Title

Efficient Transition Probability Computation for Continuous-Time Branching Processes via Compressed Sensing.

Authors
Xu, Jason
Minin, Vladimir N

Publication Date
2015-07-01

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

By Jason Xu*, Vladimir Minin*,†
*Department of Statistics, University of Washington
†Department of Biology, University of Washington

Abstract

Branching processes are a class of continuous-time Markov chains (CTMCs) with ubiquitous applications. A general difficulty in statistical inference under partially observed CTMC models arises in computing transition probabilities when the discrete state space is large or uncountable. Classical methods such as matrix exponentiation are infeasible for large or countably infinite state spaces, and sampling-based alternatives are computationally intensive, requiring a large integration step to impute over all possible hidden events. Recent work has successfully applied generating function techniques to computing transition probabilities for linear multi-type branching processes. While these techniques often require significantly fewer computations than matrix exponentiation, they also become prohibitive in applications with large populations. We propose a compressed sensing framework that significantly accelerates the generating function method, decreasing computational cost up to a logarithmic factor by only assuming the probability mass of transitions is sparse. We demonstrate accurate and efficient transition probability computations in branching process models for hematopoiesis and transposable element evolution.

1 Introduction

Continuous-time branching processes are widely used in stochastic modeling of population dynamics, with applications including biology, genetics, epidemiology, quantum optics, and nuclear fission [Renshaw, 2011]. With the exception of the well-studied class of birth-death processes, which have known expressions for many quantities relevant to probabilistic inference [Crawford et al., 2014], branching processes pose significant inferential challenges. In particular, closed forms for finite-time *transition probabilities*, the conditional probability that a trajectory ends at a given state, given a starting state and time interval, are unavailable. These transition probabilities are crucial in many inferential approaches, comprising the observed likelihood function when data from the process are available at a set of discrete times. The likelihood function is of central importance in frequentist and Bayesian methods, and any statistical framework involving observed data likelihood evaluation requires transition probability computations. Without the ability to fully leverage the branching structure, studies must rely on general CTMC estimation techniques or model approximations [Rosenberg et al., 2003, Golinelli et al., 2006, El-Hay et al., 2006].

Computation of transition probabilities is the usual bottleneck in model-based inference using CTMCs [Hajiaghayi et al., 2014], requiring a marginalization over the infinite set of possible end-point conditioned paths. Classically, this marginalization is accomplished by computing the matrix exponential of the infinitesimal generator of the CTMC. However, this procedure has cubic runtime complexity in the size of the state space, becoming prohibitive even for state spaces of moderate sizes. Alternatives also have their shortcomings: *uniformization* methods use a discrete-time "skeleton" chain to approximate the CTMC but rely on a restrictive assumption that there is a uniform bound on all rates [Grassmann, 1977, Rao and Teh, 2011]. Typically, practitioners resort to sampling-based approaches via Markov chain Monte Carlo (MCMC). Specifically, particle-based methods such as sequential Monte Carlo (SMC) and particle MCMC [Doucet et al., 2000, Andrieu et al., 2010] offer a complementary approach whose runtime depends on the number of imputed transitions rather than the size of the state space. However, these SMC methods have several limitations— in many applications, a prohibitively large number of particles is required to impute waiting times and events between transitions, and degeneracy issues are a common occurrence, especially in longer time series. A method by Hajiaghayi et al. [2014] accelerates particle-based methods by marginalizing holding times analytically, but has cubic runtime complexity in the number of imputed jumps between observations and is recommended for applications with fewer than one thousand events occurring between observations.

Recent work by Xu et al. [2014] has extended techniques for computing transition probabilities in birth-death models to linear multi-type branching processes. This approach involves expanding the probability generating function (PGF) of the process as a Fourier series, and applying a Riemann sum approximation to its inversion formula. This technique has been used to compute numerical transition probabilities within a maximum likelihood estimation (MLE) framework, and has also been applied within Expectation Maximization (EM) algorithms [Doss et al., 2013, Xu et al., 2014]. While this method provides a powerful alternative to simulation and avoids costly matrix operations, the Riemann approximation to the Fourier inversion formula requires $\mathcal{O}(N^b)$ PGF evaluations, where b is the number of particle types and N is the largest population size at endpoints of desired transition probabilities. This complexity is no worse than linear in the size of the state space, but can also be restrictive: a two-type process in which each population can take values in the thousands would require millions of PGF evaluations to produce transition probabilities over an observation interval. This can amount to hours of computation in standard computing architectures, because evaluating PGFs for multitype branching processes involves numerically solving systems of ordinary differential equations (ODEs). Such computations become infeasible within iterative algorithms.

In this paper, we focus our attention on the efficient computation of transition probabilities in the presence of sparsity, presenting a novel compressed sensing generating function (CSGF) algorithm that dramatically reduces the computational cost of inverting the PGF. We apply our algorithm to a branching process model used to study hematopoiesis as well as a birth-death-shift process with applications to molecular epidemiology, and see that the sparsity assumption is valid for scientifically realistic rates of the processes obtained in previous statistical studies. We compare performance of CSGF to transition probability computations without taking advantage of sparsity, demonstrating a high degree of accuracy while achieving significant improvements in runtime.

2 Markov Branching Processes

A linear multitype branching process follows a population of independently acting particles that reproduce and die. The random vector $\mathbf{X}(t)$ takes values in a discrete state space Ω at time t, with $X_i(t)$ denoting the number of type i particles present at time t. For exposition and notational simplicity, we will focus on the two-type case. In the continuous-time setting, each type i particle produces k type 1 particles and l type 2 particles with *instantaneous rates* $a_j(k,l)$, and the rates of no event occurring are defined as

$$\alpha_1 := a_1(1,0) = -\sum_{(k,l)\neq(1,0)} a_1(k,l),$$

$$\alpha_2 := a_2(0,1) = -\sum_{(k,l)\neq(0,1)} a_2(k,l)$$

so that $\sum_{k,l} a_i(k,l) = 0$ for $i = 1,2$. Offspring of each particle evolve according to the same set of instantaneous rates, and these rates $a_j(k,l)$ do not depend on t so that the process is *time-*

homogeneous. These assumptions imply that each type i particle has exponentially distributed lifespan with rate $-\alpha_i$, and $\mathbf{X}(t)$ evolves over time as a CTMC [Guttorp, 1995].

2.1 Transition probabilities

Dynamics of a CTMC are determined by its transition function

$$p_{\mathbf{x},\mathbf{y}}(t) = \Pr(\mathbf{X}(t+s) = \mathbf{y} | \mathbf{X}(s) = \mathbf{x}), \tag{1}$$

where time-homogeneity implies independence of the value of s on the right hand side. When the state space Ω is small, one can exponentiate the $|\Omega|$ by $|\Omega|$ *infinitesimal generator* or rate matrix $\mathbf{Q} = \left\{q_{\mathbf{x},\mathbf{y}}\right\}_{\mathbf{x},\mathbf{y}\in\Omega}$, where the entries $q_{\mathbf{x},\mathbf{y}}$ denote the instantaneous rates of jumping from state $\mathbf{x}$ to $\mathbf{y}$, to compute transition probabilities:

$$\mathbf{P}(t) := \left\{p_{\mathbf{x},\mathbf{y}}(t)\right\}_{\mathbf{x},\mathbf{y}\in\Omega} = e^{\mathbf{Q}t} = \sum_{k=0}^{\infty} \frac{(\mathbf{Q}t)^k}{k!}. \tag{2}$$

These transition probabilities are fundamental quantities in statistical inference for data generated from CTMCs. For instance, if $\mathbf{X}(t)$ is observed at times $t_1, \ldots, t_J$ and $\mathbf{D}$ represents the 2 by J matrix containing the observed data, the *observed log-likelihood* is given by

$$\ell_o(\mathbf{D}; \boldsymbol{\theta}) = \sum_{j=1}^{J-1} \log p_{\mathbf{X}(t_j),\mathbf{X}(t_{j+1})}(t_{j+1} - t_j; \boldsymbol{\theta}) \tag{3}$$

where the vector $\boldsymbol{\theta}$ parametrizes the rates $a_j(k,l)$. Maximum likelihood inference that seeks to find the value $\hat{\boldsymbol{\theta}}$ that optimizes (3) as well as Bayesian methods where likelihood calculations arise in working with the posterior density (up to a proportionality constant) fundamentally rely on the ability to calculate transition probabilities. Having established their importance in probabilistic inference, we focus our discussion in this paper to computing these transition probabilities in a continuous-time branching process.

2.2 Generating function methods

Matrix exponentiation is cubic in $|\Omega|$ and thus prohibitive in many applications, but we may take an alternate approach by exploiting properties of the branching process. Xu et al. [2014] extend a generating function technique used to compute transition probabilities in birth-death processes to the multi-type branching process setting. The probability generating function (PGF) for a two-type process is defined

$$\begin{aligned}\phi_{jk}(t, s_1, s_2; \boldsymbol{\theta}) &= \mathrm{E}_{\boldsymbol{\theta}}(s_1^{X_1(t)} s_2^{X_2(t)} | X_1(0) = j, X_2(0) = k) \\ &= \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} p_{(jk),(lm)}(t; \boldsymbol{\theta}) s_1^l s_2^m;\end{aligned} \tag{4}$$

this definition extends analogously for any m-type process. We suppress dependence on $\boldsymbol{\theta}$ for notational convenience. Bailey [1964] provides a general technique to derive a system of differential equations governing ϕ_{jk} using the Kolmogorov forward or backward equations given the instantaneous rates $a_j(k,l)$. It is often possible to solve these systems analytically for ϕ_{jk}, and even when closed forms are unavailable, numerical solutions can be efficiently obtained using standard algorithms such as Runge-Kutta methods [Butcher, 1987].

With ϕ_{jk} available, transition probabilities are related to the PGF (4) via differentiation:

$$p_{(jk),(lm)}(t) = \frac{\partial^l}{\partial s_1}\frac{\partial^m}{\partial s_2}\phi_{jk}(t)\bigg|_{s_1=s_2=0}. \tag{5}$$

This repeated differentiation is computationally intensive and numerically unstable for large l, m, but following Lange [1982], we can map the domain $s_1, s_2 \in [0,1] \times [0,1]$ to the boundary of the complex unit circle, instead setting $s_1 = e^{2\pi i w_1}, s_2 = e^{2\pi i w_2}$. The generating function becomes a Fourier series whose coefficients are the desired transition probabilities

$$\phi_{jk}(t, e^{2\pi i w_1}, e^{2\pi i w_2}) = \sum_{l,m=0}^{\infty} p_{(jk),(lm)}(t) e^{2\pi i l w_1} e^{2\pi i m w_2}$$

Applying a Riemann sum approximation to the Fourier inversion formula, we can now compute the transition probabilities via integration instead of differentiation:

$$\begin{aligned} p_{(jk),(lm)}(t) &= \int_0^1 \int_0^1 \phi_{jk}(t, e^{2\pi i w_1}, e^{2\pi i w_2}) e^{-2\pi i l w_1} \\ &\qquad \times e^{-2\pi i m w_2} dw_1 dw_2 \\ &\approx \frac{1}{N^2} \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \phi_{jk}(t, e^{2\pi i u/N}, e^{2\pi i v/N}) \\ &\qquad \times e^{-2\pi i l u/N} e^{-2\pi i m v/N}. \end{aligned} \tag{6}$$

In practice, the set of transition probabilities $S = \{p_{(jk),(lm)}(t)\}$ for all $l, m = 0, \ldots, N$, given initial values of (j, k), can be obtained via the Fast Fourier Transform (FFT), described in Section 4. It is necessary to choose $N > l, m$, since exponentiating the roots of unity can yield at most N distinct values

$$e^{-2\pi i m v/N} = e^{-2\pi i (mv \bmod N)/N};$$

this is related to the Shannon-Nyquist criterion [Shannon, 2001], which dictates that the number of samples required to recover a signal must match its highest frequency. Thus, calculating "high frequency" coefficients— when l, m take large values—requires $\mathcal{O}(N^2)$ numerical ODE solutions, which becomes computationally expensive for large N.

Sparsity: Given an initial state $\mathbf{X}(0) = (j, k)$, the support of transition probabilities is often concentrated over a small range of (l, m) values. For example, if $\mathbf{X}(t) = (800, 800)$, then the probability that the entire process becomes extinct, $\mathbf{X}(t + s) = (0, 0)$, is effectively zero unless particle death rates are very high or s is a very long time interval. In many realistic applications, $p_{(800,800),(l,m)}(s)$ has non-negligible mass on a small support, for instance only over l, m values between 770 and 820. While their values can be computed using Equation (6) for a choice of $N > 820$, requiring N^2 ODE evaluations toward computing only $(820 - 770)^2$ nonzero probabilities seems wasteful. To exploit the sparsity information in such a setting, we bridge aforementioned branching process techniques to the literature of *compressed sensing.*

3 Compressed Sensing

Originally developed in an information theoretic setting, the principle of compressed sensing (CS) states that an unknown sparse signal can be recovered accurately and often perfectly from significantly fewer samples than dictated by the Shannon-Nyquist rate at the cost of solving a convex optimization problem [Donoho, 2006, Candès, 2006]. CS is a robust tool to collect high-dimensional sparse data from a low-dimensional set of measurements and has been applied to a plethora of fields, leading to dramatic reductions in the necessary number of measurements, samples, or computations. In our setting, the transition probabilities play the role of a target sparse signal of Fourier coefficients. The data reduction made possible via CS then translates to reducing computations to a random subsample of PGF evaluations, which play the role of measurements used to recover the signal.

3.1 Overview

In the CS framework, the unknown signal is a vector $\mathbf{x} \in \mathbb{C}^N$ observed through a measurement $\mathbf{b} = \mathbf{V}\mathbf{x} \in \mathbb{C}^M$ with $M << N$. Here $\mathbf{V}$ denotes an $M \times N$ *measurement matrix* or sensing matrix. Since $M < N$, the system is underdetermined and inversion is highly ill-posed—the space of solutions is an infinite affine subspace, but CS theory shows that recovery can be accomplished under certain assumptions by seeking the *sparsest* solution. Let $\boldsymbol{\psi}$ be an orthonormal basis of $\mathbb{C}^N$ that allows a K-sparse representation of $\mathbf{x}$: that is, $\mathbf{x} = \boldsymbol{\psi}\mathbf{s}$ where $\mathbf{s}$ is a sparse vector of coefficients such that $||\mathbf{s}||_0 < K$. Candès [2006] proves that recovery can then be accurately accomplished by finding the sparsest solution

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\text{argmin}}\, ||\mathbf{s}||_0 \quad s.t. \quad \mathbf{A}\mathbf{s} = \mathbf{b} \tag{7}$$

where $\mathbf{A} = \mathbf{V}\boldsymbol{\psi}$ is the composition of the measurement and sparsifying matrices. In practice, this non-convex objective is combinatorially intractable to solve exactly, and is instead solved by proxy via ℓ_1-relaxation, resulting in a convex optimization program. In place of Equation (7), we optimize the unconstrained penalized objective

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\text{argmin}}\, \frac{1}{2}||\mathbf{A}\mathbf{s} - \mathbf{b}||_2^2 + \lambda||\mathbf{s}||_1 \tag{8}$$

where λ is a regularization parameter enforcing sparsity of $\mathbf{s}$. The signal $\mathbf{x}$, or equivalently $\mathbf{s}$, can be recovered perfectly using only $M = CK \log N$ measurements for some constant C when $\mathbf{A}$ satisfies the *Restricted Isometry Property* (RIP) [Candès and Tao, 2005, Candès, 2008]—briefly, this requires that $\mathbf{V}$ and $\boldsymbol{\psi}$ to be *incoherent* so that rows of $\mathbf{V}$ cannot sparsely represent the columns of $\boldsymbol{\psi}$ and vice versa. Coherence between $\mathbf{V}, \boldsymbol{\psi}$ is defined as

$$\mu(\mathbf{V}, \boldsymbol{\psi}) = \sqrt{n} \max_{i,j} |\langle \mathbf{V}, \boldsymbol{\psi}_j \rangle|,$$

and low coherence pairs are desirable. It has been shown that choosing random measurements $\mathbf{V}$ satisfies RIP with overwhelming probability [Candès, 2008]. Further, given $\boldsymbol{\psi}$, it is often possible to choose a known ideal distribution from which to sample elements in $\mathbf{V}$ such that $\mathbf{V}$ and $\boldsymbol{\psi}$ are maximally incoherent.

3.2 Higher dimensions

CS theory extends naturally to higher-dimensional signals [Candès, 2006]. In the 2D case which will arise in our applications (Section 5), the sparse solution $\mathbf{S} \in \mathbb{C}^{N \times N}$ and measurement

$$\mathbf{B} = \mathbf{A}\mathbf{S}\mathbf{A}^T \in \mathbb{C}^{M \times M} \tag{9}$$

are matrices rather than vectors, and we solve

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\text{argmin}}\, \frac{1}{2}||\mathbf{A}\mathbf{S}\mathbf{A}^T - \mathbf{B}||_2^2 + \lambda||\mathbf{S}||_1. \tag{10}$$

This can always be equivalently represented in the vector-valued framework: vectorizing

$$\mathrm{vec}(\mathbf{S}) = \widetilde{\mathbf{s}} \in \mathbb{C}^{N^2}, \quad \mathrm{vec}(\mathbf{B}) = \widetilde{\mathbf{b}} \in \mathbb{C}^{M^2},$$

we now seek $\widetilde{\mathbf{b}} = \widetilde{A}\widetilde{\mathbf{s}}$ as in Equations (7), (8), where $\widetilde{\mathbf{A}} = \mathbf{A} \otimes \mathbf{A}$ is the Kronecker product of $\mathbf{A}$ with itself. In practice, it can be preferable to solve (10), since the number of entries in $\widetilde{\mathbf{A}}$ grows rapidly and thus the vectorized problem requires a costly construction of $\widetilde{\mathbf{A}}$ and can be cumbersome in terms of memory.

4 CSGF Method

We propose an algorithm that allows for efficient PGF inversion within a compressed sensing framework. We focus our exposition on two-type models: linear complexity in $|\Omega|$ is less often a bottleneck in single-type problems, and all generating function methods as well as compressed sensing techniques we describe extend to higher dimensional settings.

We wish to compute the transition probabilities $p_{jk,lm}(t)$ given any $t > 0$ and $\mathbf{X}(0) = (j, k)$. These probabilities can be arranged in a matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ with entries

$$\{\mathbf{S}\}_{l,m} = p_{jk,lm}(t).$$

Without the CS framework, these probabilities are obtained following Equation (6) by first computing an equally sized matrix of PGF solutions

$$\widetilde{\mathbf{B}} = \{\ \phi_{jk}(t, e^{2\pi i u/N}, e^{2\pi i v/N})\}_{u,v=0}^{N-1} \in \mathbb{C}^{N \times N}. \tag{11}$$

For large N, obtaining $\widetilde{\mathbf{B}}$ is computationally expensive, and our method seeks to bypass this step. When $\widetilde{\mathbf{B}}$ is computed, transition probabilities are then recovered by taking the fast Fourier transform $\mathbf{S} = \mathrm{fft}(\widetilde{\mathbf{B}})$. To better understand how this fits into the CS framework, we can equivalently write the fast Fourier transform in terms of matrix operations $\mathbf{S} = \mathbf{F}\widetilde{\mathbf{B}}\mathbf{F}^T$, where $\mathbf{F} \in \mathbb{C}^{N \times N}$ denotes the discrete Fourier transform matrix (see Supplement). Thus, the sparsifying basis $\boldsymbol{\psi}$ is the Inverse Discrete Fourier Transform (IDFT) matrix $\boldsymbol{\psi} = \mathbf{F}^*$ given by the conjugate transpose of $\mathbf{F}$, and we have $\widetilde{\mathbf{B}} = \boldsymbol{\psi}\mathbf{S}\boldsymbol{\psi}^T$.

When the solution matrix $\mathbf{S}$ is expected to have a sparse representation, our CSGF method seeks to recover $\mathbf{S}$ without computing the full matrix $\widetilde{\mathbf{B}}$, instead beginning with a much smaller set of PGF evaluations $\mathbf{B} \in \mathbb{C}^{M \times M}$ corresponding to random entries of $\widetilde{\mathbf{B}}$ selected uniformly at random. Denoting randomly sampled indices $\mathcal{I}$, this smaller matrix is a projection $\mathbf{B} = \mathbf{A}\mathbf{S}\mathbf{A}^T$ in the form of Equation (9) where $\mathbf{A} \in \mathbb{C}^{M \times N}$ is obtained by selecting a subset of rows of $\boldsymbol{\psi}$ corresponding to $\mathcal{I}$. Uniform sampling of rows corresponds to multiplying by a measurement matrix encoding the *spike basis* (or standard basis): formally, this fits into the framework described in Section 3.1 as $\mathbf{A} = \mathbf{V}\boldsymbol{\psi}$, with measurement matrix rows $\mathbf{V}_j(l) = \delta(j - l)$. The spike and Fourier bases are known to be *maximally incoherent* in any dimension, so uniformly sampling indices $\mathcal{I}$ is optimal in our setting.

Now in the compressed sensing framework, computing the reduced matrix $\mathbf{B}$ only requires a logarithmic proportion $|\mathbf{B}| \propto K \log |\widetilde{\mathbf{B}}|$ of PGF evaluations necessary in Equation (11). Computing transition probabilities in $\mathbf{S}$ is thus reduced to a signal recovery problem, solved by optimizing the objective in Equation (10).

4.1 Solving the ℓ_1 problem

There has been extensive research on algorithms for solving the ℓ_1 regularization objective in Equation (8) and related problems [Tibshirani, 1996, Beck and Teboulle, 2009a]. As mentioned previously, vectorizing the problem so that it can be represented in the form (8) requires wasteful extra memory; instead we choose to solve the objective in Equation (10) using a *proximal gradient descent* (PGD) algorithm.

PGD is useful for solving minimization problems with objective of the form $f(x) = g(x) + h(x)$ with g convex and differentiable, and h convex but not necessarily differentiable. Letting

$$g(\mathbf{S}) = \frac{1}{2}||\mathbf{ASA}^T - \mathbf{B}||_2^2, \quad h(\mathbf{S}) = \lambda||\mathbf{S}||_1,$$

we see that Equation (10) satisfies these conditions. A form of generalized gradient descent, PGD iterates toward a solution with

$$\begin{aligned} x_{k+1} = \underset{z}{\operatorname{argmin}}[g(x_k) + \nabla g(x_k)^T(z - x_k) \\ + \frac{1}{2L_k}||z - x_k||_2^2 + h(z)], \end{aligned} \tag{12}$$

where L_k is a step size that is either fixed or determined via line-search. This minimization has known closed-form solution

$$x_{k+1} = \text{softh}(x_k - L_k \nabla g(x_k), L_k\lambda), \tag{13}$$

where softh is the soft-thresholding operator

$$[\text{softh}(x, \alpha)]_i = \text{sgn}(x_i) \max(|x_i| - \alpha, 0). \tag{14}$$

Alternating between these steps results in an *iterative soft-thresholding algorithm* that solves the convex problem (10) with rate of convergence $\mathcal{O}(1/k)$ when L_k is fixed. The softh() operation is simple and computationally negligible, so that the main computational cost is in evaluating $\nabla g(x_k)$. We derive a closed form expression for the gradient in our setting

$$\nabla g(\mathbf{S}) = -\mathbf{A}^*(\mathbf{B} - \mathbf{ASA}^T)\overline{\mathbf{A}}, \tag{15}$$

where $\overline{\mathbf{A}}, \mathbf{A}^*$ denote complex conjugate and conjugate transpose of $\mathbf{A}$ respectively. In practice, the inner term $\mathbf{ASA}^T$ is obtained as a subset of the inverse fast Fourier transform of $\mathbf{S}$ rather than by explicit matrix multiplication. The computational effort in computing $\nabla g(\mathbf{S})$ therefore involves only the two outer matrix multiplications.

We implement a fast variant of PGD using momentum terms [Beck and Teboulle, 2009b] based on an algorithm introduced by Nesterov, and select step sizes L_k via a simple line-search subroutine [Beck and Teboulle, 2009a]. The accelerated version includes an *extrapolation step*, where the soft-thresholding operator is applied to a momentum term

$$y_{k+1} = x_k + \omega_k(x_k - x_{k-1})$$

rather than to x_k; here ω_k is an extrapolation parameter for the momentum term. Remarkably, the accelerated method still only requires one gradient evaluation at each step as y_{k+1} is a simple linear combination of previously computed points, and has been proven to achieve the optimal worst-case rate of convergence $\mathcal{O}(1/k^2)$ among first order methods [Nesterov, 1983]. Similarly, the line-search procedure involves evaluating a bound that also only requires one evaluation of ∇g (see Supplement).

Algorithm 1 provides a summary of the CSGF method in pseudocode.

Algorithm 1 `CSGF` algorithm.

```
**Input:** initial sizes $X_1 = j, X_2 = k$, time interval $t$, branching rates $\boldsymbol{\theta}$, signal size $N > j, k$, measurement size $M$, penalization constant $\lambda > 0$, line-search parameters $L, c$.
Uniformly sample $M$ indices $\mathcal{I} \subset [0, \dots N-1]/N$
Compute $\mathbf{B} = \{\ \phi_{jk}(t, e^{2\pi i u/N}, e^{2\pi i v/N})\}_{u,v \in \mathcal{I} \times \mathcal{I}}$
Define $\mathbf{A} = \boldsymbol{\psi}_{\mathcal{I}\cdot}$ the $\mathcal{I}$ rows of IDFT matrix $\boldsymbol{\psi}$
**Initialize:** $\mathbf{S}_1 = \mathbf{Y}_1 = \mathbf{0}$
**for** $k = 1, 2, \dots, \{\text{max iterations}\}$ **do**
    Choose $L_k = \texttt{line-search}(L, c, \mathbf{Y}_k)$
    Update extrapolation parameter $\omega_k = \frac{k}{k+3}$
    Update momentum $\mathbf{Y}_{k+1} = \mathbf{S}_k + \omega_k(\mathbf{S}_k - \mathbf{S}_{k-1})$
    Compute $\nabla g(\mathbf{Y}_{k+1})$ according to (15)
    Update $\mathbf{S}_{k+1} = \text{softh}(\mathbf{S}_k - L_k \nabla g(\mathbf{Y}_{k+1}), L_k \lambda)$
**end for**
**return** $\hat{\mathbf{S}} = \mathbf{S}_{k+1}$
```

5 Examples

We will examine the performance of CSGF in two applications: a stochastic two-compartment model used in statistical studies of *hematopoiesis*, the process of blood cell production, and a birth-death-shift model that has been used to study the evolution of *transposons*, mobile genetic elements.

5.1 Two-compartment hematopoiesis model

Hematopoiesis is the process in which self-sustaining primitive hematopoietic stem cells (HSCs) specialize, or *differentiate*, into progenitor cells, which further specialize to eventually produce mature blood cells. In addition to far-reaching clinical implications — stem cell transplantation is a mainstay of cancer therapy — understanding hematopoietic dynamics is biologically interesting, and provides critical insights of general relevance to other areas of stem cell biology [Orkin and Zon, 2008]. The stochastic model, depicted in Figure 1, has enabled estimation of hematopoietic rates in mammals from data in several studies [Catlin et al., 2001, Golinelli et al., 2006, Fong et al., 2009]. Without the ability to compute transition probabilities, an estimating equation approach by Catlin et al. [2001] is statistically inefficient, resulting in uncertain estimated parameters with very wide confidence intervals. Nonetheless, biologically sensible rates are inferred. Golinelli et al. [2006] observe that transition probabilities are unknown for a linear birth-death process (compartment 1) coupled with an inhomogeneous immigration-death process (compartment 2), motivating their computationally intensive reversible jump MCMC implementation.

However, we can equivalently view the model as a two-type branching process. Under such a representation, it becomes possible to compute transition probabilities via Equation (6). The type one particle population X_1 corresponds to hematopoietic stem cells (HSCs), and X_2 represents progenitor cells. With parameters as denoted in Figure 1, the nonzero instantaneous rates defining the process are

$$\begin{aligned} &a_1(2,0) = \rho \qquad && a_1(0,1) = \nu \qquad && a_1(1,0) = -(\rho + \nu) \\ &a_2(0,0) = \mu \qquad && a_2(0,1) = -\mu. && \end{aligned} \tag{16}$$

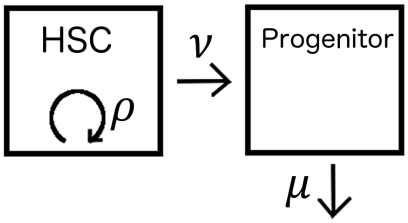

Figure 1: HSCs can self-renew, producing new HSCs at rate ρ, or differentiate into progenitor cells at rate ν. Further progenitor differentiation is modeled by rate μ.

Having specified the two-type branching process, we derive solutions for its PGF, defined in Equation (4), with details in the Supplement:

Proposition 5.1 *The generating function for the two-type model described in* (16) *is given by* $\phi_{jk} = \phi_{1,0}^{j}\phi_{0,1}^{k}$*, where*

$$\begin{cases} \phi_{0,1}(t,s_1,s_2) = 1 + (s_2 - 1)e^{-\mu t} \\ \frac{d}{dt}\phi_{1,0}(t,s_1,s_2) = \rho\phi_{1,0}^2(t,s_1,s_2) - (\rho+\nu)\phi_{1,0}(t,s_1,s_2) \\ \qquad\qquad +\nu\phi_{0,1}(t,s_1,s_2). \end{cases} \tag{17}$$

We see that $\phi_{0,1}$ has closed form solution so that evaluating ϕ_{jk} only requires solving one ODE numerically, and with the ability to compute ϕ_{jk}, we may obtain transition probabilities using Equation (6). In this application, cell populations can easily reach thousands, motivating the CSGF approach to accelerate transition probability computations.

5.2 Birth-death-shift model for transposons

Our second application examines the birth-death-shift (BDS) process proposed by Rosenberg et al. [2003] to model evolutionary dynamics of transposable elements or *transposons*, genomic mobile sequence elements. Each transposon can (1) duplicate, with the new copy moving to a new genomic location; (2) shift to a different location; or (3) be removed and lost from the genome, independently of all other transposons. These respective birth, shift, and death events occur at per-particle instantaneous rates β, σ, δ, with overall rates proportional to the total number of transposons. Transposons thus evolve according to a linear *birth-death-shift* Markov process in continuous time. In practice, genotyping technologies allow for this process to be discretely monitored, necessitating computation of finite-time transition probabilities.

Rosenberg et al. [2003] estimate evolutionary rates of the IS*6110* transposon in the *Mycobacterium tuberculosis* genome from a San Francisco community study dataset [Cattamanchi et al., 2006]. Without transition probabilities, the authors maximize an approximate likelihood by assuming at most one event occurs per observation interval, a rigid assumption that severely limits the range of applications. Doss et al. [2013] revisit their application, inferring similar rates of IS*6110* evolution using a one-dimensional birth-death model that ignores shift events. Xu et al. [2014] show that the BDS model over any finite observation interval can be modeled as a two-type branching process, where X_1 denotes the number of initially occupied genomic locations and X_2 denotes the number of newly occupied locations (see figure in Supplement). In this representation, full dynamics of the BDS model can be captured, and generating function techniques admit transition probabilities, leading to rate estimation via MLE and EM algorithms. Transposon counts in the tuberculosis dataset are low, so that Equation (6) can be computed easily, but their method does not scale well to applications with high counts in the data.

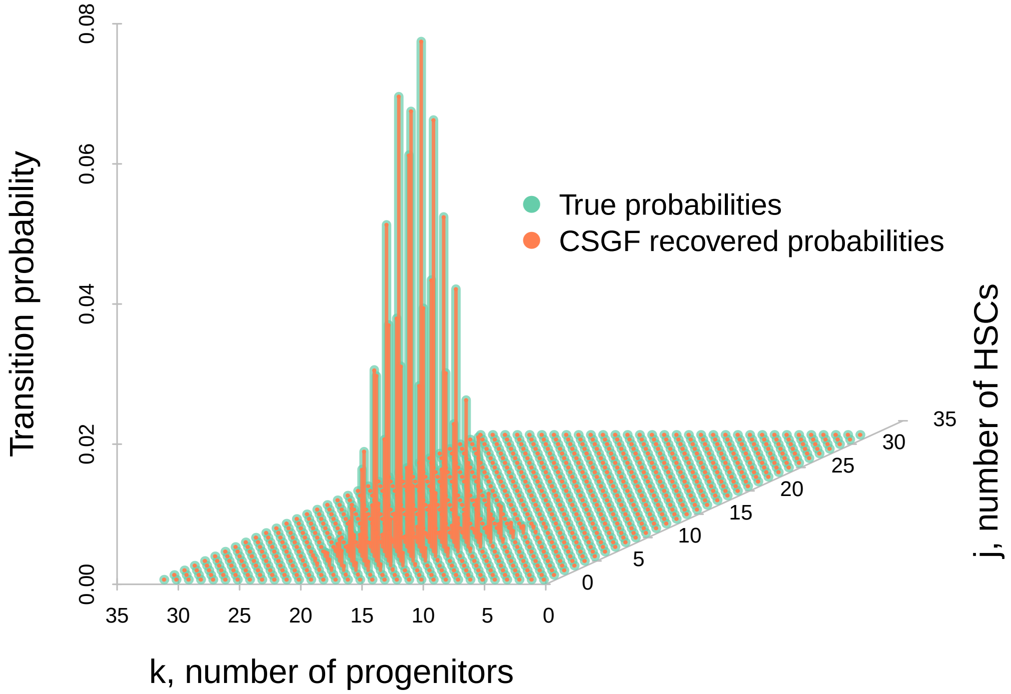

Figure 2: Illustrative example of recovered transition probabilities in hematopoiesis model described in Section 5. Beginning with 15 HSCs and 5 progenitors over a time period of one week, the CSGF solution $\hat{\mathbf{S}} = \left\{\hat{p}_{(15,5),(j,k)}(1)\right\}$, $j, k = 0, \ldots, 31$, perfectly recovers transition probabilities $\mathbf{S}$, using fewer than half the measurements.

The nonzero rates defining the two-type branching process representation of the BDS model are given by

$$\begin{aligned} &a_1(1,1) = \beta, && a_1(0,1) = \sigma, && a_1(0,0) = \delta, \\ &a_1(1,0) = -(\beta+\sigma+\delta), && a_2(0,2) = \beta, && \\ &a_2(0,1) = -(\beta+\delta), && a_2(0,0) = \delta. && \end{aligned} \tag{18}$$

and its PGF is governed by the following system derived in [Xu et al., 2014]:

$$\begin{cases} \phi_{0,1}(t,s_1,s_2) = 1 + \left[\frac{\beta}{\delta-\beta} + (\frac{1}{s_2-1} + \frac{\beta}{\beta-\delta})e^{(\delta-\beta)t}\right]^{-1} \\ \frac{d}{dt}\phi_{1,0}(t,s_1,s_2) = \beta\phi_{1,0}\phi_2 + \sigma\phi_{0,1} + \delta - (\beta+\sigma+\delta)s_1, \end{cases} \tag{19}$$

again with $\phi_{jk} = \phi_{1,0}^{j}\phi_{0,1}^{k}$ by particle independence.

5.3 Results

To compare the performance of CSGF to the computation of Equation (6) without considering sparsity, we first compute sets of transition probabilities $\mathbf{S}$ of the hematopoiesis model using the full set of PGF solution measurements $\widetilde{\mathbf{B}}$ as described in Equation (11). These "true signals" are compared to the signals computed using CSGF $\hat{\mathbf{S}}$, recovered using only a random subset of measurements $\mathbf{B}$ following Algorithm 1. Figure 2 provides an illustrative example with small cell populations for visual clarity— we see that the support of transition probabilities is concentrated (sparse), and the set of recovered probabilities $\hat{\mathbf{S}}$ is visually identical to the true signal.

In each of the aforementioned applications, we calculate transition probabilities $\mathbf{S} \in \mathbb{R}^{N \times N}$ for maximum populations $N = 2^7, 2^8, \dots 2^{12}$, given rate parameters $\boldsymbol{\theta}$, initial population $\mathbf{X}(0)$, and time intervals t. Each computation of $\mathbf{S}$ requires N^2 numerical evaluations of the ODE systems (17), (19). For each value of N, we repeat this procedure beginning with ten randomly chosen sets of initial populations $\mathbf{X}(0)$ each with total size less than N. We compare the recovered signals $\hat{\mathbf{S}}$ computed using CSGF to true signals $\mathbf{S}$, and report median runtimes and measures of accuracy over the ten trials, with details in the following sections.

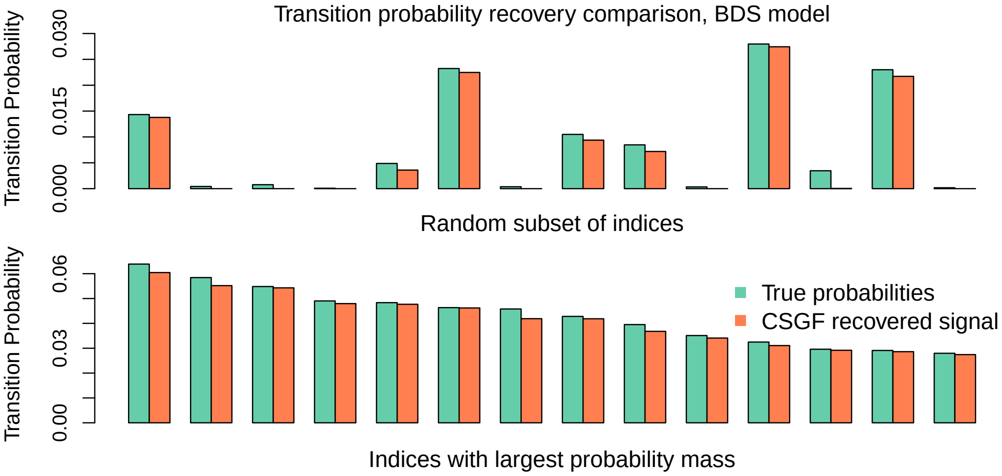

Figure 3: Randomly selected probabilities and largest probabilities recovered using CSGF are nearly identical to their true values. Probabilities displayed here correspond to a randomly selected BDS model trial with N=512; transition probabilities $\hat{\mathbf{S}}$ via CSGF are recovered from a sample $\mathbf{B}$ requiring fewer than 2% of ODE computations used to compute $\mathbf{S} = \text{fft}(\widetilde{\mathbf{B}})$.

Parameter settings: In the hematopoiesis example, we set per-week rates $\boldsymbol{\theta}_{\text{hema}} = (0.125, 0.104, 0.147)$ and observation time $t = 1$ week based on biologically sensible rates and observation time scales of data from previous studies of hematopoiesis in mammals [Catlin et al., 2001, Golinelli et al., 2006, Fong et al., 2009]. For the BDS application, we set per-year event rates $\boldsymbol{\theta}_{\text{bds}} = (0.0156, 0.00426, 0.0187)$ estimated in [Xu et al., 2014], and $t = .35$ years, the average length between observations in the San Francisco tuberculosis dataset [Cattamanchi et al., 2006].

In each case, we computed $M^2 = 3K \log N^2$ total random measurements to obtain $\mathbf{B}$ for CSGF, and we set the regularization parameters $\lambda_{\text{hsc}} = \sqrt{\log M}$, $\lambda_{\text{bds}} = \log M$, with more regularization in the BDS application as lower rates and a shorter observation interval leads us to expect more sparsity. While careful case-by-case tuning to choose λ, M would lead to optimal results, we set them in this simple manner across *all* trials to demonstrate a degree of robustness, still yielding promising performance results. In practice one may apply standard cross-validation procedures to select λ, M, and because the target solution is a set of transition probabilities, checking that entries in the recovered solution $\hat{\mathbf{S}}$ sum close to 1 offers a simpler available heuristic. Finally, though one may expedite convergence of PGD by supplying an informed initial guess with positive values near values $\mathbf{X}(0)$ in practice, we initialize PGD with an uninformative initial value $\mathbf{S}_1 = \mathbf{0}$ in all cases.

Table 1: Runtimes and error, birth-death-shift model.

		Time (sec),	Time (sec),	Time (sec),	$\varepsilon_{\max} =$	$\varepsilon_{\text{rel}} =$
N	M	$\widetilde{\mathbf{B}} \in \mathbb{C}^{N\times N}$	$\mathbf{B} \in \mathbb{C}^{M\times M}$	PGD	$\lvert\hat{p}_{ij,kl} - p_{ij,kl}\rvert_{\max}$	$\varepsilon_{\max}/\lvert p_{ij,kl}\rvert_{\max}$
128	25	39.7	2.3	1.0	5.27×10^{-3}	2.77×10^{-2}
256	33	150.2	3.8	7.8	4.86×10^{-3}	4.71×10^{-2}
512	45	895.8	7.8	25.3	2.71×10^{-3}	4.68×10^{-2}
1024	68	2508.9	18.6	58.2	1.41×10^{-3}	5.12×10^{-2}
2048	101	9788.3	26.1	528.3	8.10×10^{-4}	4.81×10^{-2}
4096	150	40732.7	57.4	2234.7	4.01×10^{-4}	5.32×10^{-2}

Table 2: Runtimes and error, hematopoiesis model

		Time (sec),	Time (sec),	Time (sec),	$\varepsilon_{\max} =$	$\varepsilon_{\text{rel}} =$
N	M	$\widetilde{\mathbf{B}} \in \mathbb{C}^{N\times N}$	$\mathbf{B} \in \mathbb{C}^{M\times M}$	PGD	$\lvert\hat{p}_{ij,kl} - p_{ij,kl}\rvert_{\max}$	$\varepsilon_{\max}/\lvert p_{ij,kl}\rvert_{\max}$
128	43	108.6	9.3	0.64	9.41×10^{-4}	2.25×10^{-2}
256	65	368.9	22.1	2.1	9.44×10^{-4}	4.73×10^{-2}
512	99	922.1	44.8	8.5	3.23×10^{-4}	3.60×10^{-2}
1024	147	5740.1	118.1	41.9	2.27×10^{-4}	5.01×10^{-2}
2048	217	12754.8	145.0	390.0	1.29×10^{-4}	5.10×10^{-2}
4096	322	58797.3	310.7	2920.3	9.43×10^{-5}	6.13×10^{-2}

Accuracy: In both models and for all values of N, each signal was reconstructed very accurately. Errors are reported in Tables 1 and 2 for the BDS and hematopoiesis models respectively. Maximum absolute errors for each CSGF recovery

$$\varepsilon_{\max} = \max_{kl} \lvert\{\hat{\mathbf{S}}\}_{kl} - \{\mathbf{S}\}_{kl}\rvert = \max_{kl} \lvert\hat{p}_{ij,kl}(t) - p_{ij,kl}(t)\rvert$$

are on the order of 10^{-3} at worst. We also report a measure of relative error, and because $\varepsilon_{\max}$ is typically attained at large probabilities, we include the maximum absolute error relative to the largest transition probability

$$\varepsilon_{\text{rel}} = \frac{\varepsilon_{\max}}{\max_{kl} \{S\}_{kl}},$$

providing a more conservative measure of accuracy. We still see that ε_{rel} is on the order of 10^{-2} in all cases. Visually, the accuracy of CSGF is stark: Figure 3 provides a side-by-side comparison of randomly selected transition probabilities recovered in the BDS model for $N = 2^9$.

Running Times: Tables 1 and 2 show dramatic improvements in runtime using CSGF, reducing the number of ODE computations logarithmically. For instance, with $N = 4096$, we see the time spent on PGF evaluations necessary for CSGF is less than 0.1% of the time required to compute $\mathbf{S}$ in the BDS model, and around 0.5% of computational cost in the less sparse hematopoiesis application. Including the time required for solving Equation (10) via PGD, we see that computing $\hat{\mathbf{S}}$ using CSGF reduces runtime by two orders of magnitude, requiring less than 6% of total computational time spent toward computing $\mathbf{S}$ in the worst case. We remark that ODE solutions are computed using a C implementation of efficient solvers via package `deSolve`, while we employ a naive R implementation of PGD. We emphasize the logarithmic reduction in required numerical ODE solutions; an optimized implementation of PGD reducing R overhead will yield further real-time efficiency gains.

6 Discussion

We have presented a novel adaptation of recent generating function techniques to compute branching process transition probabilities within the compressed sensing paradigm. While generating function approaches bypass costly matrix exponentiation and simulation-based techniques by exploiting mathematical properties in the branching structure, our contribution now makes these techniques scalable by additionally harnessing the available sparsity structure. We show that when sparsity is present in the set of transition probabilities, computational cost can be reduced up to a logarithmic factor over existing methods. Note that sparsity is the *only* additional assumption necessary to apply our CSGF method—no prior knowledge about where transition probabilities have support is necessary. Many real-world applications of branching process modeling feature such sparsity, and we have seen that CSGF achieves accurate results with significant efficiency gains in two such examples with realistic parameter settings from the scientific literature. Transition probabilities are often important, interpretable quantities in their own right, and are necessary within any likelihood-based probabilistic framework for partially observed CTMCs. Their tractability using CSGF opens doors to applying many Bayesian and frequentist tools to settings in which such methods were previously infeasible. Finally, we note that other statistically relevant quantities such as expectations of particle dwell times and restricted moments can be computed using similar generating function techniques [Minin and Suchard, 2008], and the CSGF framework applies analogously when sparsity is present.

References

C Andrieu, A Doucet, and R Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

NTJ Bailey. *The Elements of Stochastic Processes; with Applications to the Natural Sciences.* New York: Wiley, 1964.

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

A Beck and M Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009b.

JC Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods.* Wiley-Interscience, 1987.

EJ Candès. Compressive sampling. In *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pages 1433–1452, 2006.

EJ Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

EJ Candès and T Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

SN Catlin, JL Abkowitz, and P Guttorp. Statistical inference in a two-compartment model for hematopoiesis. *Biometrics*, 57(2):546–553, 2001.

A Cattamanchi, PC Hopewell, LC Gonzalez, DH Osmond, L Masae Kawamura, CL Daley, and RM Jasmer. A 13-year molecular epidemiological analysis of tuberculosis in San Francisco. *The International Journal of Tuberculosis and Lung Disease*, 10(3):297–304, 2006.

FW Crawford, VN Minin, and MA Suchard. Estimation for general birth-death processes. *Journal of the American Statistical Association*, 109(506):730–747, 2014.

DL Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

CR Doss, Ma Suchard, I Holmes, MM Kato-Maeda, and VN Minin. Fitting birth–death processes to panel data with applications to bacterial DNA fingerprinting. *The Annals of Applied Statistics*, 7(4):2315–2335, 2013.

A Doucet, S Godsill, and C Andrieu. On sequential Monte carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

T El-Hay, N Friedman, D Koller, and R Kupferman. Continuous time Markov networks. In *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI)*, Boston, Massachussetts, July 2006.

Y Fong, P Guttorp, and J Abkowitz. Bayesian inference and model choice in a hidden stochastic two-compartment model of hematopoietic stem cell fate decisions. *The Annals of Applied Statistics*, 3(4):1695–1709, 12 2009.

D Golinelli, P Guttorp, and JA Abkowitz. Bayesian inference in a hidden stochastic two-compartment model for feline hematopoiesis. *Mathematical Medicine and Biology*, 23(3):153–172, 2006.

WK Grassmann. Transient solutions in Markovian queueing systems. *Computers & Operations Research*, 4(1):47–53, 1977.

P Guttorp. *Stochastic modeling of scientific data.* CRC Press, 1995.

M Hajiaghayi, B Kirkpatrick, L Wang, and A Bouchard-Côté. Efficient continuous-time Markov chain estimation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 638–646, 2014.

K Lange. Calculation of the equilibrium distribution for a deleterious gene by the finite Fourier transform. *Biometrics*, 38(1):79–86, 1982.

VN Minin and MA Suchard. Counting labeled transitions in continuous-time Markov models of evolution. *Journal of Mathematical Biology*, 56(3):391–412, 2008.

Y Nesterov. A method of solving a convex programming problem with convergence rate O(1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

SH Orkin and LI Zon. Hematopoiesis: An evolving paradigm for stem cell biology. *Cell*, 132(4): 631–644, 2008.

VA Rao and YW Teh. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Proceedings of the 27th International Conference on Uncertainty in Artificial Intelligence.* 2011.

E Renshaw. *Stochastic Population Processes: Analysis, Approximations, Simulations.* Oxford University Press Oxford, UK, 2011.

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.

CE Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

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.

Supplement

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 is obtained by only computing and stacking a subset of M random rows from $\boldsymbol{\psi}$.

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:

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

We follow Beck and Teboulle [2009], who provide further details. In implementation, we select $L = .000005$ and $c = .5$, and reuse the gradient computed in `line-search` for step 10 of Algorithm 1 in the main paper.

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 $i = 1, 2$:

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

Algorithm 2 `line-search` procedure.

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

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

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

Of course 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

$$\begin{aligned}
\phi_1(t+h, s_1, s_2) &= \phi_1(t, \phi_1(h, s_1, s_2), \phi_2(h, s_1, s_2)) &\text{(A-2)}\\
&= \phi_1(h, \phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2)) &\text{(A-3)}
\end{aligned}$$

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

$$a_1(2,0) = \rho \qquad a_1(0,1) = \nu \qquad a_1(1,0) = -(\rho + \nu)$$
$$a_2(0,0) = \mu \qquad a_2(0,1) = -\mu$$

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

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

Plugging this solution into the other backward equation, we obtain

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

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.

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.

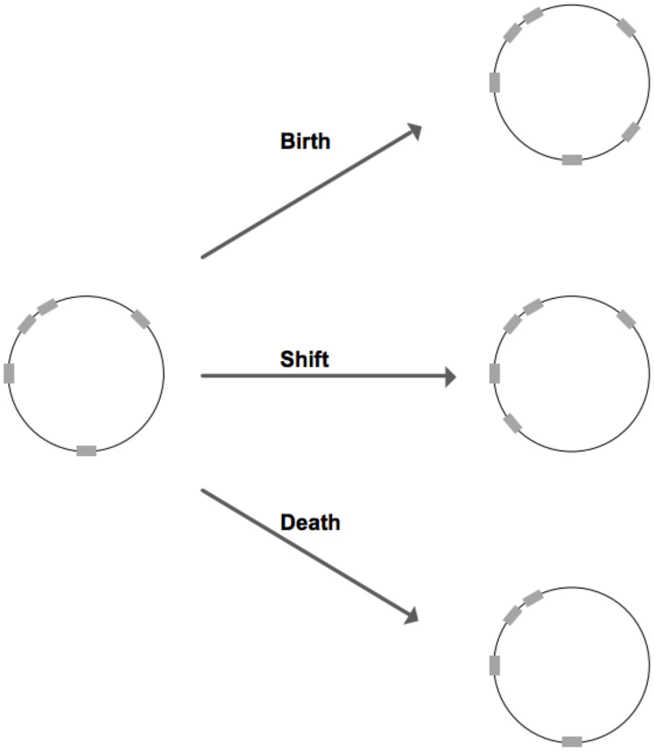

Figure C-4: Illustration of the three types of transposition—birth, death, shift—along a genome, represented by circles [Rosenberg et al., 2003]. Transposons are depicted by rectangles occupying locations along the circles/genomes. On the right set of diagrams, a birth event keeps the number of type 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.