infectious disease being modeled follows SEIR dynamics.
2. When an individual is infected, the time that they will be detected at and the time they will become infectious at are defined by two independent distributions.
3. Dynamical noise does not exist; that is, the true number of people infected each day is deterministic.
4. However, not every person who is infected will be observed. Observation noise can result in a non-deterministic number of infections being observed each day.
5. An infectee is only symptomatic during the period that they show symptoms (the Infectious period in the SEIR model). However, this is not entirely realistic in the case of COVID-19 [4].

2.5.2 Pre-simulation setup

1. Compute an observation distribution for every day in the week. For each weekday, modify a copy of `baseObservationDist` such that each weekday in the distribution is multiplied by the corresponding item in dayScalers. Then, renormalize this distribution to have a sum of 1.
2. Set values for each state variable. Set $N \leftarrow sum(S, E, I, R)$. Additionally, set $t \leftarrow -1$ for setup purposes.

2.5.3 Simulation

These steps are executed for each desired value of t between 0 and t_{max}.

1. Compute the weekday, $t \pmod 7$.
2. If t=0, then initialize *incidence* to some number. Else, if running in full dynamical noise mode, compute `incidence` $= negBinom(SI\beta(t)/N, \kappa)$ and `expectedIncidence` $= SI\beta(t)/N$.
3. Transition event times are put into a separate list with each element at each index representing the number of events at (index) days from now.

4. Transition events are distributed amongst all future days with a value equal to their respective probability density functions. $E \rightarrow I$ events are proportional to `incubationDist`, $I \rightarrow R$ is proportional to `incubationDist + infectiousDist`, and $E \rightarrow I$ is proportional to the appropriate `weekdayObservationDist` scaled by `obervationProb`.

5. The number of observations added at each point in the future is sampled from $negBinom(n_{obs}, \kappa)$.

6. Execute all other transition events occurring today. For $S \rightarrow E, S = S - 1, E = E + 1$. For $E \rightarrow I, E = E - 1, I = I + 1$. For $I \rightarrow R, I = I - 1, R = R + 1$. For $E \rightarrow O, O = O + 1$

7. Store all state variables as well as `incidence` from today.

2.5.4 Post-simulation

1. Compute `scaledIncidence` by multiplying `incidence` by `observationProb`.

2. Compute the ground truth r(t) by taking the first differences of $log($`incidence`$)$. (The r(t) for a given day is equal to $log(t+1) - log(t)$). The ground truth is undefined for days on which `incidence` $= 0$ or $t = t_{max}$.

3 r(t) Estimation

3.1 The Savitzky-Golay filter

The Savitzky-Golay filter is a low-pass filter typically used for signal processing applications. The principle behind the Savitzky-Golay filter is that, for each time t, a polynomial regression model is fit from the points surrounding it. For this work, three additional parameters are considered: window size, polynomial order, and derivative order.

For each time t, a polynomial regression model is fit using the incidence data from the times $[t - \texttt{windowWidth}, t + \texttt{windowWidth}]$, where $\texttt{windowWidth} = floor(\texttt{windowSize}/2)$. For this work, polynomial order is set to 1 for all filters (making each window a linear regression). Then, the value of the filter at each time t is equal to the `derivativeOrder`th derivative of the best-fit polynomial at time t. For the left `windowWidth` points, the regression model is fit to the first `windowSize` points in the timeseries, and the value of the filter at each of the first `windowSize` points is equal to the regression model's prediction at each of these points. The analogous method is applied to the right `windowSize` points, using the last `windowWidth` points to train the regression model.

While confidence interval estimation for this filter is not standard in the signal processing field, it can be done for derivative order 0 by taking the confidence interval on each predicted filter value. (Other combinations may also be possible; for example, for derivative order 1 and polynomial order 1 by taking the confidence interval on the slope of the linear regression.)

3.2 Estimation algorithm

1. The observed incidence timeseries is filtered using a Savitzky-Golay filter of window size 7, polynomial order 1, and derivative order 0, saving both the mean value of the filter and the confidence interval at each point. This is intended as an initial low-pass filter on the incidence data, which acts to reduce periodicity and noise.

2. Define a function `estim(incidence, windowSize, shiftAmt)`.

   (a) `logIncidence` ← log(`incidence`).

   (b) Filter `logIncidence` using a Savitzky-Golay filter of length `windowSize`, polynomial order 1, and derivative order 1. NaNs are ignored (not used as points in the linear regression). Only the mean estimate at each point needs to be considered.

   (c) Shift the resulting curve backwards by `shiftAmt`, which should be set to the rounded expected value of the time between infection and observation.

   (d) Return the shifted curve, which is the mean r(t) estimate given the input incidence timeseries.

3. Sample a new plausible incidence timeseries. That is, for `nBootstrap` iterations, for each point in the filtered timeseries obtained from step 1, randomly sample a value from the confidence interval of the point and append it to a new timeseries. Store all new plausible incidence timeseries.

4. Call `estim` on each plausible incidence timeseries to obtain an r(t) estimation.

5. Call `estim` on the mean value of the filtered timeseries obtained from step 1. This is the mean r(t) estimate.

6. For each time t, compute the lower 0.05 and upper 0.95 quantiles of the r(t) estimated from the plausible incidence timeseries. This is the lower and upper confidence interval bounds on the r(t) estimates. (For example, if the plausible incidence timeseries was stored in a `length(incidence)` × `nBootstrap` matrix where `length(incidence)` is the number of rows, then compute quantiles row-wise to obtain a `length(incidence)` $\times 1$ matrix.)

`windowSize` can be chosen as any odd number, but 7 and 15 have been tested. Lower numbers should react to changes more quickly but should also result in noisier estimates, and vice versa.

3.3 Caveats and potential improvements

- Confidence intervals assume a normally distributed confidence interval; this may be changed to a t-distributed confidence interval later on).

- Using deconvolution rather than shifting is more theoretically sound, but deconvolution may amplify noise inherent in the data [5] [6]. The effect of using deconvolution rather than shifting may be investigated later on.

4 Results

4.1 Modelling

Varying $\mathcal{R}_0(t)$ was used to define model behavior. These parameters were chosen to simulate a rise, peak, and decline in COVID-19 incidence.

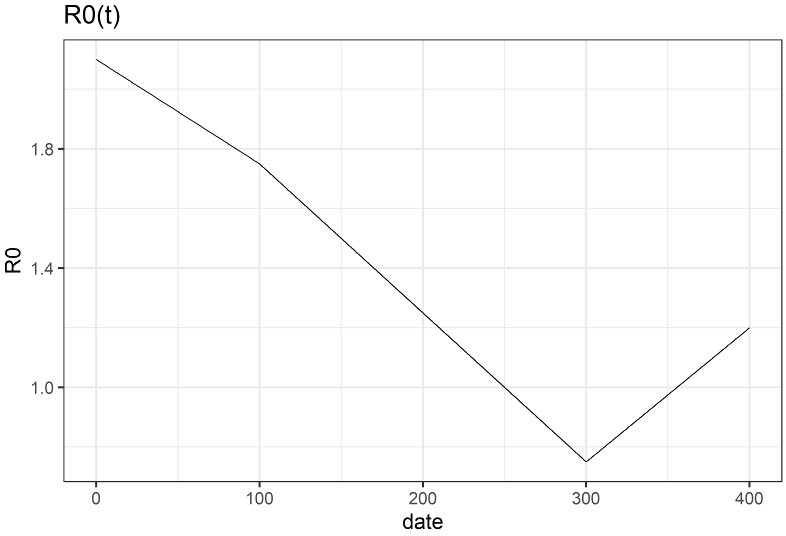

Figure 2: A piecewise linear function was used to define gradual changes in $\mathcal{R}_0(t)$, which drives the incidence in this model.

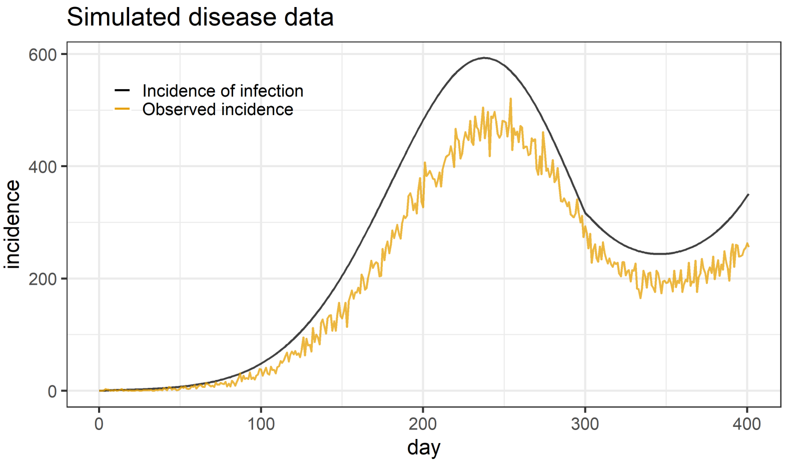

Figure 3: A plot of modeled incidence over time. The incidence of infection is perfectly deterministic, and increases exactly when an individual is infected. The observed incidence is periodic, has observation noise, is delayed by the time delay distribution between infection and observation, and reflects the 80% of observed infections.

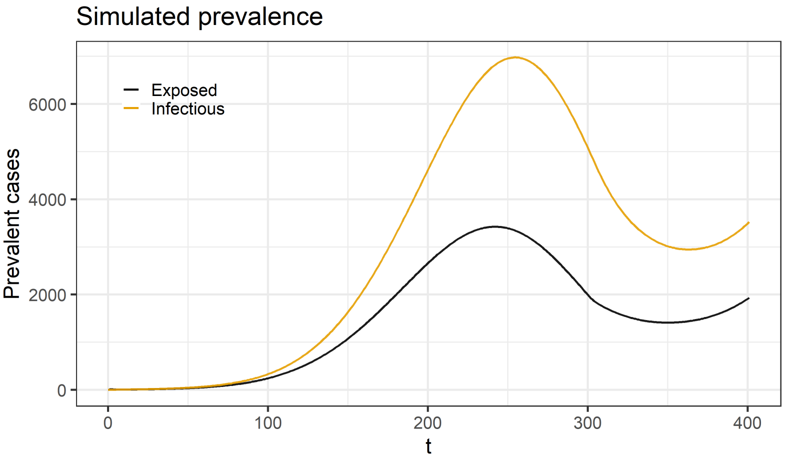

Figure 4: A plot of modeled prevalence over time. As all transitions aside from observations are deterministic, the number of infectious and exposed (non-infectious/in incubation period) individuals are also deterministic.

4.2 Smoothing and Estimation

The first step in the estimation algorithm is an initial filtered of the incidence data to reduce noise and periodicity. See the r(t) estimation section for more details.

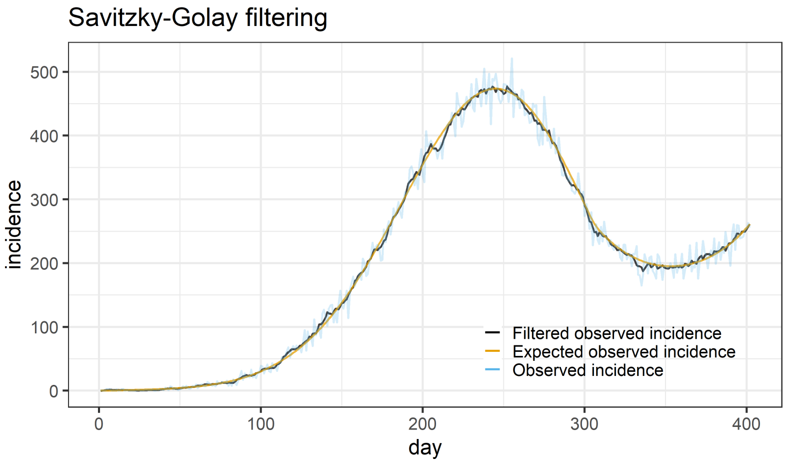

Figure 5: Application of a first-pass Savitzky-Golay filter substantially reduces noise and periodicity. The ”expected observed” data is shown as a comparison, and is the incidence of infection forward-convolved by the mean infection-observation delay distribution.

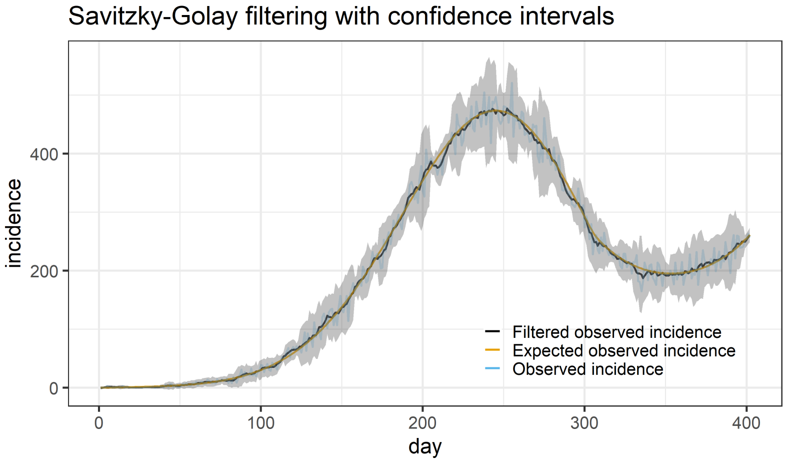

Figure 6: The same first-pass Savitzky-Golay filter as shown in Figure 7, with the confidence intervals shown.

The next step in the estimation algorithm produces r(t) estimates from the filtered data. Window sizes can be chosen arbitrarily, and have been set to 7 and 15 days in the following figures.

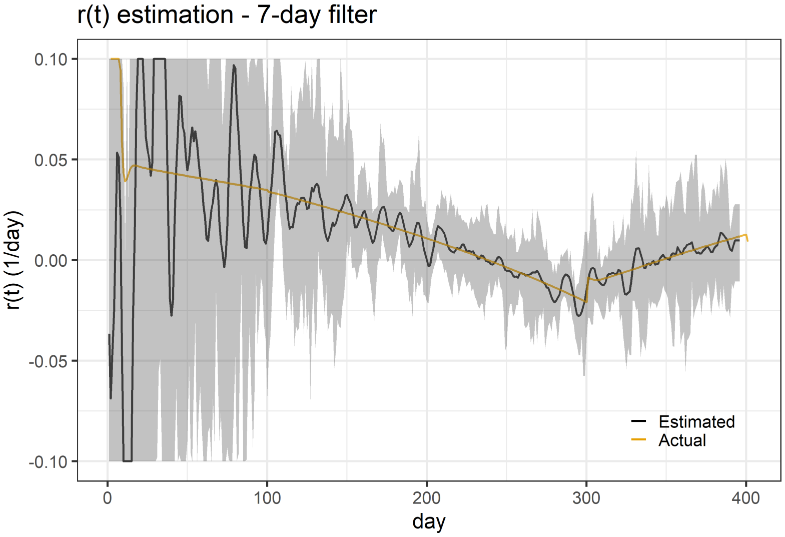

Figure 7: r(t) estimation using simulated data, with a 7-day post-filtering Savitzky-Golay window size. All values are clipped between -0.1 and 0.1 for visualization purposes. The true r(t) is within the 95% confidence interval 99% of the time.

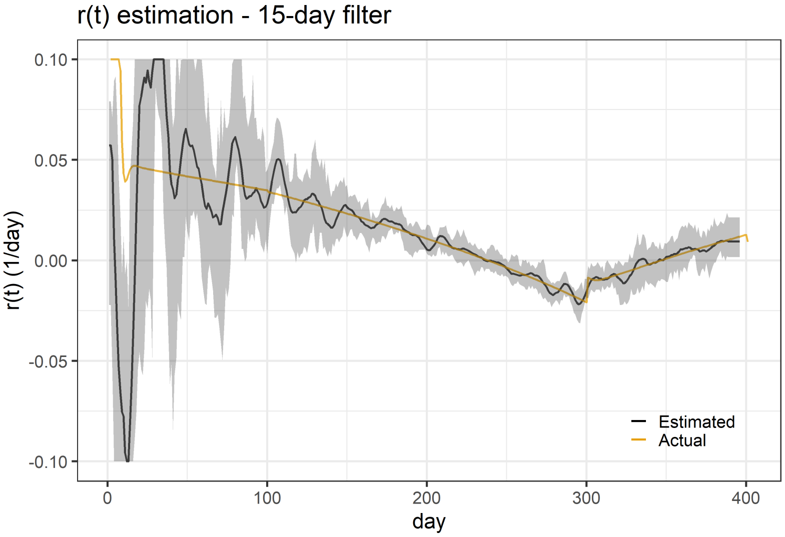

Figure 8: r(t) estimation using simulated data, with a 15-day post-filtering Savitzky-Golay window size. All values are clipped between -0.1 and 0.1 for visualization purposes. The true r(t) is within the 95% confidence interval 94% of the time.

4.3 Real-world data

While ground-truth r(t) values are unavailable for real-world data, this method can still be evaluated for qualitative stability using real-world data.

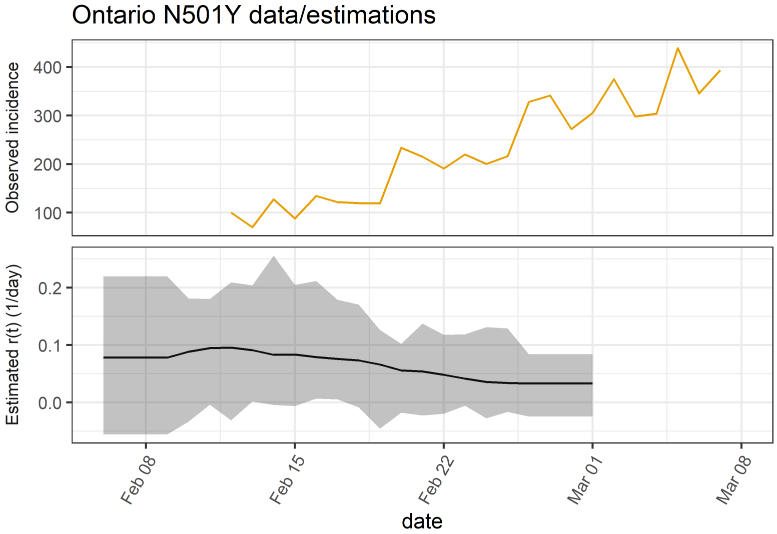

Figure 9: r(t) estimation using estimated N501Y variant data from Ontario, courtesy of Michael Li [7]. A 7-day post-filtering method and mean observation delay of 6 days was used.

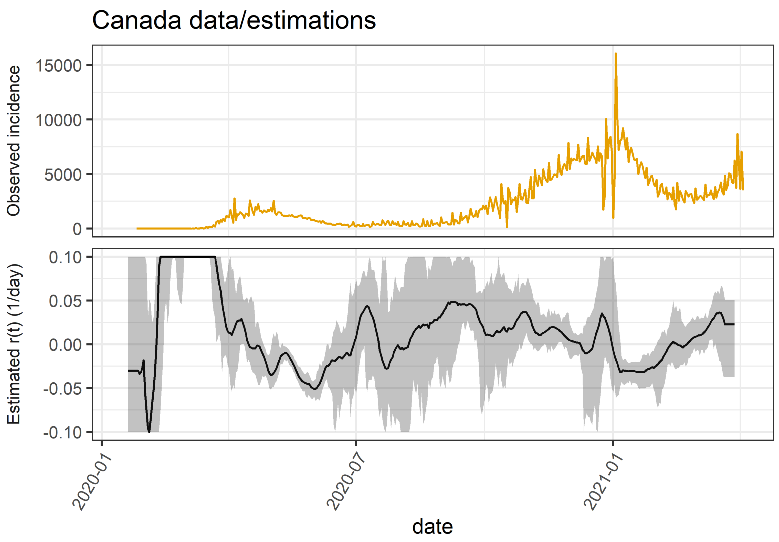

Figure 10: r(t) estimation using Canadian incidence data from OWID [1]. A 15-day post-filtering method and mean observation delay of 6 days was used. r(t) estimates are clipped between -0.1 and 0.1 for visualization purposes. (Note that during the period where estimates are cut off, the confidence interval is cut off as well.)

References

[1] Roser, M., Ritchie, H., Ortiz-Ospina, E., & Hasell, J. (2020). Coronavirus Pandemic (COVID-19). OurWorldInData.org. Retrieved 24 February 2021, from https://ourworldindata.org/coronavirus.

[2] McAloon, C., Collins, Á., Hunt, K., Barber, A., Byrne, A., & Butler, F. et al. (2020). Incubation period of COVID-19: a rapid systematic review and meta-analysis of observational research. BMJ Open, 10(8), e039652. https://doi.org/10.1136/bmjopen-2020-039652

[3] R: The Negative Binomial Distribution. Stat.ethz.ch. (2021). Retrieved 24 February 2021, from https://stat.ethz.ch/R-manual/R-devel/library/stats/html/NegBinomial.html.

[4] Pollock, A., & Lancaster, J. (2020). Asymptomatic transmission of covid-19. BMJ, m4851. https://doi.org/10.1136/bmj.m4851

[5] Gostic, K., McGough, L., Baskerville, E., Abbott, S., Joshi, K., & Tedijanto, C. et al. (2020). Practical considerations for measuring the effective reproductive number, Rt. PLOS Computational Biology, 16(12), e1008409. https://doi.org/10.1371/journal.pcbi.1008409

[6] Dey, N., Blanc-Féraud, L., Zimmer, C., Roux, P., Kam, Z., Olivo-Marin, J., & Zerubia, J. (2004). 3D Microscopy Deconvolution using Richardson-Lucy Algorithm with Total Variation Regularization. INRIA. Retrieved from https://hal.inria.fr/inria-00070726/document

[7] Li, M. (2021). COVID19-Canada. github.com. Retrieved 10 March 2021, from https://wzmli.github.io/COVID19-Canada/.