Efficient Approaches to Modeling Viral Kinetics

Progress Report

Violet Ross

January 2026

Early sketch of project update. NOT complete.

Motivating Problems

Existing assumption: 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: Make each component less expensive!

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 ω. 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 [BLANK] reduction in cost while still detecting [BLANK] of the infections.
Given: matrix of samples coming from m people over n days.
Assume: Each infection i has length ℓ_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 ℓ_0. That is, test time 0, time ℓ_0, time $2\ell_0$, etc.

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 ℓ_i is drawn.

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

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

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 $P(miss\ traj\ i)$, we get

$$P(miss\ traj\ i) \approx 0.0252.$$

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.*

[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.

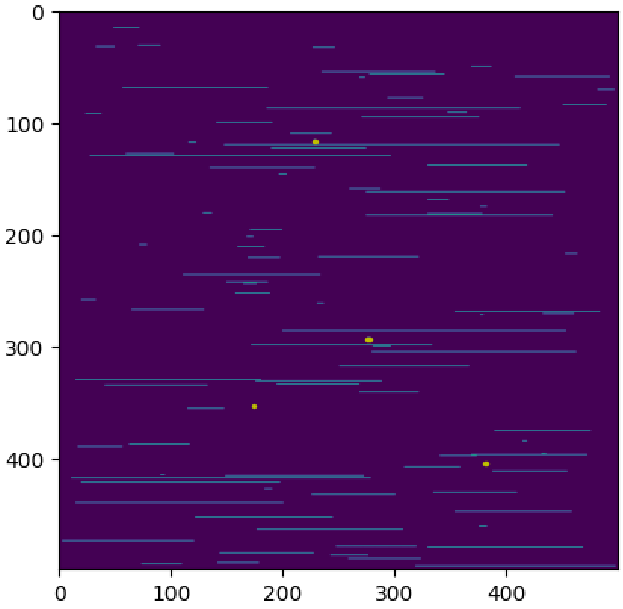

Figure 1: The results of our approach on simulated data represented as a matrix of individuals (rows) by days (columns) and colored by infection and detection status. Purple indicates that the individual was not sick on that day. Blue indicates that the individual was sick on that day, and our approach 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

Simplest Approach: Measure viral load every day, fit segmented regression (tent function) to all of the daily measurements.

A little faster: Test only some of the days, fit segmented regression (tent function) to this smaller set of measurements.

Fastest: Test on the days that minimize the variance of the slope estimators.

2.1.1 Segmented Regression

What is segmented regression? We assume two-phase segmented regression specifically.

Why do we use segmented regression?

Our approaches to modeling a single infection vary in their selection of which points (i.e. samples) to test, but not in the process through which a model is fit to those points. Our approach to model fitting once the selected points have been measured is as follows. In order to estimate the slopes of the first and second phases of regression, we must know when the transition between those two phases is. That is, we must know the location of the breakpoint. Our model fitting approach is based on this necessity.

Algorithm 1 Estimate

Input:
array of timestamps and measurements P where $P[i]$ = [time of ith sample, viral load in ith sample],
stepsize for breakpoint estimation c
Output: estimates $(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2, \psi)$

```
min_sse ← $\infty$
for $b$ in range(SECOND_MIN($P$), SECOND_MAX($P$), by = c) do
    $\hat{\alpha}^{(b)}, \hat{\beta}_1^{(b)}, \hat{\beta}_2^{(b)}$ ← ESTIMATE_COEFFICIENTS(P, b)
    sse ← COMPUTE_SSE($\hat{\alpha}^{(b)}, \hat{\beta}_1^{(b)}, \hat{\beta}_2^{(b)}, b, P$)
    if sse < min_sse then
        min_sse ← sse
        parameters ← $(\hat{\alpha}^{(b)}, \hat{\beta}_1^{(b)}, \hat{\beta}_2^{(b)}, b)$
return parameters
```

2.1.2 Uniform Random Sampling

Intuition: What if, instead of testing all n samples, we tested a subset of m samples, chosen uniformly at random?

Algorithm 2 Uniform Random Sampling

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

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

2.1.3 Varmin Sampling

Intuition: Let's choose the size m subset by selecting the samples which would minimize the variance of our slope estimates.

Algorithm 3 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 $(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2, \hat{\psi})$
  $P \leftarrow (S[0],\ S[\text{len}(S)//3],\ S[(\text{len(S)}//3) * 2],\ S[-1])$
  $M \leftarrow$ MEASURE_ALL($P$)
  $\hat{\alpha}^{(0)}, \hat{\beta_1}^{(0)}, \hat{\beta_2}^{(0)}, b^{(0)} \leftarrow$ ESTIMATE($M, c$)
  $b \leftarrow b^{(0)}$
  min_var $\leftarrow \infty$
  for $i$ in range(0, m - 4) do
      for $s$ in $S \setminus P$ do
          combo $\leftarrow$ append $s$ to $P$
          slope_var $\leftarrow$ COMPUTE_SLOPE_VARIANCE($b$, combo)
          if slope_var $<$ min_var then
              min_var $\leftarrow$ slope_var
              optimal_combo $\leftarrow$ combo
      $P \leftarrow$ optimal_combo
      $M \leftarrow$ MEASURE_ALL($P$)
      $\hat{\alpha}^{(i+1)}, \hat{\beta_1}^{(i+1)}, \hat{\beta_2}^{(i+1)}, b^{(i+1)} \leftarrow$ ESTIMATE($M, c$)
      $b \leftarrow b^{(i)}$
  return ESTIMATE($P, c$)
```

2.1.4 Comparing Algorithms

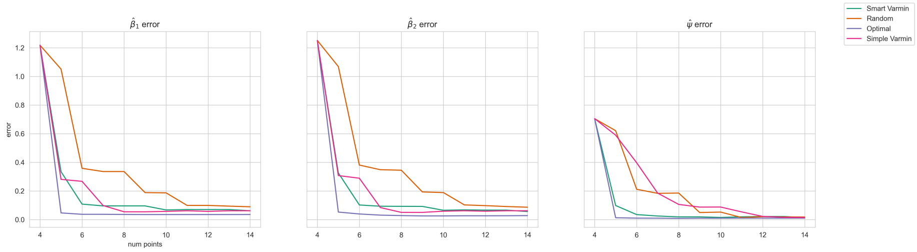

Figure 2: The averaged performance of each of the four sampling and fitting approaches over five trajectories. Each trajectory was generated according to [BLANK], after which 20 points were sampled from it with normally distributed error (standard deviation = 0.1).

Accomplished

- Began working in a brand new application area (ID Epi) and learned a lot!
- Submitted a GRFP on this topic
- Optimal approach for 1, 2, 3, done.

References