Supplementary Material: Efficient Transition Probability Computation for Continuous-Time Branching Processes via Compressed Sensing

Jason Xu[1], Vladimir N. Minin[1,2]
[1]Department of Statistics, University of Washington, Seattle, WA, U.S.A.
[2]Department of Biology, University of Washington, Seattle, WA, U.S.A.

1 Discrete Fourier matrix

The N by N discrete Fourier transform matrix $\mathbf{F}_N$ has entries

$$\{\mathbf{F}_N\}_{j,k} = \frac{1}{\sqrt{N}}(\omega)^{jk}$$

with $j, k = 0, 1, \ldots, N-1$ and $\omega = e^{i2\pi/N}$, and as we mention in the main paper, the inverse Fourier transform matrix $\boldsymbol{\psi}$ is given by its conjugate transpose. The partial M by N IDFT matrices $\mathbf{A}$ necessary in Algorithm 1 are obtained by computing and stacking only a subset of M random rows from $\boldsymbol{\psi}$.

2 Line search subroutine

We select step sizes with a simple line search algorithm summarized in the pseudocode below that works by evaluating an easily computed upper bound $\hat{f}$ on the objective f:

$$\hat{f}_L(Z, Y) := f(Y) + \nabla f(Y)^T (Z - Y) + \frac{L}{2}||Z - Y||_2^2. \tag{1}$$

We follow Beck and Teboulle [2009], who provide further details. In implementation, we select $L = 5 \times 10^{-6}$ and $c = .5$, and reuse the gradient computed in `line-search` for step 10 of Algorithm 1 in the main paper: no additional evaluations of ∇g are required for line-search.

Algorithm 1 `line-search` procedure.

```
**Input:** initial step size $L$, shrinking factor $c$, matrices $Y_k, \nabla g(Y_k)$.
Set $Z = \text{softh}(Y_k - L\nabla g(Y_k))$
**while** $g(Z) > \hat{f}_L(Z, Y_k)$ **do**
    Update $L = cL$
**end while**
**return** $L_k = L$
```

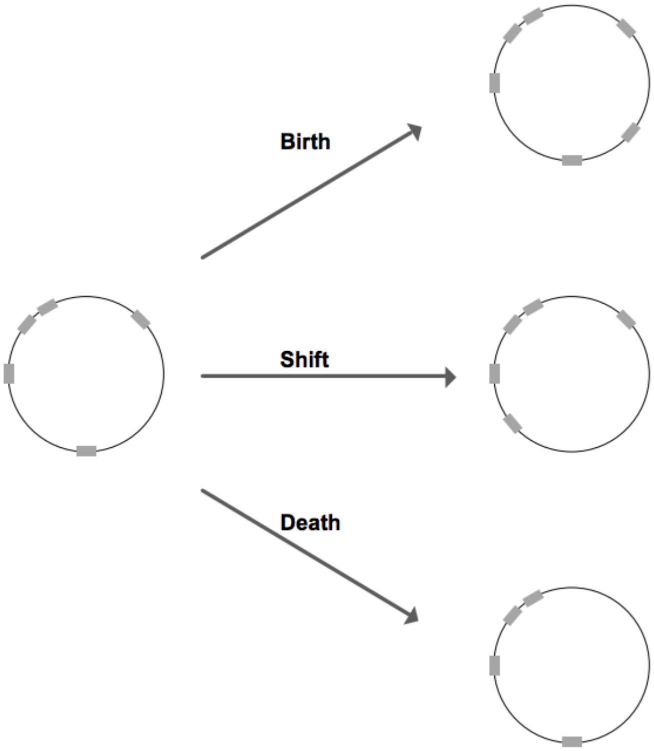

Figure 1: Illustration of the three types of transposition—birth, death, shift—along a genome, represented by circles [Rosenberg et al., 2003]. Transposons are depicted by rectangles occupying locations along the circles/genomes. On the right set of diagrams, a birth event keeps the number of type 1 particles intact and increments the number of type 2 particles by one, a death event changes the number of type 1 particles from five to four and keeps the number of type 2 particles at zero, and finally a shift event decreases the number of type 1 particles by one and increases the number of type 2 particles by one.

3 BDS model diagram

The branching process components $\mathbf{X}(t) = (x_{old}, x_{new})$ represent the number of originally occupied and newly occupied sites at the end of each observation interval. As an example, assume six particles (transposons) are present initially at time t_0, and a shift and a birth occur before the first observation t_1, and a death occurs before a second observation at t_2. When considering the first observation interval $[t_0, t_1)$, we have $\{\mathbf{X}(t_0) = (6, 0), \mathbf{X}(t_1) = (5, 2)\}$. When computing the next transition probability over $[t_1, t_2)$, we now have $\{\mathbf{X}(t_1) = (7, 0), \mathbf{X}(t_2) = (6, 0)\}$, since all seven of the particles at t_1, now the left endpoint of the observation interval, now become the initial population. Even with data over time, this seeming inconsistency at the endpoints does not become a problem because transition probability computations occur separately over disjoint observation intervals. See Xu et al. [2014] for further details.

4 Derivation for hematopoiesis process PGF

Given a two-type branching process defined by instantaneous rates $a_i(k,l)$, denote the following *pseudo-generating* functions for $i = 1, 2$:

$$u_i(s_1, s_2) = \sum_k \sum_l a_i(k,l) s_1^k s_2^l$$

We may expand the probability generating functions in the following form:

$$\begin{aligned}
\phi_{10}(t, s_1, s_2) &= E(s_1^{X_1(t)} s_2^{X_2(t)} | X_1(0) = 1, X_2(0) = 0) \\
&= \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} P_{(1,0),(k,l)}(t) s_1^k s_2^l \\
&= \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} (\mathbf{1}_{k=1,l=0} + a_1(k,l)t + o(t)) s_1^k s_2^l \\
&= s_1 + u_1(s_1, s_2)t + o(t).
\end{aligned}$$

We have an analogous expression for $\phi_{01}(t, s_1, s_2)$ beginning with one particle of type 2 instead of type 1. For short, we will write $\phi_{10} := \phi_1, \phi_{01} := \phi_2$.

Thus we have the following relation between the functions ϕ and u:

$$\frac{d\phi_1}{dt}(t, s_1, s_2)|_{t=0} = u_1(s_1, s_2)$$

$$\frac{d\phi_2}{dt}(t, s_1, s_2)|_{t=0} = u_2(s_1, s_2)$$

To derive the backwards and forward equations, Chapman-Kolmogorov arguments yield the symmetric relations

$$\phi_1(t+h, s_1, s_2) = \phi_1(t, \phi_1(h, s_1, s_2), \phi_2(h, s_1, s_2)) \tag{2}$$

$$= \phi_1(h, \phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2)) \tag{3}$$

First, we derive the backward equations by expanding around t and applying (2):

$$\begin{aligned}
\phi_1(t+h, s_1, s_2) &= \phi_1(t, s_1, s_2) + \frac{d\phi_1}{dh}(t+h, s_1, s_2)|_{h=0} h + o(h) \\
&= \phi_1(t, s_1, s_2) + \frac{d\phi_1}{dh}(h, \phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2)|_{h=0} h + o(h) \\
&= \phi_1(t, s_1, s_2) + u_1(\phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2)h + o(h))
\end{aligned}$$

Since an analogous argument applies for ϕ_2, we arrive at the system

$$\begin{cases}
\frac{d}{dt}\phi_1(t, s_1, s_2) = u_1(\phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2)) \\
\frac{d}{dt}\phi_2(t, s_1, s_2) = u_2(\phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2))
\end{cases}$$

with initial conditions $\phi_1(0, s_1, s_2) = s_1, \phi_2(0, s_1, s_2) = s_2$.

Recall the rates defining the two-compartment hematopoiesis model are given by

$$\begin{array}{lll}
a_1(2,0) = \rho & a_1(0,1) = \nu & a_1(1,0) = -(\rho + \nu) \\
a_2(0,0) = \mu & a_2(0,1) = -\mu &
\end{array}$$

Thus, the pseudo-generating functions are

$$u_1(s_1, s_2) = \rho s_1^2 + \nu s_2 - (\rho + \nu)s_1$$

$$u_2(s_1, s_2) = \mu - \mu s_2 = \mu(1 - s_2)$$

Plugging into the backward equations, we obtain

$$\frac{d}{dt}\phi_1(t, s_1, s_2) = \rho\phi_1^2(t, s_1, s_2) + \nu\phi_2(t, s_1, s_2) - (\rho + \nu)\phi_1(t, s_1, s_2)$$

and

$$\frac{d}{dt}\phi_2(t, s_1, s_2) = \mu - \mu\phi_2(t, s_1, s_2).$$

The ϕ_2 differential equation corresponds to a pure death process and is immediately solvable: suppressing the arguments of ϕ_2 for notational convenience, we obtain

$$\begin{aligned} \frac{d}{dt}\phi_2 &= \mu - \mu\phi_2 \\ \frac{d}{dt}\phi_2(\frac{1}{1 - \phi_2}) &= \mu \\ \ln(1 - \phi_2) &= -\mu t + C \\ \phi_2 &= 1 - \exp(-\mu t + C) \end{aligned}$$

Pluggin in $\phi_2(0, s_1, s_2) = s_2$, we obtain $C = \ln(1 - s_2)$, and we arrive at

$$\phi_2(t, s_1, s_2) = 1 + (s_2 - 1)\exp(-\mu t) \tag{4}$$

Plugging this solution into the other backward equation, we obtain

$$\frac{d}{dt}\phi_1(t, s_1, s_2) = \rho\phi_1^2(t, s_1, s_2) - (\rho + \nu)\phi_1(t, s_1, s_2) + \nu(1 + (s_2 - 1)\exp(-\mu t)) \tag{5}$$

This ordinary differential equation can be solved numerically given rates and values for the three arguments, allowing computation of $\phi_{i,j} = \phi_1^i\phi_2^j$ which holds by particle independence.

5 Relative Errors

To supplement the error analysis in the main paper, here we include a discussion of the relative errors

$$\varepsilon_{kl}^{\text{rel}} = \begin{cases} \frac{\varepsilon_{kl}}{S_{kl}} & S_{kl} \neq 0 \\ \varepsilon_{kl} & S_{kl} = 0 \end{cases}.$$

Most relative errors are extremely small but feature a few severe outliers due to dividing by negligibly small probabilities in **S** relative to tolerance set for PGD convergence. For instance, in one trial of the HSC model with $N = 256$, the median, mean, and maximum absolute values of relative errors $|\varepsilon_{kl}^{\text{rel}}|$ are 0, 0.00908, and 91— note the mean is still low despite outlier skew, and even the 99th percentile relative error is 0.000118, but the max error of 91 suggests the presence of rare but extreme outliers. Thus, while they are a more standard measure of accuracy than those reported in the main paper, relative errors provide limited insight to performance.

Nonetheless, we can again take a conservative look at accuracy by only considering relative errors among the nonzero values of **S** that comprise most of the total transition mass. In the aforementioned HSC example

Table 1: Summary statistics of absolute values of relative errors over transition probabilities comprising 98% of total mass, HSC model

N	Median	Mean	3rd Quartile	95 Percentile
128	0.013	0.023	0.030	0.080
256	0.021	0.068	0.093	0.27
512	0.039	0.094	0.14	0.33
1024	0.029	0.070	0.074	0.29
2048	0.043	0.098	0.13	0.38
4096	0.056	0.10	0.14	0.34

Table 2: Summary statistics of absolute values of relative errors over transition probabilities comprising 98% of total mass, BDS model

N	Median	Mean	3rd Quartile	95 Percentile
128	0.010	0.015	0.022	0.041
256	0.043	0.10	0.13	0.31
512	0.035	0.069	0.088	0.22
1024	0.10	0.16	0.24	0.42
2048	0.075	0.15	0.22	0.50
4096	0.083	0.13	0.19	0.41

with $N = 256, 221$ entries comprise 98% of support. This at once discards extreme numerical outliers while being conservative in that the many zero-valued entries of **S** do not favorably affect this measure. Table 1 and 2 include summary statistics of the absolute values of these restricted relative error measures for both models over 12 random restarts. We see that despite heavily restricting to a set that makes the relative errors look less favorable, relative errors are low overall in all cases.

6 Implementation

We provide open-source `R` code for CSGF since existing software in the `R` community for compressed sensing and ℓ_1 problems are inadequate for our purposes. For vector-valued signal recovery problems, we recommend package `R1magic` [Süzen, 2013], which provides implementations of several compressed sensing objectives using various norms. `R1magic` performs the optimization using `nlm`, which becomes prohibitively slow when the length of the solution vector to be recovered grows. Similarly, to our knowledge, `R` packages for LASSO and related problems– `glmnet` is one example providing extremely efficient procedures for many classes of regularized models– are not suited for matrix valued optimization in the form we require [Friedman et al., 2010, Simon et al., 2011]. As we have mentioned in the main paper, vectorizing our problem is inefficient and negates the performance gains achieved by CSGF.

`R` code containing examples and implementation of the CSGF algorithm using proximal gradient descent is available at `https://github.com/jasonxu90`. Users may readily replace the proximal gradient descent function with their optimization routine of choice best suited for a given application. The package `bdsem` used to evaluate ODE solutions for the BDS process is available at the same URL.

References

A Beck and M Teboulle. Gradient-based algorithms with applications to signal recovery. *Convex Optimization in Signal Processing and Communications*, 2009.

J Friedman, T Hastie, and R Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

NA Rosenberg, AG Tsolaki, and MM Tanaka. Estimating change rates of genetic markers using serial samples: applications to the transposon IS*6110* in *Mycobacterium tuberculosis*. *Theoretical Population Biology*, 63(4):347–363, 2003.

N Simon, J Friedman, T Hastie, and R Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011. URL `http://www.jstatsoft.org/v39/i05/`.

M. Süzen. R1magic: Compressive sampling: Sparse signal recovery utilities, 2013. URL `http://cran.r-project.org/web/packages/R1magic/index.html`.

J Xu, P Guttorp, MM Kato-Maeda, and VN Minin. Likelihood-based inference for discretely observed birth-death-shift processes, with applications to evolution of mobile genetic elements. *ArXiv e-prints*, arXiv:1411.0031, 2014.