# Efficient Approaches to Modeling Viral Kinetics

A Rough Project Update

Violet Ross

January 2026

## Motivating Problems

**Current modeling approaches assume:** To fit a model of viral kinetics, you need to measure viral load regularly from infection time to clearance time in many individuals.
**Problems with this assumption:**

- Finding infections: Finding an infected individual is expensive. (It is even more expensive to find such an individual *before* they get infected, but this is necessary for the assumed modeling approach.)
- Learning from infections: Current models take in regular (e.g. daily) measurements of viral RNA concentration and output a fit model. However, taking daily measurements of viral load from an infected individual is expensive.

Basically . . . everything is expensive!
**Our goal:** Reduce cost (specifically with regard to testing).

## On Sampling and Testing

*Sampling* and *testing* are not equivalent in this report. When I discuss *sampling* from an individual, I mean only the act of collecting a sample (e.g. saliva) from them. This is does not include taking measurements of or performing tests on the sample. When I *test* a sample, I measure the concentration of virus in it (e.g. through a qt-PCR test). There are costs associated with both sampling and testing. We say testing a sample incurs cost $\omega$. This work describes an approach to minimize the testing cost associated with viral kinetics modeling; we assume some sampling work has been done upfront. However, we foresee this work having natural extensions which focus on lowering sampling costs.

# 1 Finding Infections

## The Problem

We have collected daily samples from $m$ individuals over $n$ days. Our samples are arranged into an $n$ by $m$ matrix. Find out who was sick and when.

## Brute Force

Test every sample. Cost incurred is $m \cdot n \cdot \omega$.

## Our Approach

We propose the following pooling approach, which offers a reduction in cost while still detecting 95% of the infections.
**Given**: matrix of samples coming from $m$ people over $n$ days.
**Assume**: Each infection $i$ has length $\ell_i$ drawn from some continuous positive distribution where $\ell_i > \ell_0$ with probability 0.95.

For each person:

1. Combine all $n$ samples into a single pool and test that pool.
2. If the test in step 1 is positive, then individually test that person's samples in increments of $\ell_0$. That is, test time 0, time $\ell_0$, time $2\ell_0$, etc.

## Cost

Applying our approach to a matrix of $m$ individuals over $n$ days with cumulative incidence $c$ incurs cost

$$\begin{aligned}
\text{cost} &= \begin{pmatrix} \text{num} \\ \text{people} \end{pmatrix} \begin{pmatrix} \text{cost per} \\ \text{person} \end{pmatrix} + \begin{pmatrix} \text{num positive} \\ \text{people} \end{pmatrix} \begin{pmatrix} \text{cost per positive} \\ \text{person} \end{pmatrix} \\
&= m(1) + cm\frac{n}{\ell_0} \\
&= mn\left(\frac{1}{n} + \frac{c}{\ell_0}\right)
\end{aligned}$$

## Accuracy Guarantees

Under this approach, with what probability will we miss a trajectory?

$$\begin{aligned}
P(\text{miss traj } i) &= P(\ell_i < \ell_0) P(\text{miss traj } i | \ell_i < \ell_0) \\
&= \int_0^\infty P(\ell_i) P(\ell_i < \ell_0) P(\text{miss} | \ell_i < \ell_0) d\ell_i \\
&= \int_0^{\ell_0} P(\ell_i) P(\text{miss} | \ell_i < \ell_0) d\ell_i \\
&= \int_0^{\ell_0} P(\ell_i)(1 - \frac{\ell_i}{\ell_0}) d\ell_i \\
&= \int_0^{\ell_0} P(\ell_i) d\ell_i - \int_0^{\ell_0} P(\ell_i) \frac{\ell_i}{\ell_0} d\ell_i \\
&= P(\ell_i < \ell_0) - \int_0^{\ell_0} P(\ell_i) \frac{\ell_i}{\ell_0} d\ell_i \\
&= 0.05 - \int_0^{\ell_0} P(\ell_i) \frac{\ell_i}{\ell_0} d\ell_i
\end{aligned}$$

The actual value of this probability depends on the distribution from which $\ell_i$ is drawn.

**Example 1.1 (Exponential Distribution)** *Suppose* $\ell_i \sim Expo(\lambda)$.

$$\begin{aligned}
P(\textit{miss traj } i) &= 0.05 - \frac{1}{\ell_0} \int_0^{\ell_0} P(\ell_i) \ell_i d\ell_i \\
&= 0.05 - \frac{1}{\ell_0} \int_0^{\ell_0} \lambda e^{-\lambda \ell_i} \ell_i d\ell_i \\
&= 0.05 - \frac{1}{\ell_0} \lambda \int_0^{\ell_0} e^{-\lambda \ell_i} \ell_i d\ell_i \\
&= 0.05 - \left[ \left( -\frac{1}{\lambda} \ell_i - \frac{1}{\lambda^2} \right) e^{-\lambda \ell_i} \right]_0^{\ell_i} \\
&= 0.05 - \frac{\lambda}{\ell_0} \left( \left( -\frac{1}{\lambda} \ell_0 - \frac{1}{\lambda^2} \right) e^{-\lambda \ell_0} + \frac{1}{\lambda^2} \right) \\
&= 0.05 + e^{-\lambda \ell_0} + \frac{1}{\ell_0 \lambda} e^{-\lambda \ell_0} - \frac{1}{\ell_0 \lambda}
\end{aligned}$$

*Recall our guarantee that* $\ell_i > \ell_0$ *with probability* 0.95. *In order to achieve this, we set*

$$\lambda = -\ln(0.95)/\ell_0.$$

*Substituting this into our equation for* $P(\textit{miss traj } i)$, *we get*

$$P(\textit{miss traj } i) \approx 0.0252.$$

*Figure 1 shows the results of our approach on simulated data with $\ell_i \sim Expo(\lambda)$. The proportion of misses in the simulation align with the above analytical result*[1]*. See simulation code here.*

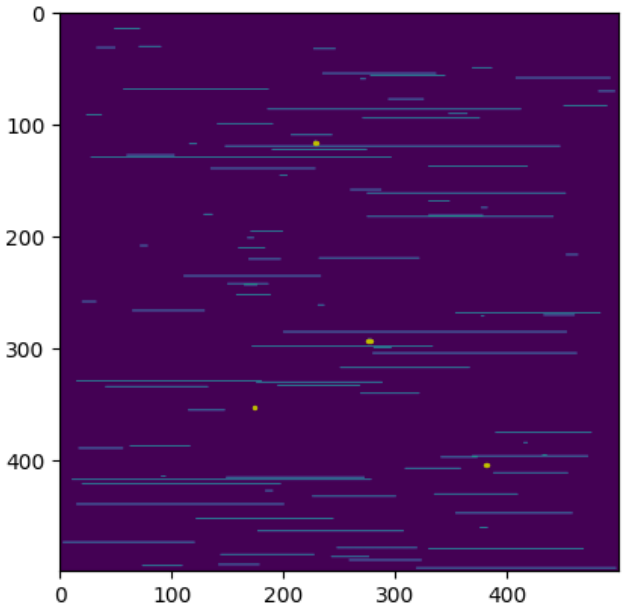


Figure 1: The results of our approach on simulated data represented as a matrix of individuals (rows) by days (columns) and colored by infection and detection status. Purple indicates that the individual was not sick on that day. Blue indicates that the individual was sick on that day, and our approach identified this infection. Yellow indicates that the individual was sick on that day, and our approach did not identify this infection. The simulation assumes exponentially distributed duration of infection with $\lambda = -\ln(0.95)/4$ and a cumulative incidence of 0.2.

# 2 Learning from Infections

## 2.1 Fitting to a Single Infection

### 2.1.1 The Problem

We have collected regular samples from an infected individual over the entire trajectory of their infection. How many, and which, of these samples should we test to fit a model of viral kinetics to this infection?

[1]Note that the few trajectories we do miss are very short. By not accounting for these trajectories in our model, we may bias our parameters in favor of longer trajectories.

### 2.1.2 The Model

We model viral kinetics using a two-phase segmented regression model (sometimes called a "tent function"). Descriptions of segmented regression and justifications of its application to viral kinetics modeling abound. Recall that

$\alpha$ is the y-intercept of the first line segment

$\beta_1$ is the slope of the first line segment

$\beta_2$ is the difference between the slope of the second line segment and the slope of the first

$\psi$ is the x-value of the breakpoint (the location where the two segments meet)

Our approach to fitting a segmented regression model to tested samples and their timestamps, given a step size by which to increment breakpoint estimates, is given in Algorithm 1. The function ESTIMATE_COEFFICIENTS performs simple linear regression on the proliferation phase (before breakpoint) and clearance phase (after breakpoint) to compute the coefficient estimates $\hat{\alpha}, \hat{\beta_1}$, and $\hat{\beta_2}$.

**Algorithm 1** Estimate Coefficients and Breakpoint

**Input:**
array of timestamps and measurements $P$ where $P[i]$ = [time of $i$th sample, viral load in $i$th sample],
stepsize for breakpoint estimation $c$
**Output:** estimates $(\hat{\alpha}, \hat{\beta_1}, \hat{\beta_2}, \psi)$

1: min_sse $\leftarrow \infty$
2: **for** $b$ in range(SECOND_MIN($P_{\text{time}}$), SECOND_MAX($P_{\text{time}}$), by = c) **do**
3: $\quad \hat{\alpha}^{(b)}, \hat{\beta_1}^{(b)}, \hat{\beta_2}^{(b)} \leftarrow$ ESTIMATE_COEFFICIENTS(P, b)
4: $\quad$ sse $\leftarrow$ COMPUTE_SSE($\hat{\alpha}^{(b)}, \hat{\beta_1}^{(b)}, \hat{\beta_2}^{(b)}, b, P$)
5: $\quad$ **if** sse $<$ min_sse **then**
6: $\quad\quad$ min_sse $\leftarrow$ sse
7: $\quad\quad$ parameters $\leftarrow (\hat{\alpha}^{(b)}, \hat{\beta_1}^{(b)}, \hat{\beta_2}^{(b)}, b)$
**return** parameters

### 2.1.3 Uniform Random Sampling

The input to Algorithm 1 is a set of tested samples, so before we apply the algorithm, we must decide which samples to test. One approach to selecting these samples is uniform random sampling. The algorithm for selecting and testing $m$ samples through uniform random sampling is given in Algorithm 2. We set the initial condition in line 1 to meet the requirement of that Algorithm 1 that $P$ contains at least 4 points.

**Algorithm 2** Uniform Random Sampling

**Input:**
array of sample timestamps $S$ where $S[i]$ = [time of $i$th sample],
number of samples to measure $m \geq 4$,
stepsize for breakpoint estimation $c$
**Output:** estimates $(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2, \hat{\psi})$

```
P ← (S[0], S[len(S)//3], S[(len(S)//3) * 2], S[−1])
for i in range(0, m - 4) do
    select s ∈ S \ P u.a.r
    P ← append s to P
P ← MEASURE_ALL(P)
return ESTIMATE(P, c)
```

### 2.1.4 Varmin Sampling

Think back to simple linear regression. We assume points are generated by the underlying model

$$y = \alpha + \beta_1 x + \varepsilon$$

where $x$ and $y$ are the independent and dependent variables, $\alpha$ is the y-intercept of the line, $\beta_1$ is the slope of the line, and $\varepsilon$ is random error whose distribution has mean 0 and variance $\sigma^2$. We then estimate $\alpha$ and $\beta_1$ from the data $(x_1, y_1), (x_2, y_2), \ldots$ using the least squares estimates

$$\hat{\alpha} = \bar{y} - \hat{\beta}_1 \bar{x}$$
$$\hat{\beta}_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}.$$

The variance of the slope estimate is

$$\mathrm{Var}(\hat{\beta}_1) = \frac{\sigma^2}{\sum_i (x_i - \bar{x})^2}.$$

So the more spread out our $x_i$, the less variable our $\beta_1$ estimator. Note that the value of the dependent variable does not play a role in the variance calculation. Can we extend this result to segmented regression, getting better estimation by testing the samples whose positions on the time axis minimize variance?

The equation for two-phase segmented regression looks similar to simple linear regression, but with an additional term to represent the change in slope following the breakpoint:

$$y = \alpha + \beta_1 x + \beta_2 \mathbb{1}_{\{x > \psi\}} + \varepsilon$$

where $\mathbb{1}_{\{x>\psi\}}$ is the indicator that equals 1 when x is past the breakpoint and

0 otherwise. Then the variances of the $\beta$ estimates are

$$\mathrm{Var}(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i:x_i<\psi}(x_i - \bar{x}_-)^2}$$
$$\mathrm{Var}(\hat{\beta}_2) = \frac{\sigma^2}{\sum_{i:x_i<\psi}(x_i - \bar{x}_-)^2} + \frac{\sigma^2}{\sum_{i:x_i>\psi}(x_i - \bar{x}_+)^2}$$

where $\bar{x}_-$ and $\bar{x}_+$ are the pre- and post-breakpoint means, respectively. We design an algorithm that, given $m$, selects the subset of $m$ points that minimizes the sum of the $\hat{\beta}$ variances

$$\mathrm{Var}(\hat{\beta}_1) + \mathrm{Var}(\hat{\beta}_1) = 2\frac{\sigma^2}{\sum_{i:x_i<\psi}(x_i - \bar{x}_-)^2} + \frac{\sigma^2}{\sum_{i:x_i>\psi}(x_i - \bar{x}_+)^2}.$$

The algorithm (Algorithm 3) initializes the subset to include four points and estimates an initial breakpoint $b$ using those four points. The algorithm then proceeds by, at each step, adding the sample which minimizes the sum of variances (computed using $b$) for the subset and then setting $b$ equal to a new breakpoint estimate generated from the new subset.

**Algorithm 3** Smart Varmin Sampling

**Input:**
array of sample timestamps $S$ where $S[i]$ = [time of $i$th sample],
number of samples to measure $m \geq 4$,
stepsize for breakpoint estimation $c$
**Output:** estimates $(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2, \hat{\psi})$

```
P ← (S[0], S[len(S)//3], S[(len(S)//3) * 2], S[−1])
M ← MEASURE_ALL(P)
α̂^(0), β̂_1^(0), β̂_2^(0), b^(0) ← ESTIMATE(M, c)
b ← b^(0)
min_var ← ∞
for i in range(0, m - 4) do
    for s in S \ P do
        combo ← append s to P
        slope_var ← COMPUTE_SLOPE_VARIANCE(b, combo)
        if slope_var < min_var then
            min_var ← slope_var
            optimal_combo ← combo
    P ← optimal_combo
    M ← MEASURE_ALL(P)
    α̂^(i+1), β̂_1^(i+1), β̂_2^(i+1), b^(i+1) ← ESTIMATE(M, c)
    b ← b^(i)
return ESTIMATE(P, c)
```

### 2.1.5 Comparing Algorithms

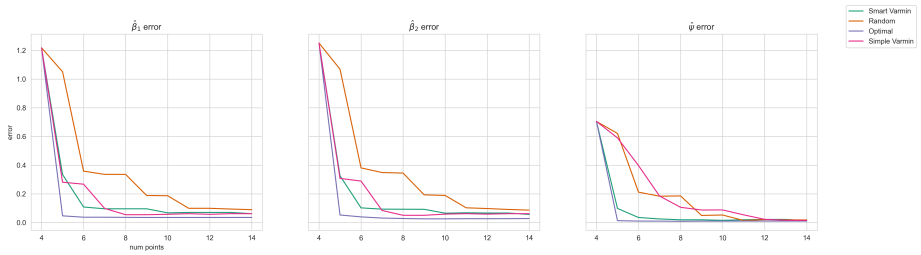


Figure 2: The averaged performance of each of the four sampling and fitting approaches over five trajectories. Each trajectory was stochastically generated, after which 20 points were sampled from it with normally distributed error (standard deviation = 0.1).

# Discussion

## Completed Steps

- Developed an efficient and accurate approaches for Phases 1 (finding an infection) and 2 (fitting a model to an infection).
- Submitted a GRFP on this topic.

## Next Steps

- In Phase 1, we locate an infection. In Phase 2, we skip a step and assume that the start and end times of the infection are known. Can we **fit a segmented regression model starting only with a single positive point?**
- Right now, our approaches are designed to learn a model of a single trajectory. Can we **design a (Bayesian Hierarchical) approach which learns about the distributions from which the trajectories are generated?** That is, can we say something about the disease dynamics in general, rather than just discretely describing dynamics within individuals?
  - This also helps us move from descriptive modeling land to predictive modeling land
- Can we **translate these mathematical results into an explicit approach** to collecting and processing data for within-host viral kinetics research?
  - Open-source software?