UNIVERSITY OF CALIFORNIA

Los Angeles

Simulation and Numerical Methods for Stochastic Processes

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Biomathematics

by

Timothy Charles Stutz

2020

ABSTRACT OF THE DISSERTATION

Simulation and Numerical Methods for Stochastic Processes

by

Timothy Charles Stutz
Doctor of Philosophy in Biomathematics
University of California, Los Angeles, 2020
Professor Kenneth L Lange, Chair

Stochastic processes and randomness are vital features of mathematical modeling in biology. Unfortunately analytical results are rarely available for even moderately complex stochastic processes leaving simulation and numerical techniques the main avenues of attack. We begin this work by exploring coupling bounds for birth-death processes, a fundamental type of stochastic process that describes how populations of individuals change over time. By forming a coupling between a truncated version of the process and the original unbounded version, we are able to compute both moments and transition probabilities for the true process within an acceptable error bound. Second, we present an algorithm design framework for Interacting Particle Systems (IPSs). These are complex stochastic processes with wide application to spatial phenomenon across many scientific disciplines. Here we describe a method for efficiently sorting particles into classes based off of their type and spatial configuration in such a fashion that reduces the spatial simulation to that of a non-spatial well-mixed process, albeit with a more complicated update step. This also allows us to apply a large suite of well-developed stochastic simulation algorithms to IPSs with little additional coding cost. Third, we return to numerical methods, this time for multi-type branching processes applied to gene therapy. We derive a series of ordinary differential equations that govern the evolution of the probability generating function and provide a straightforward numerical inversion approach to obtain marginalized probability distributions for probabilistic quantities of interest. We provide examples of our techniques applied to lentiviral gene therapy and the associated risk of oncogenesis in transplanted hematopoietic stem cell lines.

Finally, we conclude with a chapter on future directions, both related to the previous three chapters as well as projects not previously addressed in this work.

The dissertation of Timothy Charles Stutz is approved.

Zhilin Qu

Mary Elizabeth Sehl

Janet S Sinsheimer

Hua Zhou

Kenneth L Lange, Committee Chair

University of California, Los Angeles

2020

For Connie.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

My path has not been straightforward. It would not have been possible to reach this point without the guidance of Ken Lange, Mary Sehl, Janet Sinsheimer, and Jason Xu. Their unbounded patience, compassion, and empathy has been just as important as their excellent scholastic insights. I could not have asked for a better collection of mentors.

Similarly I must thank my cohort Lindsay Riley, Song Xu, and Bhaven Mistry. Their comraderie along with the support of the entire Biomathematics department has provided invaluable solidarity. In particular I must single out both Alfonso Landeros and Bhaven Mistry for being exceptional colleagues and friends.

Finally I must thank my parents Tom and Karen for their love and faith in me. I could not have asked for a better brother than Kenton. And of course there is Connie, with all my love.

VITA

2012 B.A. (Biology and Applied Mathematics), Pomona College.

2014 M.S. (Biomathematics), University of California, Los Angeles.

PUBLICATIONS

Morgens, DW, Stutz, TC, Cavalcanti, AR. "Novel population genetics in ciliates due to life cycle and nuclear dimorphism." *Molecular biology and evolution* 31, 2084-2093, 2014.

Crawford, FW, Stutz, TC, Lange KL. "Coupling bounds for approximating birth-death processes by truncation." *Statistics & Probability Letters* 109, 30-38, 2016.

Landeros, A, Stutz, TC, Alekseyenko, A, Keys, KL, Lange, KL, Sehl, ME. "BioSimulator: A Julia Package for Stochastic Simulation of Complex Biological Systems." *Computer Methods and Programs in Biomedicine* 167, 23-35, 2018.

Stutz, TC, Landeros, A, Xu, J, Sinsheimer, JS, Sehl, ME, Lange, KL. "Stochastic simulation algorithms for interacting particle systems." Manuscript in review in *PLOS Compuational Biology.*

Stutz, TC, Sehl, ME, Xu, J (2020) "Numerical techniques for computing stem cell hierarchies via multi-type branching processes with applications to gene therapy.", Manuscript in preparation.

CHAPTER 1

Introduction

1.1 Stochastic Processes in Biology

No mathematical approach to biology is complete without random processes. Noise is present in species fluctuations, switching in genetic regulation, the arrival of mutations in evolutionary processes, and countless other phenomenon. The cornerstone of correctly modeling these diverse random mechanisms lies in the mathematical discipline of stochastic processes.

Population processes are frequently described by systems of ordinary differential equations that govern how concentrations of different species interact and change over time. These models assume that there are enough particles to be accurately described by a continuous concentration and that the dynamics of the system are purely deterministic. Both of these assumptions fail when the system has random noise and when modeling individual particles with low counts become important. Classic examples include predator-prey dynamics, disease outbreak, the recruitment of driver mutations in tumorigenesis, population extinction from demographic noise, and other rare events. Differential equations fail to accurately model these phenomena, so we are required to introduce stochasticity.

Unfortunately, stochastic models are frequently too complicated to be treated analytically outside of well-mixed systems with few particle types. Important quantities such as finite time transition probabilities, moments and correlation functions, and statistics about first passage times are often intractable. Numerical methods and simulation offer solutions where none would otherwise be available.

This work focuses on both numerical and simulation based techniques for specific classes of stochastic processes. The first chapter focuses on birth-death processes (BDPs), a well-

behaved class of single species stochastic models where a population of individuals changed by at most one count per event. Classic examples of BDPs include the Poisson process, Kendall's birth-death-immigration process, the Moran process, and the logistic process. These form fundamental models for disease dynamics, tumor progression, phylogenetics, and queuing theory [NKK06].

While robust numerical techniques exist for numerically computing finite time transition probabilities for arbitrary BDPs [CS12], less work has been done on computing their moments. Chapter 2 presents a novel coupling argument that bounds the total variation distance between a non-explosive BDP and a truncated version based on the truncation index N [CSL16]. Once a given truncation index is found, there exist efficient numerical techniques for computing moments and transition probabilities from the simplified Markov chain. From this, modelers can quickly test different combinations of birth and death rates to see if a model produces the desired average behavior. This application provides a powerful example of how coupling arguments can greatly simplify computations for stochastic processes.

The third chapter switches focus from well-mixed processes to simulation approaches for spatial particle processes which are prohibitavely resistant to numerical approaches. Spatial variation is vital for accurately modeling many evolutionary processes, such as tumor growth, dessertification, disease spreading, and maintenance of species biodiversity. Simulation remains the primary approach for observing novel behavior in spatial models that cannot be observed in their well-mixed counterparts.

We present several simulation algorithms for spatial interacting particle systems (IPSs). IPSs are a class of stochastic models with full spatial detail, tracking each particle's location on a lattice [Lig12]. Interactions are assumed to be completely local, meaning particles must be adjacent to each other to react. This limits IPSs to modeling at most bimolecular reactions, but this is not a significant limitation; higher order reactions can be modeled as a series of bimolecular reactions, each forming a new complex that takes part in the next reaction. Importantly, IPSs preserve volume exclusion, meaning at most one particle can be present on any given lattice site. Diffusive movement is typically modeled via particles obeying random walks between sites, respecting exclusion. This is in constrast with more

common reaction diffusion master equation (RDME) approaches that couple compartments obeying well-mixed dynamics.

We provide software, written in the open source language `Julia`, that implements different algorithms for simulating IPSs. These include the exact stochastic simulation algorith (SSA), a version of the celebrated τ-leaping algorithm, and a novel extension of the SSA that is ideal for importance sampling and addressing first passage problems. Our software provides a simple, intuitive interface through which nonspecialists can quickly observe the behavior of spatial models with multiple interacting species and other complex behaviors. This allows for straightforward checking for the appropriate species and interactions in a proposed model. From this, modelers can determine which parameters are important for producing a certain desired behavior. A recent example comes from using immune therapy to treat cancer: a complex model of tumor-immune system interactions was analyzed to determine which two parameters generate the appropriate hot or cold immune responses [KPS17].

The third chapter returns to numerical methods for computing probabilistic quantities, this time for multi-type branching processes. Here we present a method for computing the probability generating function (PGF) of the process and describe how to invert the PGF to obtain marginalized probability distributions. This has direct application to lentiviral gene addition through stem cell transduction, specifically to quantifying the risk of leukemogenesis after gene therapy. We present numerical examples of our techniques applied to gene therapy for hematopoietic stem cells and discuss clinically relevant probabilistic quantities related to the accumulation of mutations that lead to oncogenesis.

Finally, the fourth chapter focuses on future directions for the three previous projects. We also include potential future projects that go beyond the extensions of the previous chapters, particularly with regards to computing the mean time to extinction in a stochastic model of tumor quiescence.

Chapter two has been published in Statistics and Probability Letters. Chapter three has been submitted for publication to PLOS Computational Biology, and chapter four is in

preparation for submission.

CHAPTER 2

Coupling bounds for approximating birth-death processes by truncation

2.1 Background

A birth-death process (BDP) is a single species stochastic process $X(t)$ that counts the number of individuals in a system at time $t > 0$. The process evolves over the non-negative integers $\mathbb{N}$ according to two state-specific rates, λ_n and μ_n. $\lambda_n dt$ is the probability over a very short time interval dt that the system transitions from state n to $n+1$ through birth. μ_n likewise gives the rate of death from state n to $n-1$. 21 BDPs have a well developed mathematical history. Linear BDPs can be fully solved using generating functions and partial differential equations [Fel08], while asymptotic methods exist for characterizing processes with strong metastable behavior [AM17]. Even if a process is too complex for either of these approaches, a robust numerical technique exists for computing the finite time transition probabilities using a truncated continued fractions representation of the Laplace transform [CS12].

Even so, a general method does not exist for computing the moments of an arbitrary non-explosive birth death process. Substantial progress can be made by observing that most BDPs are well behaved and rarely voyage beyond a certain large population threshold. If we assumed that there is negligible probabilistic distance between the original BDP and a version of the BDP truncated at said large population threshold, we could use standard numerical linear algebra techniques to solve for the moments of the original process. The linear algebra is further aided by the fact that the finite-dimensional truncated stochastic matrix is tridiagonal, as the only transitions from state n are to $n-1$ and $n+1$. A

straightforward coupling argument gives rigor to this heuristic approach.

Once a coupling is established, we derive two different methods for choosing the large truncation index N such that a certain error in the total variation distance between the original and truncated processes is obtained. The first involves solving a series of recurrence relationships for the Laplace transform of the truncated process, and the second involves a straightforward application of Chernoff's bound. The first is more computationally difficult, while the second fails for certain rate functions. Which bound to use depends on the process being considered.

2.2 Coupling bounds and hitting times

We begin by forming a coupling between the initial process $X(t)$ that evolves fully over $\mathbb{N}$ and a truncated version of X, denoted $X_n(t)$. $n \in \mathbb{N}$ is called the truncation level, and the birth and death rates between $X(t)$ and $X_n(t)$ are identical except that for $X_n(t)$, $\lambda_n = 0$. This means that X_n is truncated at n and cannot grow past it. We form a coupling between $X(t)$ and $X_n(t)$ through the following: first, set $X(0) = X_n(0)$. While $X(t) < n$, the coupling stipulates that $X(t)$ and $X_n(t)$ evolve together, meaning $X(t) = X_n(t)$ provided they are below the truncation level. When $X(t) = n$, the processes are still coupled, but now $X(t)$ has probability of jumping upwards to $n + 1$ via a birth event. When this happens, the two processes evolve independently. However, if in the future $X(t)$ drops below n and hits $X_n(t)$, then the two processes resume evolving together such that $X(t) = X_n(t)$. For a review of coupling, see [Lin02]. It is also straightforward to show that all moments satisfy $E[X_n^k(t)] \leq E[X^k(t)]$.

The truncated process $X_n(t)$ approximates the true process $X(t)$. We measure the goodness of the approximation through a total variation (TV) distance. Let $P_{ij}^n(t) = \Pr(X_n(t) = j | X_n(0) = i)$ be the transition probability of the process truncated at n and suppose $X(0) = X_n(0) = i$. Let ν be the measure induced by $P_{ij}(t)$ and let ν_n be the measure

induced by $P_{ij}^n(t)$ with i, t, and n held fixed. That is, for a subset $S \subset \mathbb{N}$,

$$\nu(S) = \sum_{j \in S} P_{ij}(t) \qquad \text{and} \qquad \nu_n(S) = \sum_{j \in S} P_{ij}^n(t). \tag{2.1}$$

The TV distance between the random variables $X(t)$ and $X_n(t)$ is

$$d_{TV}(\nu, \nu_n) = \frac{1}{2} \sum_{j=0}^{\infty} |P_{ij}(t) - P_{ij}^n(t)| = \sup_{S \subset \mathbb{N}} |\Pr(X(t) \in S) - \Pr(X_n(t) \in S)|. \tag{2.2}$$

2.2.1 Hitting Times for the Full Process

Let T_{n+1} be the hitting time at which $X(t)$ first reaches state $n+1$ and let $S = \{0, \ldots, n\}$. Conditioning yields

$$\begin{aligned} \Pr(X(t) \in S) = \Pr(X(t) \in S \mid T_{n+1} > t) \Pr(T_{n+1} > t) \\ + \Pr(X(t) \in S \mid T_{n+1} \leq t) \Pr(T_{n+1} \leq t). \end{aligned} \tag{2.3}$$

Given the event $T_{n+1} > t$, $X(t)$ cannot have exceeded $n+1$ before time t. In this setting $X(t)$ has the same state space and the same transition rates as the truncated process $X_n(t)$. Replacing the first conditional probability produces

$$\begin{aligned} \Pr(X(t) \in S) &= \Pr(X_n(t) \in S) \Pr(T_{n+1} > t) + \Pr(X(t) \in S \mid T_{n+1} \leq t) \Pr(T_{n+1} \leq t) \\ &= \Pr(X_n(t) \in S)[1 - \Pr(T_{n+1} \leq t)] + \Pr(X(t) \in S \mid T_{n+1} \leq t) \Pr(T_{n+1} \leq t) \\ &= \Pr(X_n(t) \in S) + [\Pr(X(t) \in S \mid T_{n+1} \leq t) - \Pr(X_n(t) \in S)] \Pr(T_{n+1} \leq t). \end{aligned} \tag{2.4}$$

Rearranging (2.4) and applying Markov's inequality give

$$\begin{aligned} |\Pr(X(t) \in S) - \Pr(X_n(t) \in S)| &\leq |\Pr(X(t) \in S \mid T_{n+1} \leq t) - \Pr(X_n(t) \in S)| \Pr(T_{n+1} \leq t) \\ &\leq \Pr(T_{n+1} \leq t) \\ &\leq e^{\theta t} E[e^{-\theta T_{n+1}}] \end{aligned} \tag{2.5}$$

for $\theta \geq 0$. The total variation bound $d_{TV}(\nu, \nu_n) \leq e^{\theta t} \, \mathrm{E}(e^{-\theta T_{n+1}})$ is an immediate consequence.

Now let $w_i(\theta) = E(e^{-\theta T_{n+1}} \mid X(0) = i)$. If we condition on the first event of the birth-death process, then we can deduce the recurrences

$$\begin{aligned} w_0(\theta) &= \frac{\lambda_0}{\lambda_0 + \theta} w_1(\theta) \\ w_n(\theta) &= \frac{\mu_n}{\mu_n + \lambda_n + \theta} w_{n-1}(\theta) + \frac{\lambda_n}{\mu_n + \lambda_n + \theta} \end{aligned} \tag{2.6}$$

and, for $1 \leq i \leq n-1$,

$$\begin{aligned} w_i(\theta) &= \frac{\mu_i}{\mu_i + \lambda_i} \frac{\mu_i + \lambda_i}{\mu_i + \lambda_i + \theta} w_{i-1}(\theta) + \frac{\lambda_i}{\mu_i + \lambda_i} \frac{\mu_i + \lambda_i}{\mu_i + \lambda_i + \theta} w_{i+1}(\theta) \\ &= \frac{\mu_i}{\mu_i + \lambda_i + \theta} w_{i-1}(\theta) + \frac{\lambda_i}{\mu_i + \lambda_i + \theta} w_{i+1}(\theta). \end{aligned} \tag{2.7}$$

These can be written as the matrix-vector equation

$$\begin{pmatrix} 1 & a_0 & & & & 0 \\ b_1 & 1 & a_1 & & & \\ & b_2 & 1 & & & \\ & & & \ddots & & \\ & & & & \ddots & a_{n-1} \\ 0 & & & & b_n & 1 \end{pmatrix} \begin{pmatrix} w_0(\theta) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_n(\theta) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \frac{\lambda_n}{\mu_n + \lambda_n + \theta} \end{pmatrix} \tag{2.8}$$

where $a_i = -\frac{\lambda_i}{\mu_i + \lambda_i + \theta}$ and $b_i = -\frac{\mu_i}{\mu_i + \lambda_i + \theta}$. There are well established numerical methods for solving the above tridiagonal system for the expectations $w_i(\theta)$.

According to Chernoff's method, for fixed n one achieves the tightest bound on $d_{TV}(\nu, \nu_n)$ by choosing θ to minimize $e^{\theta t} E(e^{-\theta T_{n+1}})$. The function $f(\theta) = E\left[e^{\theta(t - T_{n+1})}\right]$ is convex with derivative $f'(\theta) = E\left[(t - T_{n+1}) e^{\theta(t - T_{n+1})}\right]$. Since $f'(0) = t - E(T_{n+1})$, the minimum on $[0, \infty)$ occurs at 0 where $f(0) = 1$ whenever $t \geq E(T_{n+1})$. Otherwise, the minimum occurs on the interior of $(0, \infty)$. Minimizing $f(\theta)$ for $\theta \geq 0$ can be achieved via Newton's method. The derivatives $\omega_i'(\theta)$ are calculated via a tridiagonal system similar to equation (2.8) derived by differentiating equations (2.6) and (2.7). For fixed n, these maneuvers allow one to calculate the bound

$$d_{TV}(\nu, \nu_n) \leq \min_{\theta \geq 0} e^{\theta t} E(e^{-\theta T_{n+1}}). \tag{2.9}$$

As $n \to \infty$, $E(e^{-\theta T_{n+1}}) \to 0$ for any $\theta > 0$. Therefore increasing n will decrease the bound until a desired error $d_{TV}(\nu, \nu_n) < \epsilon$ is achieved.

2.3 Explicit Chernoff bound

Calculation of hitting probabilities via equation (2.8) is appealing, but it can be computationally burdensome. We can perform a simpler coupling that gives a looser bound by considering a pure-birth process $Y(t)$ that shares its birth rates with $X(t)$ but has all death rates $\mu_j = 0$. We form the coupling by initialing having $X(0) = Y(0)$. Again, when $X(t) = Y(t)$, the processes move together if the next event is a birth event. If the processes get separated through a death event, they evolve independently. Likewise if at any point $X(t) = Y(t)$, the coupling is re-established. Obviously $X(t) \leq Y(t)$ for all $t > 0$, hence $Y(t)$ furnishes an upper bound for $X(t)$.

We assume that $Y(t)$ is non-explosive such that it remains finite with probability 1. This is equivalent to $\sum_{i=0}^{\infty} \lambda_i^{-1} = \infty$. This in turn means $X(t)$ is non-explosive. This is required for guaranteeing that the coupling between $X(t)$ and the truncated version $X_n(t)$ introduced in the previous section is valid. It will be useful to show that $\lim_{n\to\infty} X_n(t) = X(t)$:

Proposition 1. *Consider a BDP $X(t)$ with $X(0) = i$, and assume the corresponding pure birth process $Y(t)$ is nonexplosive. Then the truncated processes $X_n(t)$ are increasing in n and converge to X_t as $n \to \infty$. Furthermore, the conditional moments $E[X_n^k(t)|X_n(0) = i]$ converge monotonically to $E[X^k(t)|X(0) = i]$, and the transition probabilities $\Pr(X_n(t) = j|X_n(0) = i)$ converge to $\Pr(X(t) = j|X(0) = i)$.*

Proof. If the increasing sequence $X_n(t)$ converges to $X(t)$, then the limit relation

$$\lim_{n\to\infty} E[X_n^k(t)|X_n(0) = i] = E[X^k(t)|X(0) = i] \tag{2.10}$$

follows from the monotone convergence theorem. Likewise, the limit relation

$$\lim_{n\to\infty} \Pr(X_n(t) = j|X_n(0) = i) = \Pr(X(t) = j|X(0) = i) \tag{2.11}$$

follows from the bounded convergence theorem since each transition probability in question is the expectation of an indicator random variable and as such is bounded by the constant 1.

To show that the truncated processes are increasing, imagine that the $X_n(t)$ are increasing and occur in blocks. Thus, $X_m(t)$ and $X_n(t)$ occupy the same block whenever $X_m(t) = X_n(t)$. Different blocks behave independently. Within a block the processes move as one except when $X_n(t)$ in a block takes its maximal value n. In this case a proposed birth is declined and the process $X_n(t)$ stays in place. Monotonicity is preserved by this protocol for updating the participating processes. Note the total number of blocks at any time $s \leq t$ is bounded above by the finite number $Y(s) \leq Y(t)$. Finally, convergence can be proved by considering a typical realization of the combined processes. The state $Y(t)$ represents the outermost limit that $X(s)$ or any $X_n(s)$ can reach during $[0, t]$. Hence, $X_n(t) = X(t)$ as soon as n exceeds $Y(t)$. In other words, $X_n(t)$ converges monotonically to $X(t)$. □

Now that the limiting behavior is well established, we can use the pure-birth process $Y(t)$ to obtain an analytical TV bound between the original and truncated processes.

Proposition 2. *Suppose $X(0) = X_n(0) = Y(0) = i < n$ and $t < \sum_{m=i}^{n} \lambda_m^{-1}$. Then the total variation distance between the $X(t)$ and $X_n(t)$ is bounded above,*

$$d_{TV}(\nu, \nu_n) \leq e^{st} \prod_{m=i}^{n} \frac{\lambda_m}{\lambda_m + s}, \tag{2.12}$$

where $s > 0$ satisfies $t = \sum_{m=i}^{n} (\lambda_m + s)^{-1}$.

Proof. First, the bound

$$\begin{aligned} d_{TV}(\nu, \nu_n) &\leq \Pr(X(t) \neq X_n(t) \mid X(0) = X_n(0) = i) \\ &\leq \Pr(Y(t) > n \mid Y(0) = i) \end{aligned} \tag{2.13}$$

applies because the event $X(t) \neq X_n(t)$ is contained in the event $Y(t) > n$. Furthermore, $Y(t) > n$ is equivalent to the event $W_i + \cdots + W_n \leq t$, where W_j is the exponential waiting time from the arrival of the pure-birth process $Y(t)$ in state j until it moves to $j+1$. Applying the Chernoff bound [Lan10], we have

$$\Pr(W_i + \cdots + W_n \leq t) \leq \min_{s \geq 0} \; e^{st} \prod_{m=i}^{n} E\left[e^{-sW_m}\right] \tag{2.14}$$

Because W_i has exponential distribution with rate λ_i, its Laplace transform $E[e^{-sW_i}] = \lambda_m/(\lambda_m + s)$. It follows that

$$\Pr(W_i + \cdots + W_n \le t) \le \min_{s \ge 0} \; e^{st} \prod_{m=i}^{n} \frac{\lambda_m}{\lambda_m + s}, \tag{2.15}$$

and the optimal s for a given n minimizes the convex function

$$f_n(s) = st - \sum_{m=i}^{n} \log(\lambda_m + s). \tag{2.16}$$

Differentiating $f_n(s)$ with respect to s, we find that the optimal s satisfies

$$t = \sum_{m=i}^{n} \frac{1}{\lambda_m + s} \tag{2.17}$$

for $t < \sum_{m=i}^{n} \lambda_m^{-1}$, as claimed. □

The condition $t < \sum_{m=i}^{n} \lambda_m^{-1}$ guarantees that t is less than the expected first passage time of the pure-birth process from i to n. Clearly if this were not true, we would expect $Y(t) > n$ with high probability, and hence approximation of $X(t)$ by $X_n(t)$ would be poor.

The representation

$$e^{st} \prod_{m=i}^{n} \frac{\lambda_m}{\lambda_m + s} = \exp\left[st - \sum_{m=i}^{n} \log\left(1 + \frac{s}{\lambda_m}\right)\right] \tag{2.18}$$

suggests exploiting the logarithmic inequality $-\log(1+u) \le -u + u^2/2$ for $u \in [0, \infty)$. It follows that

$$e^{st} \prod_{m=i}^{n} \frac{\lambda_m}{\lambda_m + s} \le \exp\left[st + \sum_{m=i}^{n} \left[-\frac{s}{\lambda_m} + \frac{1}{2}\left(\frac{s}{\lambda_m}\right)^2 \right] \right]. \tag{2.19}$$

The right-hand side of this inequality is minimized by the choice

$$s = \frac{\sum_{m=i}^{n} \lambda_m^{-1} - t}{\sum_{m=i}^{n} \lambda_m^{-2}}. \tag{2.20}$$

Inserting this argument in the previous inequality produces the upper bound

$$d_{TV}(\nu, \nu_n) \le e^{-\frac{a_n^2}{2b_n}}, \tag{2.21}$$

where $a_n = \sum_{m=i}^{n} \lambda_m^{-1} - t$ and $b_n = \sum_{m=i}^{n} \lambda_m^{-2}$. If the BDP is nonexplosive, $\lim_{n\to\infty} a_n = \infty$. When the condition

$$\lim_{n\to\infty} \frac{\left(\sum_{m=1}^{n} \lambda_m^{-1}\right)^2}{\sum_{m=i}^{n} \lambda_m^{-2}} = \infty \tag{2.22}$$

holds, one can easily find a sufficiently large n that renders $d_{TV}(\nu, \nu_n)$ as small as desired.

2.3.1 Matrix exponentiation for computing moments

One approach to calculating moments and probabilities is to substitute the truncated process just described for the actual process. Given the tridiagonal nature of the infinitesimal generator, this will deliver accurate estimates at a computational cost of $O(n^2)$ [DP04]. Let us assume that the original process possesses a well-behaved equilibrium distribution $\boldsymbol{\pi}$. Note this precludes transient behavior. The first task is to approximate $\boldsymbol{\pi}$ accurately. Detailed balance gives

$$\begin{aligned} \pi_0 &= \frac{1}{1+\sum_{k=1}^{\infty}\prod_{j=1}^{k}\frac{\lambda_{j-1}}{\mu_j}} \\ \pi_k &= \pi_0\prod_{j=1}^{k}\frac{\lambda_{j-1}}{\mu_j} \quad \text{for} \quad k>0. \end{aligned}$$

The implied products and sums can be computed in $O(n)$ operations. The point at which the approximation to π_0 stabilizes gives a trial value of n.

Now consider the process $X_n(t)$ with truncated equilibrium distribution $\tilde{\boldsymbol{\pi}}$ and truncated infinitesimal generator $\tilde{\boldsymbol{A}}$ with $\lambda_n = 0$. Owing to detailed balance, the matrix $\boldsymbol{B} = \sqrt{\tilde{\boldsymbol{\pi}}}\tilde{\boldsymbol{A}}\sqrt{\tilde{\boldsymbol{\pi}}}^{-1}$ is symmetric and tridiagonal, where $\sqrt{\tilde{\boldsymbol{\pi}}}$ denotes the diagonal matrix with kth diagonal entry $\sqrt{\pi_k}$. The spectral decomposition

$$\boldsymbol{B} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^t = \sum_{k=1}^{n} d_k\boldsymbol{v}_k\boldsymbol{v}_k^t$$

of $\boldsymbol{B}$ can be computed very efficiently [DMP08]. Furthermore, the matrix exponential

$$e^{s\boldsymbol{B}} = \sum_{k=1}^{n} e^{sd_k}\boldsymbol{v}_k\boldsymbol{v}_k^t$$

follows directly and gives

$$e^{s\tilde{\boldsymbol{A}}}\boldsymbol{u} = \sum_{k=1}^{n} e^{sd_k}(\sqrt{\tilde{\boldsymbol{\pi}}}\boldsymbol{v}_k)^t\boldsymbol{u}(\sqrt{\tilde{\boldsymbol{\pi}}}^{-1}\boldsymbol{v}_k)$$

for any vector $\boldsymbol{u}$. Thus, all truncated transition probabilities and moments are available at little additional cost. As we have emphasized, the truncated moments bound the moments of the original process from below, providing an easily-computed criterion for finding the truncation index n.

2.4 Examples

2.4.1 Logistic Growth

[CS12] give a BDP for logistic population growth,

$$\lambda_j = j^2\lambda\frac{e^{-\alpha j}}{1+e^{\beta(j-M)}} \qquad \text{and} \qquad \mu_j = j\mu$$

where M is the "carrying capacity" of the environment and α and β are non-negative parameters. This model seems to have no analytic solution, but we can easily find bounds for matrix truncation:

$$d_{TV}(\nu,\nu_n) \leq \exp\left\{-\frac{\left(\sum_{m=i}^{n}\left[m^2\lambda e^{-\alpha m}/(1+e^{\beta(m-M)})\right]^{-1}-t\right)^2}{2\sum_{m=i}^{n}[m^2\lambda e^{-\alpha m}/(1+e^{\beta(m-M)})]^{-2}}\right\}. \tag{2.23}$$

The birth rates in this example do not satisfy the condition (2.22) for the Chernoff bound to converge. Figure 2.1 illustrates the superior performance of the Laplace bound.

2.4.2 Faddy Distributions

Pure-birth processes are a special case of BDPs with $\mu_j = 0$ for all $j \in \mathbb{N}$. [Fad97] proposes a generalization of the Poisson distribution with $\lambda_j = \lambda(\gamma+j)^c$. Setting $c = 0$ recovers the Poisson process, $c > 0$ indicates over-dispersion, and $c < 0$ indicates under-dispersion relative to the Poisson distribution. [Fad97] employs a diffusion approximation to compute moments of the process, but the bounds derived in this paper apply directly. The Chernoff bound is

$$d_{TV}(\nu,\nu_n) \leq \exp\left\{-\frac{[\sum_{m=i}^{n}(\gamma+m)^{-c}-\lambda t]^2}{2\sum_{m=i}^{n}(\gamma+m)^{-2c}}\right\}. \tag{2.24}$$

Figure 2.2 shows examples of the bounds.

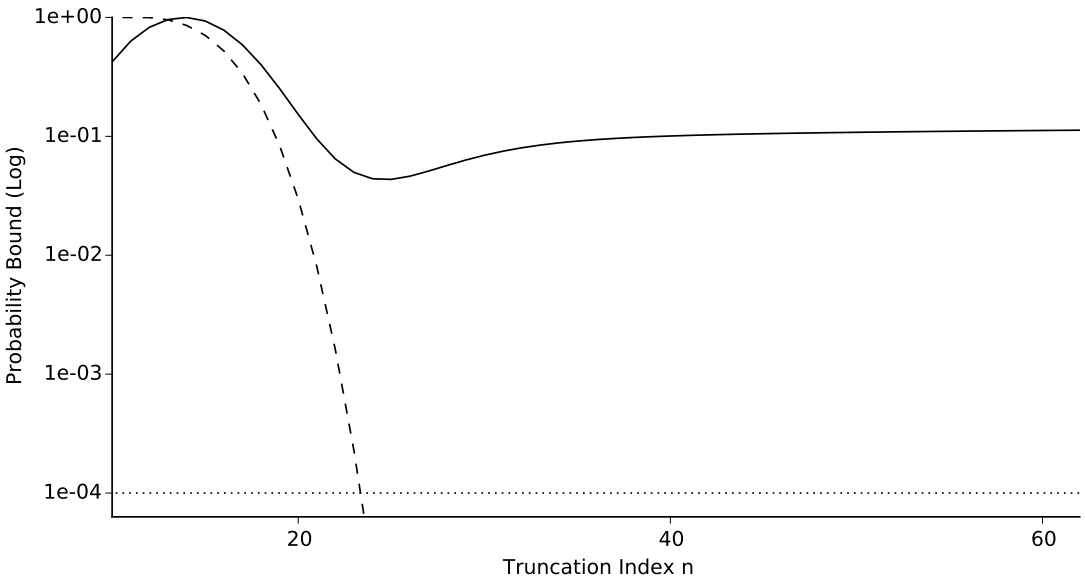

Figure 2.1: Probability bounds on the log scale for the total variation distance obtained between the Logistic process truncated at index n and the untruncated process, where $X(0) = i = 4$, $t = 1$, $\alpha = 0.2$, $\beta = 0.3$, $M = 20$, $\mu = 1$, and $\lambda = 1$. The solid line shows the derived bound (2.23). The dashed line shows the bounds obtained from the Laplace transforms (2.9). The dotted line shows the error tolerance $\epsilon = 10^{-4}$.

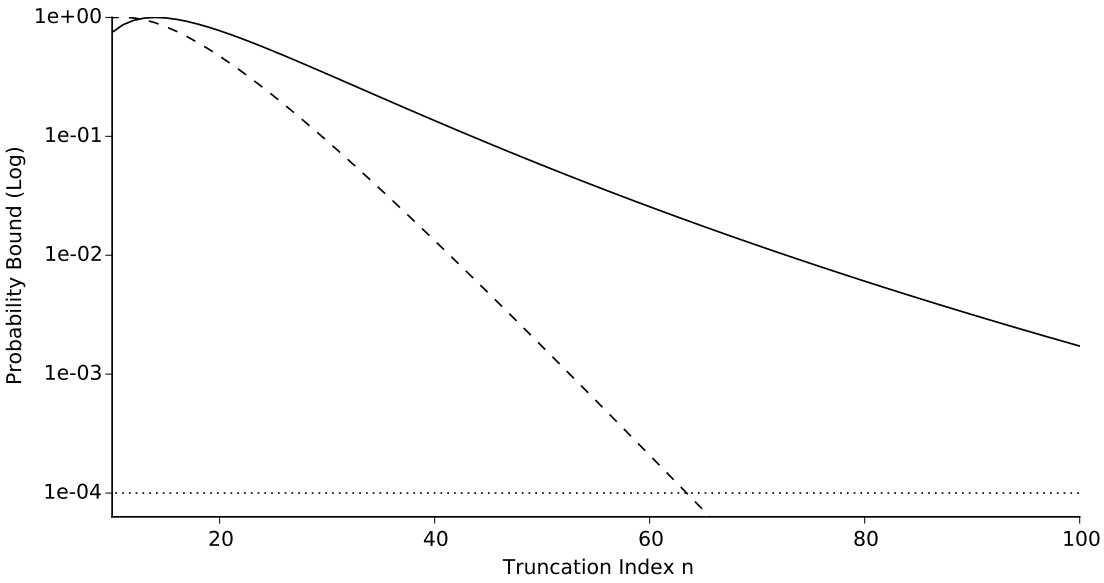

Figure 2.2: Probability bounds on the log scale for the total variation distance obtained between the Faddy process truncated at index n and the untruncated process, where $X(0) = i = 4$, $t = 1$, $\gamma = 1.1$, $\mu = 1$, and $\lambda = 1$. The solid line shows the derived bound (2.24). The dashed line shows the bounds obtained from the Laplace transforms (2.9). The dotted line shows the error tolerance $\epsilon = 10^{-4}$.

CHAPTER 3

Stochastic Simulation Algorithms for Interacting Particle Systems (IPSs)

3.1 Background

Stochastic effects are crucial for accurately modeling evolutionary and biological processes such as tumor growth, desertification, disease spread, embryonic development, maintenance of species biodiversity, and pattern formation in general [WBP15, KRA07, RMF07]. The associated spatial mathematical models are commonly analytically intractable. Fortunately the advent of efficient computing has allowed simulation to serve as a common first approach to stochastic modeling. Non-spatial well-mixed versions of these models are often substituted due to their tractibility and ease of use. Many celebrated simulation algorithms such as the exact Stochastic Simulation Algorithm (SSA), τ-leaping, and the next-reaction method have been developed and extensively modified to address a wide range of well-mixed stochastic phenomena [AG07]. However, well-mixed models fail to capture the appropriate statistics and pattern formation seen in the spatial setting. Phenomenon due to volume exclusion and spatial dispersion cannot be accurately captured via well-mixed Chemical Reaction Network (CRN) simulation. One common approach to stochastic spatial simulation is to partition the spatial domain into well-mixed voxels. This approach utilizes a Reaction-Diffusion Master Equation (RDME) to model the movement of particles between voxels and the reaction of particles within the same voxel. While this method has substantial algorithmic development [GHP13], it fails to take into account important volume exclusion effects and fine-grained spatial variation.

Interacting Particle Systems (IPSs) provide an alternative to both well-mixed CRN and

RDME based modeling. IPSs are a class of stochastic models with full spatial detail, tracking each particle's location on a lattice [Lig12]. Interactions are assumed to be local, meaning particles must be adjacent to each other to interact. Notions of locality and adjacency are details that must be specified in a given model. For a set of example reactions, see Table 3.1. Importantly, IPSs preserve volume exclusion, meaning at most one particle can be present on any given lattice site. Diffusive movement is typically modeled as particles undergoing random walks between sites, respecting exclusion. This is in contrast with more common RDME approaches that couple compartments obeying well-mixed dynamics through non-excluding Brownian motion. Example IPSs include the voter and contact processes as well as the classic Ising model from statistical mechanics [Lig13]. These specific models have a large body of theoretical results from the mathematics community, specifically on their critical behavior. Unfortunately these results do not readily extend to multi-type processes and complicated spatial domains. Numerical approaches are computationally prohibitive, leaving direct simulation as the first and frequently only line of attack.

Type	Example reactions	Processes
On-site	$\emptyset \to A$	Immigration
	$A \to \emptyset$	Death
	$A \to B$	Transformation
Pairwise	$A + \emptyset \to A + \emptyset$	Migration
	$A + \emptyset \to A + A$	Binary fission
	$A + A \to B + \emptyset$	Dimerization
	$A + B \to C + D$	Pairwise transformation

Table 3.1: Example processes with reaction diagrams. $\emptyset$ denotes an open site that is part of a reaction.

The current paper extends the classic n-fold simulation method, defined later, to IPSs [BKL75]. Our extension enjoys three major advantages over previous approaches. First,

we generate the minimum number of required reaction channels for a simulation, avoiding the combinatorial difficulties that arise from counting adjacent configurations of particles. Second, we provide efficient local updates after a reaction channel fires; thus only particles adjacent to a reaction are updated. Critically, this prevents the computational complexity of the simulations from scaling with the size of the lattice. Third, and perhaps most important, we separate the time and reaction sampling steps from the configuration update steps in the algorithm. This reduces our spatial process to the computational complexity of a CRN simulation, albeit with an additional complicated update. Accordingly, we can implement any CRN sampling algorithm for our spatial setting with little additional effort. Well-mixed CRN simulation is extensively developed [Gil77, Gil01, CGP06, SAL09, MB07, And08, ACK06]; therefore, spatial IPSs directly benefit from these prior innovations.

We build on the software package `BioSimulator` [LSK18], written in the open source programming language `Julia` [BEK17]. `BioSimulator` implements different algorithms for simulating IPSs, including the exact stochastic simulation algorithm (SSA) and versions of the next reaction method and the sorting direct method [MPC06]. Our software provides a simple, intuitive interface through which nonspecialists can quickly observe complex behaviors of spatial models with multiple interacting species. Summary statistics and particle count trajectories permit straightforward model checking for the proposed systems. Within this framework, modelers can determine which reactions and parameters are important for producing a certain desired behavior. A recent example of an IPS in action has been reported in a recent immunotherapy model for cancer treatment [KPS17]. This complex model of tumor-immune system interactions illustrates which parameters generate the appropriate immune responses and spatial patterns.

Our software is primarily directed at systems biologists, cancer researchers, ecologists, evolutionary biologists, epidemiologists, and other scientists who are interested in the spatio-temporal effects of discrete actors. We anticipate that `BioSimulator`'s ease of use and flexibility will encourage researchers unfamiliar with stochastic processes to investigate the stochastic and spatial features of their models via simulation. Additionally, we expect that applied probabilists, physicists working in statistical mechanics, and mathematicians familiar

with the field of stochastic simulation can take advantage of our software and avoid tedious re-implementation of different algorithms.

The remaining exposition is organized as follows. First we give a mathematical description of IPSs. We then enumerate the different sample classes for probabilistically equivalent particles using the species types and neighborhood configurations of the lattice or graph. This enumeration plus a description of the reaction rates across these sample classes provides a straightforward means of extending the well-mixed SSA to IPSs. Lastly, we summarize how our software implements each reaction, including updating the sample classes and reaction rates. This is followed by a series of examples of complex, multi-species spatial stochastic phenomenon. We conclude with a brief description of the benefits of writing `BioSimulator` in the `Julia` programming language.

3.2 Design and Implementation

3.2.1 IPSs and pairwise reactions

An IPS models a collection of particles moving and reacting stochastically over some spatial domain. Particles are discrete entities that may model animals, proteins, wildfire patches, or cancer cells. Like well-mixed CRNs, these particles interact through a series of reaction channels. While stochastic CRNs assume every particle interacts uniformly with every other particle, IPSs restrict these interactions to neighboring particles. Each IPS has an associated graph describing the spatial domain over which the process evolves. Nodes on the graph are sites that a particle may occupy. Edges specify that two nodes are adjacent and hence liable to interact. Typically we restrict nodes to contain at most one particle at a time; we refer to this effect as *volume exclusion*.

Fortunately, embedding the IPSs on a graph allows us to restrict the reactions to being *pairwise*. We use the term pairwise instead of bimolecular deliberately; unimolecular reactions that produce two product particles require an open adjacent site due to the volume excluding effect. For example, birth through binary fission is written in well mixed reaction

notation as $A \to A + A$. On a graph with exclusion, birth requires an open adjacent site and becomes $A + \emptyset \to A + A$ where $\emptyset$ denotes an open site that becomes occupied by one of the offspring particles. This schema emphasizes volume exclusion since birth cannot occur when the A particle has no open adjacent sites. We classify reactions into two groups, on-site and pairwise. For a non-exhaustive list of examples, see Table 3.1; for a specific predator-prey example see Table 3.2.

Name	Diagram	Type
Fox Predation	$F + R \to F + F$	Pairwise
Fox Migration	$F + \emptyset \to \emptyset + F$	Pairwise
Fox Death	$F \to \emptyset$	On-site
Rabbit Reproduction	$R + \emptyset \to R + R$	Pairwise
Rabbit Migration	$R + \emptyset \to \emptyset + R$	Pairwise
Rabbit Death	$R \to \emptyset$	On-site

Table 3.2: Predator-prey reactions. Foxes (F) and rabbits (R) interact on a 2D hexagonal lattice with open sites ($\emptyset$). Reactions are either on-site involving a single animal interacting only with itself, or pairwise involving an animal interacting with an adjacent site.

These two reaction types, on-site and pairwise, are sufficient for describing the majority of biological applications. Higher-order reactions are reduced to pairwise interactions through the formation of intermediate complexes.

3.2.2 Markovian dynamics, reaction channels, and sample classes

Particles evolve on the graph according to standard Markovian dynamics where the waiting time to the next reaction is exponentially distributed [Lan10]. If a particle can take part in multiple reactions, then its exponential waiting time has rate equal to the sum of the rates of each individual reaction under mass-action kinetics. Note that more complicated kinetics

are allowed provided that we restrict the interactions to neighboring particles. Longer range interactions are feasible in principle, though they introduce combinatorial complexity in enumerating the neighboring configurations. The current version of `BioSimulator` is restricted to mass-action kinetics for immediate neighbors.

The rate at which a particle undergoes reactions depends on both the species of the particle and the number and species of its neighboring particles. Although open sites are not collectively considered a species, open sites next to occupied sites play a negative role in volume exclusion. In order to draw parallels with well-mixed CRNs, we split each pairwise reaction into a series of *reaction channels*. Each pairwise reaction channel is associated with a center particle interacting with up to D neighboring particles of the appropriate type, where D is the number of adjacent neighbors. D takes the values 4, 6, and 8, respectively, on a square planar lattice, a hexagonal planar lattice, and 3-dimensional cubic lattice. Therefore the total number of reaction channels is $R = D \times \#$ pairwise reactions $+ \#$ on-site reactions. See Figure 3.1 for a depiction of a predator-prey process involving foxes and rabbits on a hexagonal lattice and Table 3.3 for its associated reaction channels. For instance, when the third predation reaction channel fires, the simulation searches for a fox adjacent to exactly three rabbits to undergo the predation.

There are two approaches to sampling a reaction channel and associated particle. The more rudimentary approach is to scan through the particles in the lattice, sum the per-particle reaction rates, and select a particular particle to fire with probability proportional to its contribution to this sum [CV07]. A more sophisticated method is given by Bortz, Kalos, and Lebowitz under the n-fold way [BKL75]. Here particles are grouped into classes such that all members of a given class take part in a specific reaction with the same rate. This explicitly forms a series of reaction channels for sampling using Markovian dynamics and avoids time consuming searches of the lattice.

We provide an extension to the n-fold way that decouples the sampling of the reaction channels, an inherently non-spatial maneuver, from the sampling of a particle to undergo the reaction. This in turn separates the spatial dependencies inherent in IPSs from the Markovian dynamics of the reaction channels. Thus, spatial correlations are handled during

Reactants		Products		Per Particle Rate	Sample Index
F	R	F	F	α	1
F	R	F	F	2α	2
F	R	F	F	3α	3
F	R	F	F	4α	4
F	R	F	F	5α	5
F	R	F	F	6α	6
R	$\emptyset$	R	R	β	7
R	$\emptyset$	R	R	2β	8
R	$\emptyset$	R	R	3β	9
R	$\emptyset$	R	R	4β	10
R	$\emptyset$	R	R	5β	11
R	$\emptyset$	R	R	6β	12
F	$\emptyset$	$\emptyset$	F	γ	13
F	$\emptyset$	$\emptyset$	F	2γ	14
F	$\emptyset$	$\emptyset$	F	3γ	15
F	$\emptyset$	$\emptyset$	F	4γ	16
F	$\emptyset$	$\emptyset$	F	5γ	17
F	$\emptyset$	$\emptyset$	F	6γ	18
R	$\emptyset$	$\emptyset$	R	γ	7
R	$\emptyset$	$\emptyset$	R	2γ	8
R	$\emptyset$	$\emptyset$	R	3γ	9
R	$\emptyset$	$\emptyset$	R	4γ	10
R	$\emptyset$	$\emptyset$	R	5γ	11
R	$\emptyset$	$\emptyset$	R	6γ	12
F		$\emptyset$		μ	19
R		$\emptyset$		μ	20

Table 3.3: Reaction channels and associated sample indices. Each initial pairwise reaction in Fig. 1a is split into six reaction channels, one for each number of adjacent reactants. Each reaction channel has an associated per particle rate and sample index. This sample index points to the collection of particles that the reaction channel samples a reactant from. The total rate of each reaction channel is equal to the per animal rate times the number of animals in the associated sample class. Note that the rabbit reproduction and migration channels share the same sample indices because they share the same reactants.

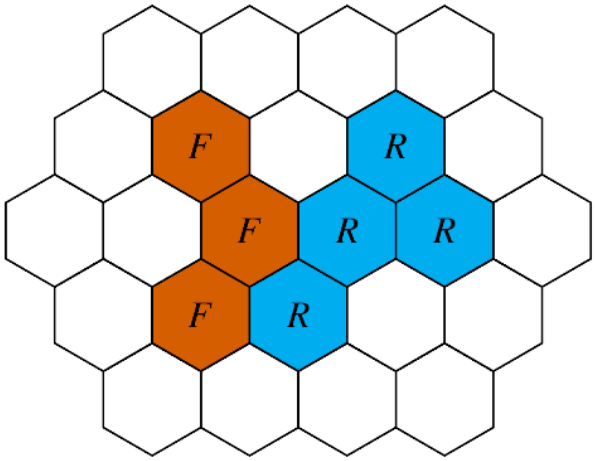

Figure 3.1: Initial configuration: Sites are color coded by occupancy; vermillion denotes a fox F, and cyan denotes a rabbit R. A site can be occupied by at most one animal at a time. Open sites are left blank.

the update step. We do this via generating *sample classes*, which are collections of particles that can be sampled by one or more reaction channels. The sample classes are motivated by the observation that the exact configuration of neighboring particles does not matter for a given reaction channel firing. Only the number of neighboring particles of the appropriate type influence the reaction rate. Therefore each sample class contains particles of a specific species that are adjacent to a specific number of particles of a type that the particle under consideration can react with.

This is best demonstrated by an example; see the sample classes associated with each reaction channel in Table 3.3. The rabbits in the predator-prey example are sorted into seven different sample classes, numbers 7 through 12 and 20, one for each central rabbit interacting with one to six open adjacent sites and a final class containing only rabbits. As an example for the pairwise reaction channels, sample class 9 contains rabbits adjacent to three open sites. This sample class is targeted by two different reaction channels, one for rabbit migration with three neighbors and one for rabbit reproduction with three neighbors. Likewise there is a sample class associated with each on-site reaction; sample class 20 contains every rabbit that can undergo death.

Because multiple reaction channels may sample particles from the same sample class, the total number of sample classes is less than or equal to the number of reaction channels. Specifically, the number sample classes is equal to $D\times$ # unique pairs of reactants + # unique on-site reactants. Using the list of reactants, we assign each reaction channel to its appropriate sample class. Multiple reaction channels will map to the same sample class when the reactions use the same pair of reactants, for example rabbit migration and reproduction in Table 3.3.

3.2.3 Local updates

For a reaction channel to fire, a particle is sampled uniformly from the appropriate sample class. This particle, possibly along with a neighbor of the appropriate interacting species, then undergoes the reaction. At this point, the reaction rates must be updated to reflect the changing configuration. Again there are two possible methods for updating these rates [CV07]. The first, called a *global update*, scans the entire lattice grouping particles into classes and calculating reaction rates. While straightforward, this is inefficient due to the fact that configuration changes take place over at most two neighboring particles. In contrast, we use a *local update* that changes the rates associated with particles immediately adjacent to or involved in the reaction. There is overhead associated with sorting the particles by class and the local updates, but these improvements prevent the simulation step from scaling with the number of particles in computational complexity.

We now expand upon how we perform local updates. Because each particle's behavior depends only on its adjacent particles, it suffices to enumerate these different neighborhood configurations. Specifically, we count the number of ways L species can be distributed across D neighboring sites. The standard stars-and-bars argument shows that the total number of configurations K is equal to the binomial coefficient $\binom{D+L}{L}$. Highly efficient algorithms exist to systematically enumerate all configurations [NW14]. For example, with $D = 4$ neighbors and $L = 2$ species, the configuration $1 + 1 + 2$ corresponds to one open adjacent site, one adjacent particle of the first type, and two adjacent particles of the second type. See Figure

3.2 for an example predator-prey model using the neighborhood configurations.

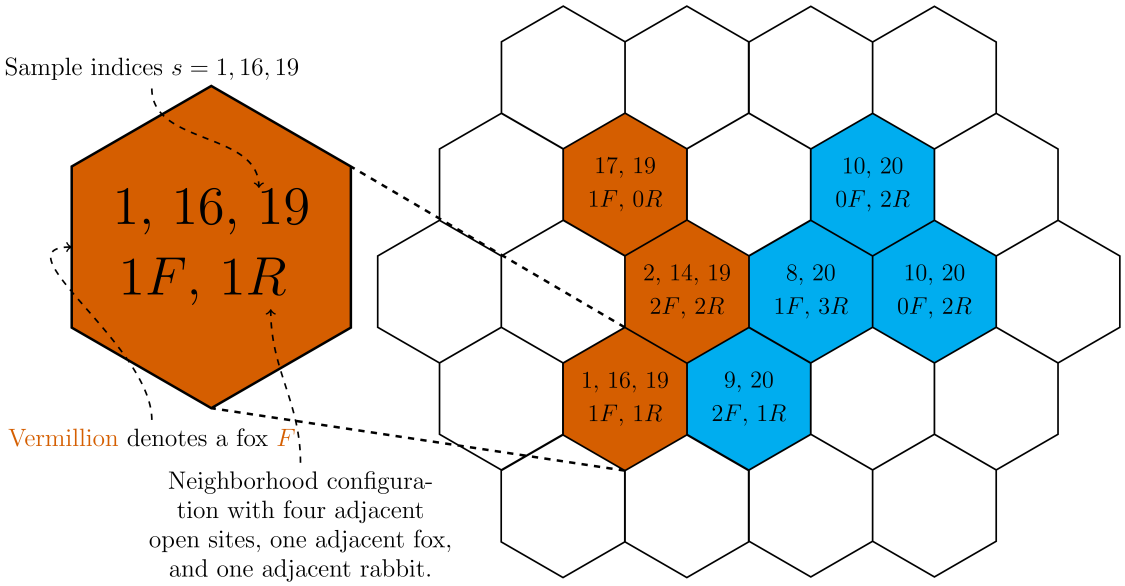

Figure 3.2: Initial configuration with sample indices and neighborhood configurations. Note that the number of adjacent open sites can be inferred from the number of foxes and rabbits.

The naive approach to sampling particles for each reaction channel would be to group particles together by species and neighborhood configuration $k \in \{1, 2, ..., K\}$. However, K scales factorially with the number of species in the simulation. This scaling issue further motivates our previous discussion of the sample classes, which scale with the number of reaction channels. We therefore restrict the use of neighborhood configurations purely for updating the sample classes after a reaction has occurred. We will now expand on how the neighborhood configurations, sample classes, and reaction channels interact.

Figure 3.3 provides an example of the local update procedure after a reaction channel has been chosen to fire. In this scenario, reaction channel 1 is firing, meaning the simulation searches for a fox F adjacent to exactly one rabbit R to undergo the predation reaction. Foxes that satisfy this condition are contained within sample class 1 as denoted in Figure **??**. Suppose the bolded fox in the first configuration of Figure 3.3 is sampled from sample class 1 to undergo the reaction. Since it has only one adjacent rabbit, also bolded, this rabbit is likewise sampled to be the target of the predation reaction. At this point the rabbit changes type to a fox, shifting from vermillion to cyan. The neighboring indices and sample

classes of both particles and their adjacent species now are updated to reflect the rabbit changing type. For example, the sampled fox loses an adjacent rabbit and is removed from sample class 1 to reflect this change. Lastly we update the rates of the reaction channels that have changed in terms of numbers of associated particles.

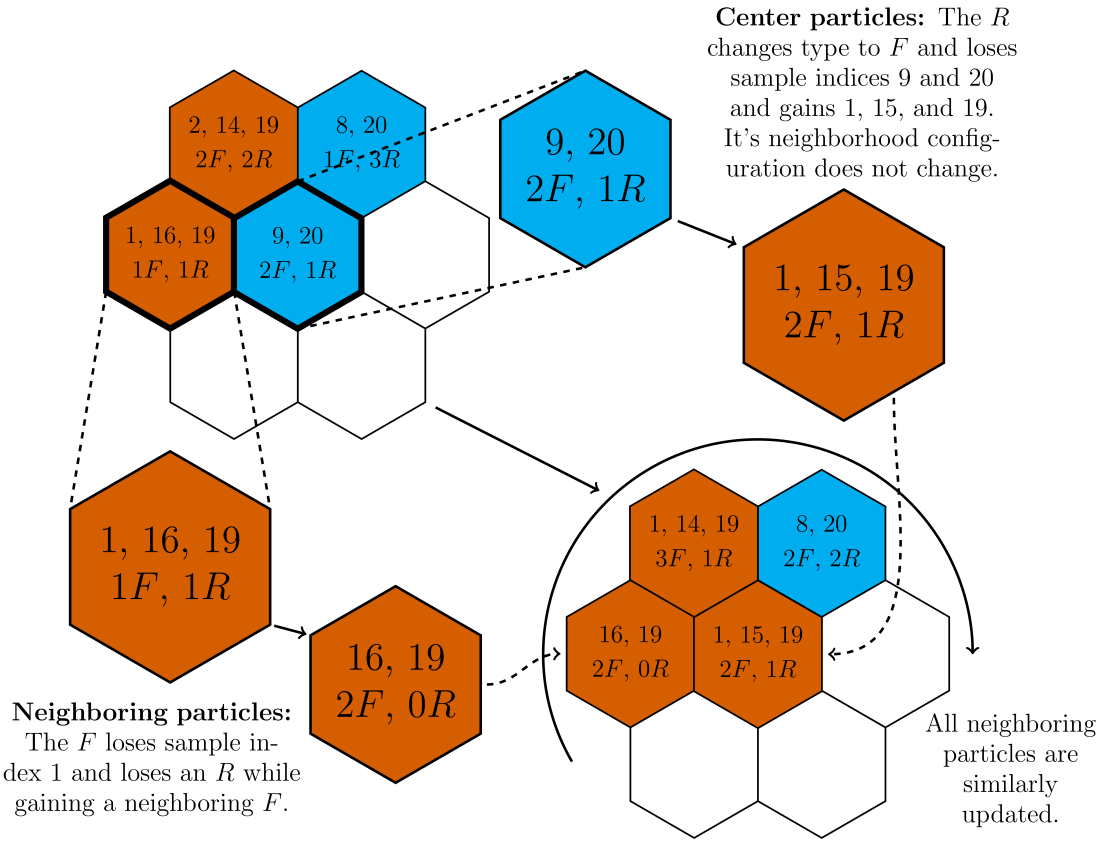

Figure 3.3: Updating the neighborhood and sample indices after a reaction. Suppose the highlighted fox and rabbit sites undergo a predation event. The rabbit is replaced with a fox, and so the neighborhood and sample indices of the sites surrounding the former R require updating to reflect the new configuration. The sample indices of the new F will change as well, but its neighborhood index will not. This update procedure need only be done for and around sites that change species.

This approach has two major benefits. First, it decouples the size of the simulation from the size of the lattice. The most intensive operations required are searches to find a particle in a given sample class, which scale $O(\log n)$ with the number of elements n in the sample

class. Second, this decouples the sampling algorithm from the update step, allowing us to extend our approach to arbitrary simulation algorithms.

3.2.4 Extension to arbitrary simulation algorithms

We begin by demonstrating how our IPS sampling method maps neatly onto the SSA. Let $r = 1, 2, ..., R$ denote the index of a reaction channel, and let λ_r denote the associated reaction rate for the r-th reaction channel. $\lambda_0 = \sum_r \lambda_r$ is the total reaction rate for the process. Given U_1 and U_2 independent uniform $[0, 1]$ random variables, we determine the time T to the next reaction and the next reaction channel j to fire by the conditions

$$T = -\frac{\log(U_1)}{\lambda_0} \quad \text{and} \quad \sum_{r=1}^{j-1} \lambda_r < U_2 \lambda_0 \leq \sum_{r=1}^{j} \lambda_r.$$

Since the update step is kept separate from the time and reaction channel sampling steps, we are able to decouple the stochastic simulation algorithm of choice from the spatial considerations of the system. This applies to arbitrary simulation algorithms. For example, τ-leaping proceeds exactly as described in [CGP06]: the time increment is chosen to satisfy the leap condition, and a Poisson number of events from each reaction channel is chosen to fire. However, the update step is no longer commutative as updates after a reaction must be carried out sequentially. We cannot sum the total changes to the sample classes in the same fashion as in the well-mixed case. Thus a small additional overhead is needed to randomly shuffle the order in which each reaction channel fires. Similarly we can use a dependency graph to restrict the reaction rates that are updated after each event to the subset that is dependent on the fired reaction channel. As mentioned earlier, we will follow this manuscript with an extensive review of the many different available well-mixed simulation algorithms applied to IPSs.

3.3 Results

We provide four example simulation outputs generated by our software for models of varying complexity. Each demonstrates a phenomenon that is observed in the spatial IPS version

of the process but not in the well-mixed CRN version. We also provide `Jupyter` notebooks [Per18] that were used to generate each animation at `https://github.com/alanderos91/BioSimulator.jl`. Additional tutorial notebooks explaining syntax, model construction, and simulation output are also provided through the link. For a list of reactions and parameters for each example, see the supplementary materials, Tables I-IV.

First, we have the predator-prey model previously described in Figure **??**. The output from the simulation is visualized in Figure 3.4. An animation of Figure 3.4 shows spiral wave patterns created by the prey migrating into unoccupied areas while being chased by predators, see the supplementary materials. The predators at the end of the wave die off, leaving space for the prey to migrate into and repeat the process. In this example, spatial dispersion promotes increased biodiversity. It prevents the large spike in predators that can lead to extinction or dramatic fluctuations in the number of predators and prey commonly seen in the CRN version of the model.

Second, we have a three species rock-paper-scissors game shown in Figure 3.5. Each species undergoes a birth-death-migration process and has an additional predation reaction: rock preys on scissors, scissors prey on paper, and paper preys on rock. Spiral wave patterns are also observed in the animation. Spatial dispersal similarly maintains biodiversity. Migration at a high rate can destroy this diversity as the populations mix [RMF07].

Third, we have a more complicated model of an immune system interacting with a growing tumor in Figure 3.6. The cancer cells undergo a standard birth-death-migration process. Immune cells migrate in from the barrier cells at a constant rate and destroy tumor cells on contact. This predation may produce a fibrotic cell that is weakly porous to immune cells, simultaneously blocking the spread of the cancer and the eradication of the cancer by the immune system. The formation of a protective shell of fibroblasts is not seen in the well-mixed or RDME cases due to the lack of volume exclusion. Our simulations recapture the immune-excluded response result shown in [KPS17]. This model is useful for exploring potential barriers to tumor eradication during immuno-therapy.

Lastly, we have a model of polyunsaturated fatty acid (PUFA) oxidation in lipid mem-

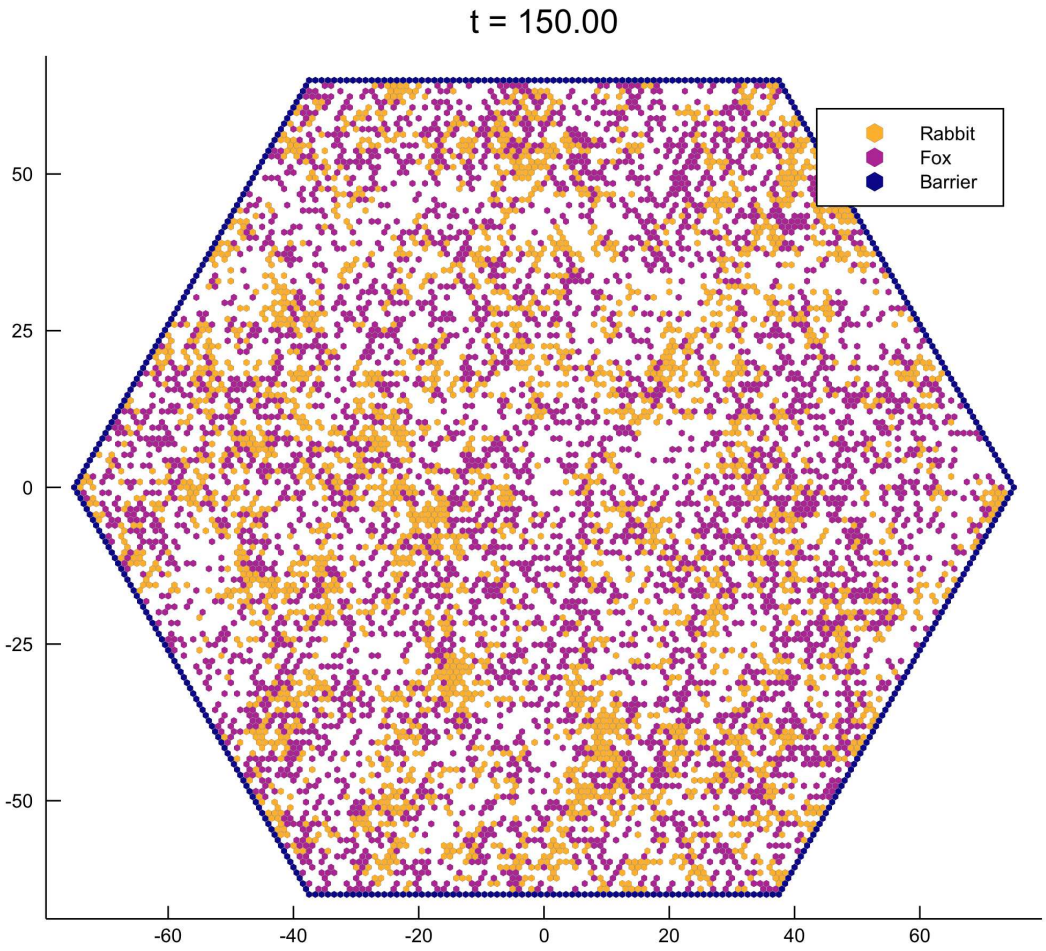

Figure 3.4: A realization of the predator-prey process. Foxes (predators) and rabbits (prey) diffuse within a bounded domain, undergoing the reactions described in Fig 1a.

branes. Certain PUFAs are susceptible to oxidation which creates a kink in their long unsaturated hydrocarbon tails. As a result, a membrane with a significant number of oxidated PUFAs loses flexibility and can lead to neurodegeneration and aging [FFB19]. Replacing the affected hydrogen atoms with deuterium significantly reduces the rate of oxidation, acting as a vaccine of sorts against the infective nature of reactive oxygen species. Figure 3.6 shows the trail of depleted (kinked) PUFAs left behind a reactive oxygen species jumping to unoxidated PUFAs. There exists a phase transition where having approximately a 20% frequency of deuteration drastically reduces the length of the depleted PUFA chains left by an oxidated species. This reduction has been observed in vitro through mortality experiments on yeast. It can be observed using our software as a consequence of the oxidated species becoming

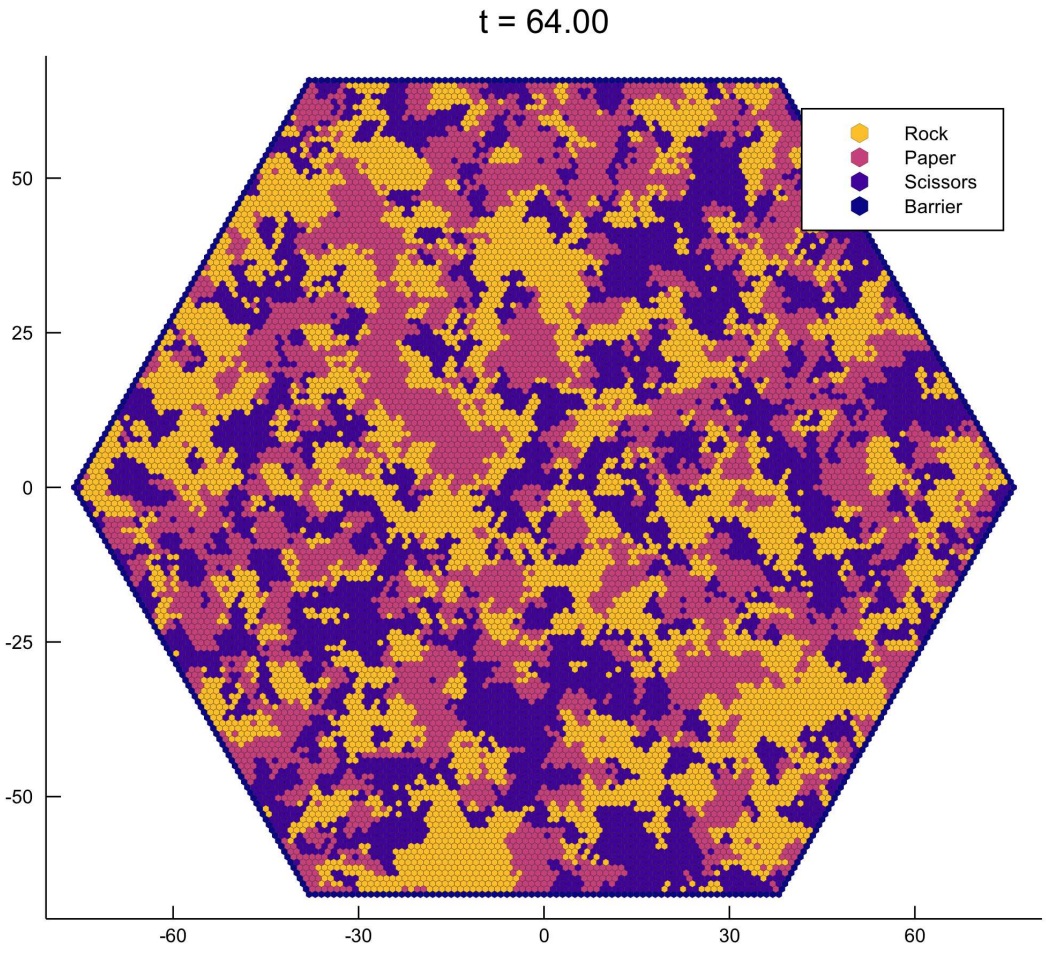

Figure 3.5: A realization of the rock-paper-scissors game. Three different species undergo a birth-death-migration process. Additionally, rock preys on scissors, scissors prey on paper, and paper preys on rock in a cyclic fashion.

trapped by its own tail and the deuterated PUFAs.

3.4 Availability and Future Directions

We have presented a principle for algorithm design that stresses elegance, performance, reproducibility, and wide applicability. These benefits can be broken down into three larger points.

First, our design allows for model standardization based on interacting particle systems.

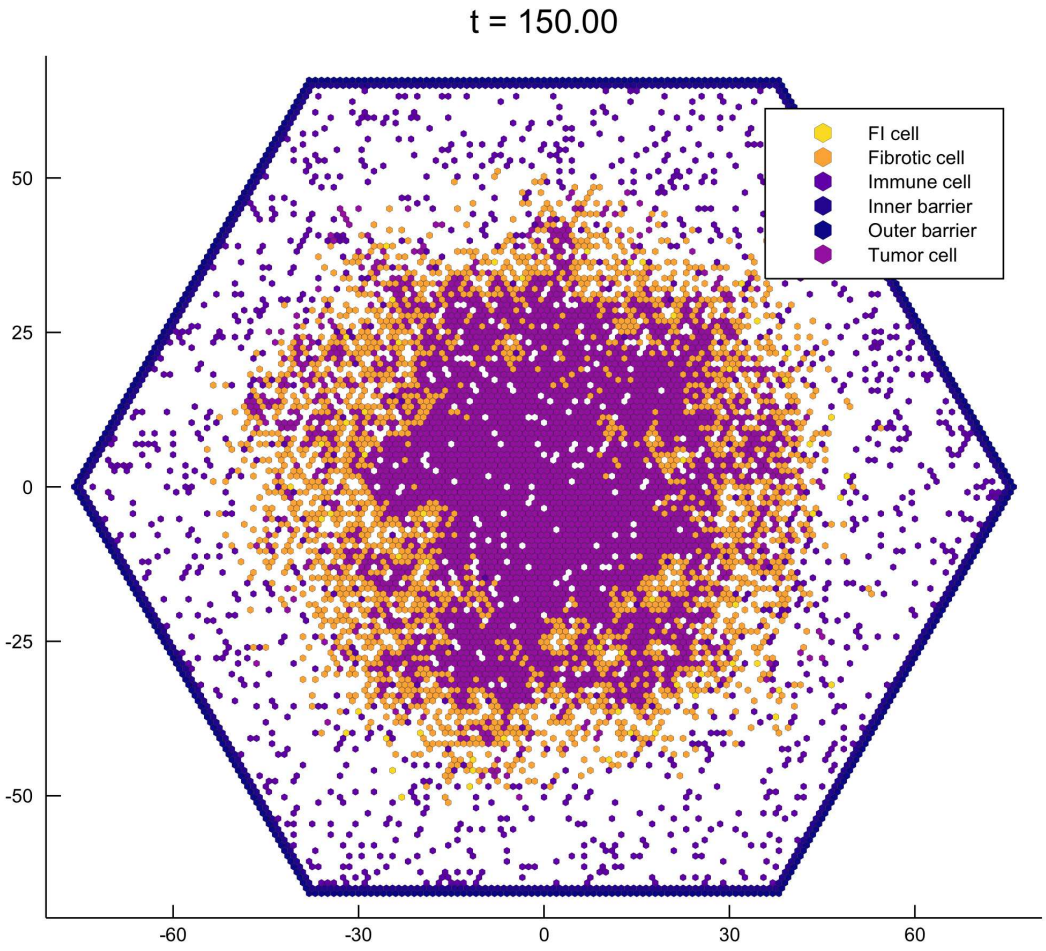

Figure 3.6: A realization of the model of immunotherapy. Tumor cells grow under a birth-death-migration process. Immune cells immigrate from the barrier at constant rate, migrate to cancer cells, then destroy the cancer cells occasionally producing fibrotic cells after this predation. Fibrotic cells are slightly porous to the immune cells but block the diffusion of the cancer cells. FI stands for a fibrotic cell that has an immune cell currently passing through it.

Many *in-silico* studies of spatial particle processes are haphazard in their construction and do not follow continuous time Markovian reaction dynamics. This limits the comparisons that can be made between models and creates barriers for new researchers looking to perform their own simulation studies. Adopting IPSs as a standard mathematical model enhances the most useful application of spatial stochastic simulation, namely generating hypotheses for given phenomena. Having a set of concrete, mechanistic rules with a straightforward probabilistic interpretation allows researchers to develop a hypothesis based off reaction dynamics that reproduce a given behavior *in-silico* and then take these dynamics back to an experimental setting for verification. The PUFA oxidation example serves as a demonstration of how hypotheses about an experimentally observed phenomenon can be tested using our software.

Second, our software is open-source and easily modifiable to individual needs. We have coded our implementation in `Julia`, a fast, expressive, and flexible open-source programming language designed for scientific computing [BEK17]. Notably `Julia` aims to solve the two-language problem where prototyping is done in a high-level language like `Python` but implementation is reserved for a fast low-level language like `C++`. `Julia` succeeds in producing high performance code using high-level programming language syntax and design, including easy parallelization. In particular, `Julia`'s ease of use readily allows for individual extensions to our software, for example genealogies, particle tracking, and potentially long-range interactions between particles. The ease of use and model standardization taken together make any research done with our software easily reproducible and straightforward to document.

Finally, our algorithm design allows us to apply arbitrary well-mixed stochastic simulation algorithms to spatial IPSs. This will be explored later in a review article that compares how each algorithm behaves in the spatial setting. Regardless, we can now apply a large swath of algorithms to spatial stochastic simulation without tedious re-implementation.

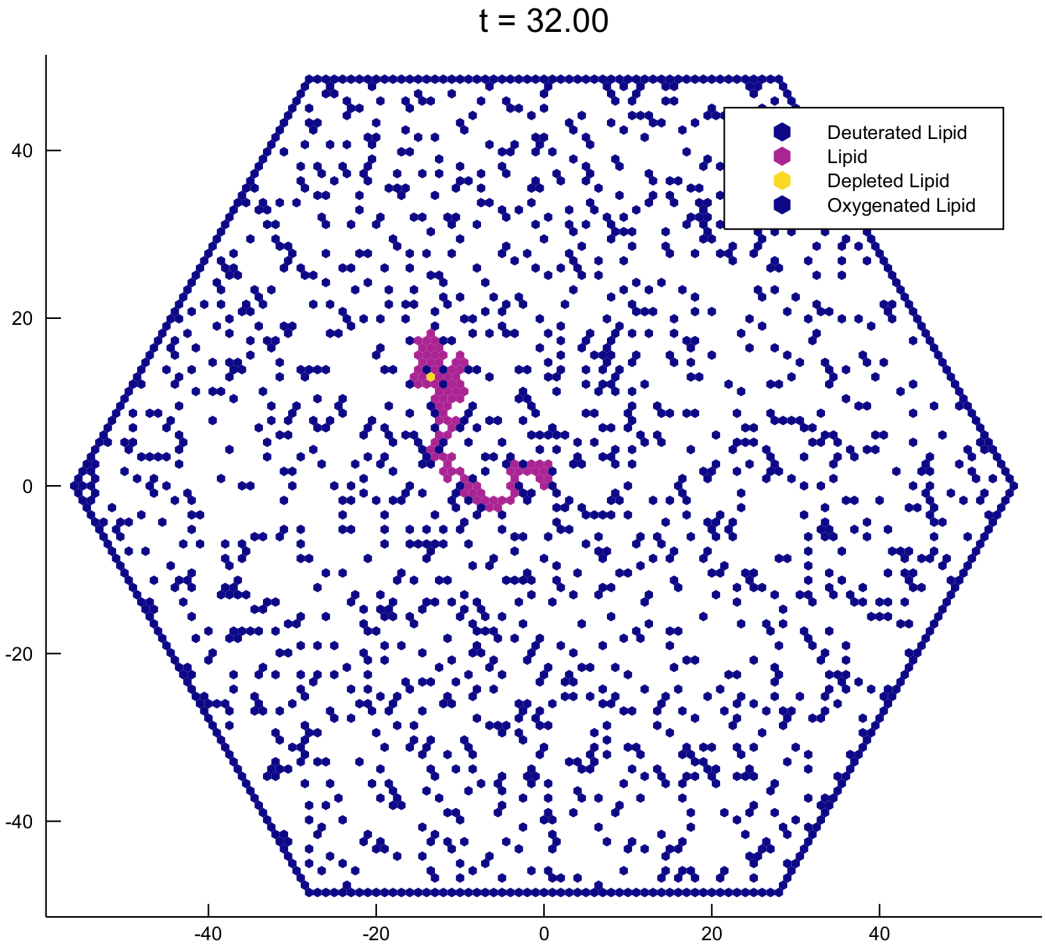

Figure 3.6: A realization of the model of lipid oxidation. Polyunsaturated fatty acids (PUFAs), denoted by open sites, are present in the lipid membranes of cells. Reactive oxygen species can oxygenate a PUFA, resulting in a depleted PUFA that reduces the flexibility of the lipid membrane. Deuterated PUFAs are resistant to the oxygenation. Adding a certain percentage of deuterated PUFAs to the membrane can drastically reduce the length of the depleted lipid chain. The simulation was initiated with a single oxygenated PUFA at the origin.

CHAPTER 4

Numerical Methods for Multi-type Branching Processes

4.1 Background

Branching processes comprise a class of stochastic process that describe how individuals reproduce, die, and transform type in the multi-type case. These processes were first studied in relation to the extinction of family names among the aristocracy by Francis Galton and the Reverend Henry Watson [Ken66]. Branching processes have since been used across a wide array of scientific disciplines, from phylogeny [BF06] to cancer biology [Dur15] and even nuclear physics [PP07].

The main mathematical tool used for studying branching processes is the probability generating function (PGF). The PGF greatly simplifies the computation of transition probabilities, the fundamental quantities that determine how a stochastic process behaves. One avenue of attack is to work with a series of infinite coupled ordinary differential equations (ODEs), frequently called the Master Equations, that govern the evolution of the transition probabilities. Instead we exploit the independence of particle lineages to reduce the computation to a small number of ODEs for the PGFs of the process originating with a single particle of each type. This use of particle independence makes the mathematical manipulations much easier by simplifying the computation of expectations greatly. Independence specifically enables the application of the pseudo-generating function technique shown later in section 4.3 upon which this chapter hinges [Bai90].

Notably our manipulations are limited to linear branching processes, meaning we assume that there are no interactions between particles. Most biological systems will be nonlinear due to resource limitations and inbuilt carrying capacities. Our computations are thus generally

restricted to the initial growth phases of the population before a sufficient density is reached at which interactions become non-negligible.

In this chapter we develop a numerical method for computing a wide range of quantities with regards to a multi-type system of branching processes. We focus primarily on computing marginalized transition probabilities, such as the probability of observing a double-mutant occur by time t, but we are also able to handle the computation of expectations and other probabilistic quantities with ease. Our derivations revolve around three core steps. We first factorize the PGF for the full process into a product of PGFs of the process beginning with a single particle of each type. Second, we solve for these PGFs using the pseudo-generating function technique which involves a simple numerical computation of a small series of coupled ODEs. Finally, we invert the PGF to obtain our desired quantities using the Fast Fourier Transform.

Previous literature has focused on the long-time limiting behavior of these branching processes [Dur15] or the emergence of a specific double-mutant [BRA13]. These series of manipulations enable us to compute transient behavior beyond the specific scenarios that have already been studied. Additionally, we aim to apply our techniques to performing inference on multi-type branching processes in a future work. This will provide an extension of previous work that has focused on using simulation for inference which requires considerable computation time [LJC17].

We finish the chapter with a numerical examination of gene therapy in hematopoietic stem cell lines. Stochastic modeling via branching processes, specifically using the methods we describe below, can quantify the probability of rare events occurring such as mutational oncogenesis and engraftment failure that may occur during stem cell transplant. We investigate multiple phenomenon involving gene therapy, including the time until two oncogenic events occur leading to leukemogenesis as well how changing the proliferative advantage of the transduced population alters the probability of observing oncogenesis. Finally, we outline how our methods may be used to study other clinical scenarios in cancer research such as inactivation of the two alleles of a tumor suppressor gene and the acquisition of multiple driver mutations.

4.2 Mathematical Notation for Branching Processes

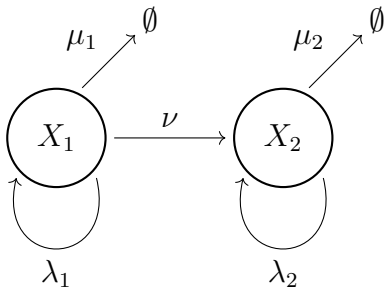

Figure 4.1: Reaction diagram for the two-type branching process. λ_i corresponds to reproduction through binary fission, μ_i corresponds to removal via death, and ν corresponds to transformation via mutation.

We begin with a linear, multi-type branching process whose counts are denoted by the vector $\mathbf{X}(t)$. Each component $X_i(t)$ denotes the number of particles of type i at time $t \geq 0$. For simplicity's sake, we shall restrict ourselves to a two-type branching process with time-homogeneous rates for the remainder of the definitions and derivations, as shown in Figure 4.1. Let $a_i(k, l)$ denote the rate at which a particle of type i produces k particles of type one and l particles of type two upon completion of its lifespan. For example, $a_1(2, 0)$ is the rate at which a single particle of type one reproduces, producing two offspring through binary fission. We define the negative total rate as

$$\alpha_i \equiv a_i(1, 0) = -\sum_{(k,l)\neq(1,0)} a_i(k, l)$$

which sets the total sum $\sum_{(k,l)} a_i(k, l) = 0$. Linearity implies that the overall rate of each event is the individual particle rate multiplied by the number of particles of the appropriate type.

As is expected from our branching process being a time-homogeneous continuous time Markov chain [Lan10], the short-time probability of jumping from only j particles of type 1 to k particles of type one and l particles of type two is given by

$$\Pr\left(\mathbf{X}(h) = (k, l) \mid \mathbf{X}(0) = (j, 0)\right) = j \times a_1(k, l) \times h + o(h) \tag{4.1}$$

as $h \downarrow 0$. Additionally, the lifespan of each particle of type i is exponentially distributed with rate $-\alpha_i$.

We define the transition probabilities

$$P_{(i,j),(k,l)}(t) \equiv \Pr\left(\mathbf{X}(t) = (k,l) \mid \mathbf{X}(0) = (i,j)\right). \tag{4.2}$$

Similarly we have the PGF

$$\phi_{ij}(t, s_1, s_2) \equiv \sum_{k=0}^{\infty}\sum_{j=0}^{\infty} P_{(i,j),(k,l)}(t) s_1^k s_2^l = E\left[s_1^{X_1(t)} s_2^{X_2(t)} \mid X_1(0) = i, X_2(0) = j\right] \tag{4.3}$$

with complex arguments s_1 and s_2. We will frequently shorten the notation for the PGF to $\phi_{ij}(t)$.

The PGF is the primary mathematical object that our derivations will focus on. Notably, it is straightforward to invert the PGF using the FFT to obtain the transition probabilities (4.2). This is not an entirely trivial numerical task; it requires $O\left(\max(k,l)^2\right)$ computations [XM15]. It is often the case that we are concerned primarily with the distribution of a single count, typically the number of particles in the mutant or second-type population. These are easy to obtain by marginalizing out the first type by plugging $s_1 = 1$ into the PGF (4.3),

$$\phi_{ij}(t, s_1, s_2)\big|_{s_1=1} = \sum_{l=0}^{\infty} P_{(i,j),(\cdot,l)}(t) s_2^l.$$

Obtaining these marginalized probabilities is much more straightforward as it requires inverting a univariate probability generating function.

4.3 Computing the Probability Generating Function

Using a technique detailed in [Bai90], we shall generate a series of Ordinary Differential Equations (ODEs) that govern the evolution of the PGF over time. We first make the observation that, by particle independence,

$$\phi_{ij}(t) = \phi_{(1,0)}(t)^i \phi_{(0,1)}(t)^j. \tag{4.4}$$

Since each clan of particles is independent of every other clan, we can view the full PGF as being composed of the individual contributions from each original ancestor particle present at time $t = 0$. For simplicity's sake we shall shorten the notation $\phi_{(1,0)} \equiv \phi_1$ and $\phi_{(0,1)} \equiv \phi_2$.

This means that it suffices to work with the PGF of the process starting with a single particle of type one and the PGF of the process starting with a single particle of type two. Consider the small-time expansion of the PGF ϕ_1 using (4.1),

$$\phi_1(t, s_1, s_2) = E\left[s_1^{X_1(t)} s_2^{X_2(t)} \mid X_1(0) = 1, X_2(0) = 0\right] = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} P_{(1,0),(k,l)}(t) s_1^k s_2^l$$
$$= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \left[\mathbf{1}_{k=1,l=0} + a_1(k,l)t + o(t)\right] s_1^k s_2^l = s_1 + t\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} a_1(k,l) s_1^k s_2^l + o(t) \tag{4.5}$$

as $t \downarrow 0$. $\mathbf{1}$ denotes the indicator function. (4.5) involve the pseudo-generating function [Bai90]

$$u_i(s_1, s_2) = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} a_i(k,l) s_1^k s_2^l. \tag{4.6}$$

We can rearrange (4.5), divide by t, then send $t \downarrow 0$ to get an expression for the derivatives of the single-particle PGFs based off the pseudo-generating functions (4.6),

$$\frac{d\phi_1(t, s_1, s_2)}{dt}\Big|_{t=0} = u_1(s_1, s_2), \qquad \frac{d\phi_2(t, s_1, s_2)}{dt}\Big|_{t=0} = u_2(s_1, s_2). \tag{4.7}$$

Now we use the Chapman-Kolmogorov equations for ϕ_1 along with (4.4) to show that

$$\begin{aligned}
\phi_1(t+h, s_1, s_2) &= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} P_{(1,0),(k,l)}(t+h) s_1^k s_2^l \\
&= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\left[\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P_{(1,0),(i,j)}(t) P_{(i,j),(k,l)}(h)\right] s_1^k s_2^l \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P_{(1,0),(i,j)}(t)\left[\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} P_{(i,j),(k,l)}(h) s_1^k s_2^l\right] \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P_{(1,0),(i,j)}(t)\phi_{ij}(h, s_1, s_2) \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P_{(1,0),(i,j)}(t)\phi_1(h, s_1, s_2)^i \phi_2(h, s_1, s_2)^j \\
&= \phi_1\big(t, \phi_1(h, s_1, s_2), \phi_2(h, s_1, s_2)\big).
\end{aligned}$$

The relationship is time-symmetric, therefore we also have

$$\phi_1(t+h, s_1, s_2) = \phi_1\big(h, \phi_1(t, s_1, s_2), \phi_2(t, s_1, s_2)\big). \tag{4.8}$$

Naturally the same relationships exist for ϕ_2.

We now derive the backward Kolmogorov equations for ϕ_1 and ϕ_2. These are obtained by Taylor expanding ϕ_1 around t and utilizing (4.7) and(4.8),

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

The same arguments produce an analogous expression for ϕ_2.

Performing the previous rearrangement, dividing by h, and sending h to zero gives us our desired system of backwards Kolmogorov equations with initial conditions,

$$\begin{aligned}\frac{d}{dt}\phi_1(t) &= u_1(\phi_1(t),\phi_2(t)), \qquad \phi_1(0) = s_1 \\ \frac{d}{dt}\phi_2(t) &= u_2(\phi_1(t),\phi_2(t)), \qquad \phi_2(0) = s_2.\end{aligned}$$

The initial conditions follow from the definition of the PGF (4.3).

We will now focus on the specific rates $a_i(k,l)$ that are used in the two-type branching process, listed below:

$$\begin{gathered}a_1(2,0) = \lambda_1, \qquad a_1(0,0) = \mu_1, \qquad a_1(0,1) = \nu, \qquad a_1(1,0) = -(\lambda_1+\mu_1+\nu) \\ a_2(0,2) = \lambda_2, \qquad a_2(0,0) = \mu_2, \qquad a_2(0,1) = -(\lambda_2+\mu_2).\end{gathered}$$

We plug these into the pseudo-generating functions to get

$$\begin{aligned}u_1(s_1,s_2) &= \lambda_1 s_1^2 + \nu s_2 - (\lambda_1+\mu_1+\nu)s_1 + \mu_1 \\ u_2(s_1,s_2) &= \lambda_2 s_2^2 - (\lambda_2+\mu_2)s_2 + \mu_2.\end{aligned}$$

We use these expressions in the differential equations for the single-particle generating functions to get our desired backwards equations,

$$\frac{d}{dt}\phi_1(t) = \lambda_1\phi_1^2 + \nu\phi_2 - (\lambda_1+\mu_1+\nu)\phi_1 + \mu_1, \qquad \phi_1(0) = s_1 \tag{4.9}$$

$$\frac{d}{dt}\phi_2(t) = \lambda_2\phi_2^2 - (\lambda_2+\mu_2)\phi_2 + \mu_2, \qquad \phi_2(0) = s_2. \tag{4.10}$$

4.3.1 Solving for ϕ_2

The differential equation (4.10) is a nonlinear first order Riccati equation, which is fortunately straightforward to solve. This is because the ansatz $\phi_2 = K$ gives us the constant particular solutions $\phi_2 = 1$ and $\phi_2 = \mu_2/\lambda_2$. As is standard for solving a Riccati equation, we make the substitution $z = \frac{1}{\phi_2 - 1}$, implying that $\phi_2 = 1 + 1/z$. Plugging this substitution into (4.10) gives

$$\begin{aligned}
\phi_2' = -\frac{z'}{z^2} &= \mu_2 - (\lambda_2 + \mu_2)\left(1 + \frac{1}{z}\right) + \lambda_2\left(1 + \frac{1}{z}\right)^2 \\
&= \mu_2 - (\lambda_2 + \mu_2) - \frac{\lambda_2 + \mu_2}{z} + \lambda_2\left(\frac{1}{z^2} + \frac{2}{z} + 1\right) \\
&= \frac{\lambda_2 - \mu_2}{z} + \frac{\lambda_2}{z^2}.
\end{aligned}$$

Multiplying through by $-z^2$ and rearranging gives a linear first order differential that is straightforward to solve using an integrating factor,

$$\begin{aligned}
z' &= (\mu_2 - \lambda_2)z - \lambda_2 \\
z &= \frac{\lambda_2}{\mu_2 - \lambda_2} + Ce^{(\mu_2 - \lambda_2)t}
\end{aligned}$$

for some constant C that depends on the initial conditions. We plug z back into $\phi_2 = 1 + 1/z$ and include the initial conditions $\phi_2(0) = s_2$ to get the solution

$$\phi_2(t, s_2) = 1 + \left[\frac{\lambda_2}{\mu_2 - \lambda_2} + \left(\frac{1}{s_2 - 1} + \frac{\lambda_2}{\lambda_2 - \mu_2}\right)e^{(\mu_2 - \lambda_2)t}\right]^{-1}. \tag{4.11}$$

Note that in the critical case $\lambda_2 = \mu_2$, we get the following solution

$$\phi_2(t, s_2) = 1 + \left[\frac{1}{s_2 - 1} - \lambda_2 t\right]^{-1}$$

4.3.2 Solving for ϕ_1

The differential equation (4.9) governing ϕ_1 is substantially more difficult due to the presence of a mutation term involving ϕ_2. We now have an inhomogeneous nonlinear Riccati equation that cannot be solved by a simple substitution trick. In the Future Directions chapter, we will present the beginnings of an asymptotic analysis that solves for ϕ_1 using the fact that

the mutation rate ν is much much smaller than the other rates of the process. For now, it suffices to note that it is straightforward to solve ϕ_1 numerically using any number of differential equation solvers. We have chosen the `DifferentialEquations.jl` package in the `Julia` programming language [RN17, BEK17].

4.3.3 Inverting the Numerical PGF

Once we have solved for ϕ_1 and ϕ_2 to a specified time point, we can compute the full PGF ϕ_{ij} via (4.4). We then use a technique from [Lan82] to invert the PGF to obtain our desired marginalized univariate probability function. We begin with a change of variables $s = e^{2\pi i w}$, placing our PGF argument s on the unit circle in the complex plane. This transforms the generating function into a periodic function,

$$\phi_{ij}\left(t, e^{2\pi i w}\right) = \sum_{k=0}^{\infty} c_k(t) e^{2\pi i w k}$$

where $c_k(t)$ are the coefficients that contain our desired transition probabilities and $i = \sqrt{-1}$. c_k is also the kth coefficient of a Fourier series, and therefore can be obtained by a straightforward inversion

$$c_k(t) = \int_0^1 \phi_{ij}\left(t, e^{2\pi i w}\right) e^{-2\pi i w k} dw.$$

We can now approximate the integration through a Riemann sum for large N,

$$c_k(t) \equiv P_{(ij),k}(t) \approx \frac{1}{N-1} \sum_{l=0}^{N-1} \phi_{ij}\left(t, e^{2\pi i l/N}\right) e^{-2\pi i l k/N}.$$

Note that we can increase the accuracy of the approximation by increasing the size of the gridlength, i.e. make N large. Most importantly, the FFT allows us to capture all the coefficients for $k = 0$ to $N-1$ in a single computation. This makes obtaining the marginalized transition probabilities very straightforward.

4.3.4 Computing Relevant Probabilities and Expectations

Once we've obtained our numerical solution for ϕ_{ij}, obtaining expectations is trivial through numerical differentiation,

$$E\left[X_i(t) \mid \mathbf{X}(0) = (m,n)\right] = \frac{\partial}{\partial s_i}\phi_{(m,n)}(t,s_1,s_2)\big|_{s_1=s_2=1}.$$

Higher order moments can be obtained in a similar fashion, see [Lan10].

We can also obtain information about thresholds, for example the probability that a specific population has passed a certain number. These probabilities can be obtained by marginalizing out one population and summing the probability density of the other after the threshold,

$$\Pr\left(X_2(t) \geq M \mid \mathbf{X}(0) = (i,j)\right) = \sum_{k=0}^{\infty}\sum_{l=M}^{\infty} P_{(i,j),(k,l)}(t).$$

For populations with more than two types, we can marginalize over multiple species by plugging $s_i = 1$ into the joint PGF.

However, we would prefer to marginalize out all but one of the populations to take advantage of our FFT inversion technique. Specifically marginalizing over X_1 can be done via

$$\phi_{ij}(t,s_1,s_2)\big|_{s_1=1} = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} P_{(i,j),(k,l)}(t)s_2^l = \sum_{l=0}^{\infty} P_{(i,j),(\cdot,l)}(t)s_2^l.$$

This gives the marginalized PGF for X_2 which is notably univariate in s_2. Therefore we can use the previous numerical approach to invert the PGF and obtain the marginal transition probabilities $P_{(i,j),(\cdot,l)}(t)$ for $X_2(t)$. We choose a suitable truncation level for inverting our PGF such that our marginal transition probabilities capture the probability mass within a suitable level of error. Then we can compute the desired threshold probabilities as

$$\Pr\left(X_2(t) \geq M \mid \mathbf{X}(0) = (i,j)\right) = 1 - \sum_{l=0}^{M-1} P_{(i,j),(\cdot,l)}(t).$$

These manipulations will be used extensively in the next section.

4.4 Numerical Results

We will now investigate the three-type branching process as a model for diploid mutations in a growing population of hematopoietic stem cells. This applies to a growing population of transplanted stem cells in gene therapy, particularly to the acquisition of mutations that lead to oncogenesis. The rate parameters are listed in Table 4.1 and the reactions are depicted in Figure 4.2.

Rate Symbol	Diagram	Parameter Value
λ_1	$X_1 \to X_1 + X_1$	2.4×10^{-2}
μ_1	$X_1 \to \emptyset$	1.4×10^{-2}
ν_1	$X_1 \to X_2$	$\lambda_1 \times 10^{-8}$
λ_2	$X_2 \to X_2 + X_2$	Varies
μ_2	$X_2 \to \emptyset$	1.4×10^{-2}
ν_2	$X_2 \to X_3$	$\lambda_2 \times 10^{-8}$
λ_3	$X_3 \to X_3 + X_3$	Varies
μ_3	$X_3 \to \emptyset$	1.4×10^{-2}

Table 4.1: Reactions and parameters for the three-type branching process. The parameters have units of 1/week and were sourced from [ACM02, PDV10, PZF17, MHI05].

Again we abbreviate $\phi_1 = \phi_{(1,0,0)}$, $\phi_2 = \phi_{(0,1,0)}$, and $\phi_3 = \phi_{(0,0,1)}$. The differential equations that govern the single-particle generating functions are as follows:

$$\begin{aligned}
\frac{d}{dt}\phi_1 &= \lambda_1\phi_1^2 - (\lambda_1 - \mu_1 - \nu_1)\,\phi_1 + \nu_1\phi_2 + \mu_1 \\
\frac{d}{dt}\phi_2 &= \lambda_2\phi_2^2 - (\lambda_2 - \mu_2 - \nu_2)\,\phi_2 + \nu_2\phi_3 + \mu_2 \\
\frac{d}{dt}\phi_3 &= \lambda_3\phi_3^2 - (\lambda_3 - \mu_3)\,\phi_3 + \mu_3.
\end{aligned}$$

Note that $\frac{d}{dt}\phi_3$ has the same structure as $\frac{d}{dt}\phi_2$ from the two-type case, (4.10). We can use

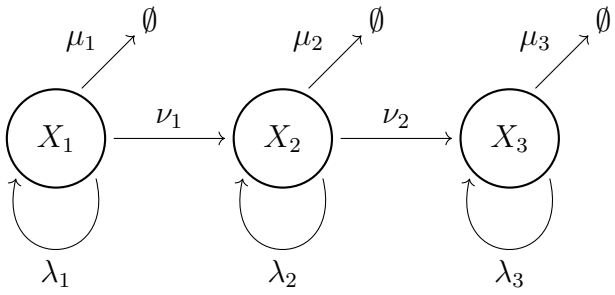

Figure 4.2: Reaction diagram for the three-type branching process. λ_i corresponds to reproduction through binary fission, μ_i corresponds to removal via death, and ν_i corresponds to transformation via mutation.

the same argument from section 4.3.1 to show that

$$\phi_3(t, s_3) = 1 + \left[\frac{\lambda_3}{\mu_3 - \lambda_3} + \left(\frac{1}{s_3 - 1} + \frac{\lambda_3}{\lambda_3 - \mu_3} \right) e^{(\mu_3 - \lambda_3)t} \right]^{-1}.$$

We begin by comparing how varying the initial counts of the wild-type and single-mutants affects the probability of observing at least one double-mutant one year after transplant. Figure 4.3 depicts these probabilities and draws the expected conclusion that increasing the number of single-mutants over wild-type particles initially present has a larger effect on seeing a double-mutant occur. This has clinical relevance as it places a bound on how mutagenic the lentiviral gene addition can be before deleterious effects can be expected. If the mutagenesis is rare, producing only one to ten single-mutants in the transplanted population, then there is a very small risk that a deleterious second mutation event will occur within a year. Additionally, the number of transplanted wild-type stem cells can be quite large before mutational events become likely, which bodes well for transplanting large populations of stem cells.

Next we compare varying the birth rates across the population of wild-type stem cells and single-mutants. Figure 4.4 assumes no initial single-mutants, and so the birth rate of the wild-type population X_1 is rate-limiting on the probability of observing a double mutant. Note that changing the wild-type birth rate by a factor of ten greatly increases the odds of observing a double-mutant, while changing the single-mutant birth rate has a much decreased

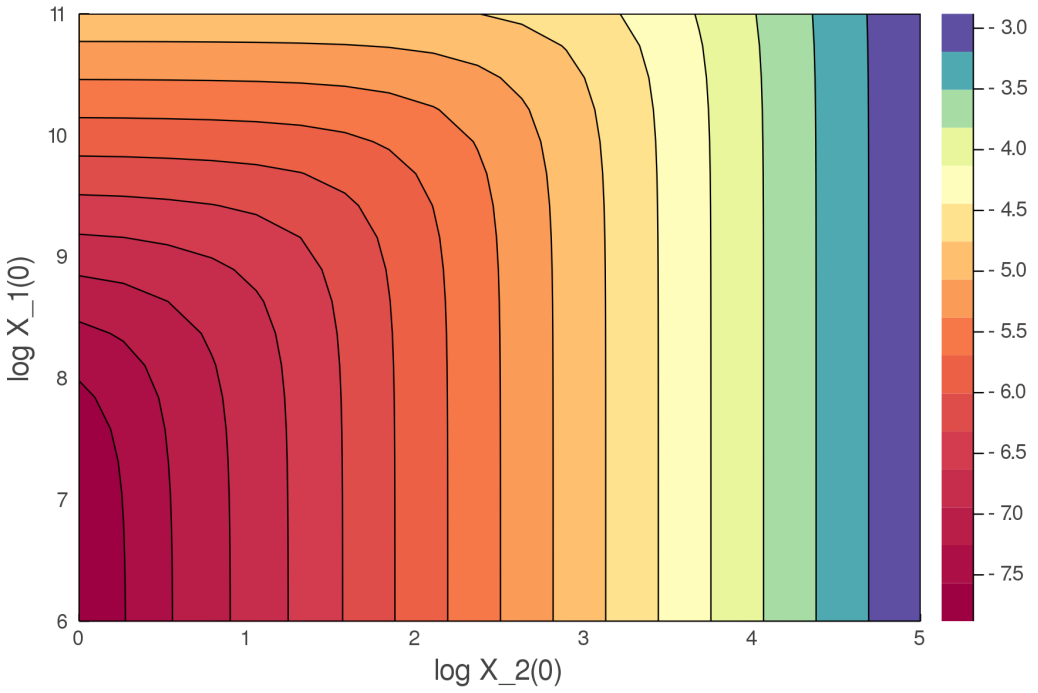

Figure 4.3: Probability of observing at least one double-mutant at $t = 52$ weeks while varying the initial counts of X_1 and X_2. The probability is depicted in log scale.

effect. Therefore it is relatively safe for the single-mutations to be quite proliferative if the lentiviral gene addition avoids creating any single-mutants in the original insertion process.

Since we are solving the ODEs for the single-particle PGFs forward in time, we can examine how the probability of observing a double-mutant changes over time based off of varying initial population counts. There are three phases of growth corresponding to a fast initial phase, followed by an exponential growth in the probability, and finally tapering out towards one as a double mutant becomes guaranteed to be observed.

Finally, we note that there are significantly more probabilistic quantities that we can study using our numerical techniques. There also are a wide variety of problems in cancer modeling that can also be addressed. We will leave exploring these questions to the final chapter on future directions.

In conclusion, we have numerically computed the PGF ϕ_{ijk} for a three-type branching process. We've then marginalized the PGF to obtain a univariate PGF for the X_3 double-mutant

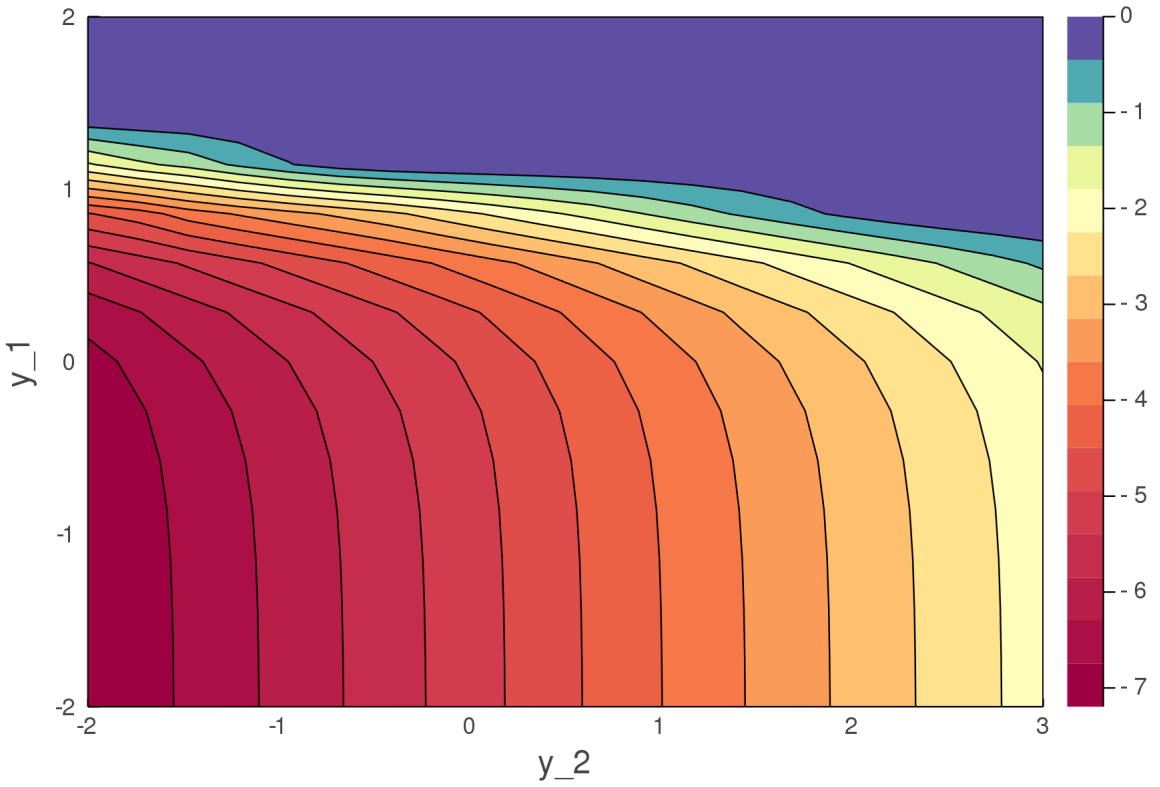

Figure 4.4: Probability of observing at least one double mutant at $t = 52$ weeks while varying the birth rates λ_i of X_1 and X_2. The probability is depicted in log scale. We have $\lambda_i = 0.024 \times 10^{y_i}$, e.g. when $y_i = 0$, $\lambda_i = 0.024$. The initial counts are $X_1(0) = 1000$ and $X_2(0) = 0$.

population alone. This allows us to perform a fast, straightforward numerical inversion of the univariate PGF using the FFT to obtain probabilistic information about the probabilities of leukemogenesis after gene therapy. This presents clinically relevant information about how leukemia may form under a variety of conditions including mutagenesis from the initial gene insertion as well as differing proliferative rates among the mutant and wild-type populations.

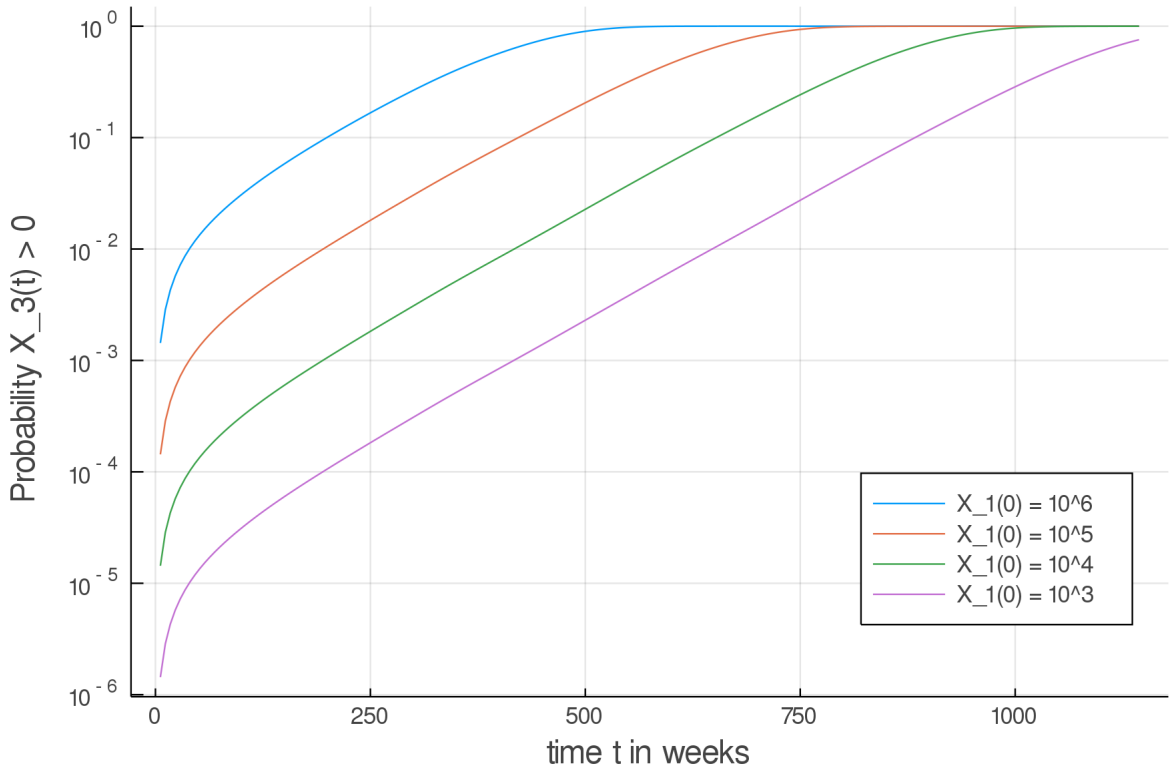

Figure 4.5: Probability of observing at least one double mutant over time while varying the initial counts of X_1 and X_2. The probability is depicted in log scale.

CHAPTER 5

Future Directions

5.1 Extensions of Dissertation Work

Each chapter has significant future research potential. The work on truncating birth-death processes (BDPs) from chapter two can be extended to bivariate BDPs. This in turn enables inference on more complex processes, such as the susceptible-infected-recovered model from epidemiology.

There are substantial advancements to be made to the stochastic simulation algorithms for Interacting Particle Systems (IPSs) from chapter three. First we aim to release a review article detailing the differences in performance and accuracy between different well-mixed stochastic simulation algorithms applied to spatial IPSs. Second we can incorporate both longer-distance interactions between particles as well as gradient fields of nutrients, oxygen, and other concentrated species that overlay our grid and influence the reaction rates of the individual particles. This would be highly useful for studying how hypoxia influences tumor structure and development, particularly with regards to vascularization present in the surrounding tissue. In addition to software improvements and algorithmic comparisons, we aim to extend the IPS simulation to study problems involving immunotherapy in growing cancers. We intend to model the tumor microenvironment including fibroblasts, immune cells, cancer cells, and specific myeloid-derived suppressor cells that reduce immune activity in response to cancer. This project will also involve additional algorithm design to incorporate chemotaxis wherein immune cells track down cancer cells for destruction. This is non-trivial and involves biasing the direction of diffusion during the update step.

Finally, the fourth chapter has two significant lines of expansion. The most obvious

and scientifically relevant involves extending the previously described work on multi-type branching processes to consider questions of parameter and model inference. The general idea is very similar to the Method of Moments from statistics wherein we average over observed data to find some sort of sample expectation that we can reproduce using our model. Means and variances are the most obvious and straightforward. Second, we choose a model design and search its associated parameter space to minimize the squared absolute distance between the sampled quantity and the expectation obtained by our model. Since our computations are quite fast and can incorporate multiple time points with the same parameters, like shown in Figure 4.5, the parameter search would not be computationally burdensome.

This has applications to inferring the structure of stem-cell hierarchies in barcoded cancers [LJC17]. Our methodology has significant advantages over previous approaches, notably that we can alter the structure of the underlying mathematical process without significantly changing the numerical processes used to obtain the transition probabilities. For example, both the shift process $X_1 \to X_2$ and the mutational birth process $X_1 \to X_1 + X_2$ can be incorporated into our system of equations. These alterations will change the structure of the underlying differential equations to be solved, but they do not change the overall numerical manipulations required to solve for and invert the probability generating function. To be specific, our approach can handle any multitype branching process so long as each cell lineage behaves independently. This is required by the step in the derivation where the probability generating function $\phi_{i,j}$ can be factored into $\phi_1^i \phi_2^j$ via independence.

In addition to the numerical applications detailed in chapter four, there are potential analytical approaches involving asymptotics that we detail in the following section.

5.2 Asymptotic Analysis of Multi-Type Branching Processes

This builds upon the mathematical framework presented in chapter four. For the sake of the derivations, we will again focus primarily on the two-type process. The aim, albeit unfinished, of this section is to derive an asymptotic expansion for the probability generating

function (PGF) (4.3). This is achieved by deriving another asymptotic expansion for the single-particle generating function ϕ_1 under the conditions that the mutation rate ν is much much smaller than the other parameters in the process. From this we would like to expand the analytic expression for the PGF in powers of s_1 and s_2 to obtain an asymptotic expansion for the transition probabilities. Similar to section 4.3.4, we can also marginalize out one of the species to greatly simplify the computation.

This work is intended to be submitted as a separate paper from Chapter four. A non-trivial amount of work remains, particularly obtaining the coefficients that are the transition probabilities from the asymptotic expression for the PGF.

5.2.1 Initial Equations

We begin with the differential equations for the individual probability generating functions for the two-type case $X_1(t)$ and $X_2(t)$, starting from a single particle of each type. We focus on the reaction diagram in Figure 5.1.

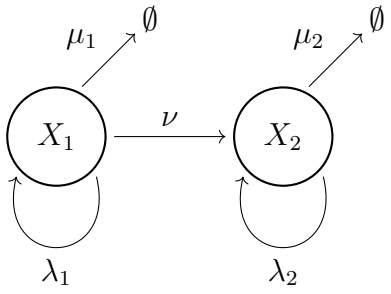

Figure 5.1: Reaction diagram for the two-type branching process.

The PGF for a single particle of type X_2 obeys the ODE

$$\phi_2' - \lambda_2 \phi_2^2 + (\lambda_2 + \mu_2)\,\phi_2 = 0. \tag{5.1}$$

Chapter 4 contains the details for solving this equation,

$$\phi_2(t, s_2) = 1 + \left[\frac{\lambda_2}{\mu_2 - \lambda_2} + \left(\frac{1}{s_2 - 1} + \frac{\lambda_2}{\lambda_2 - \mu_2} \right) e^{(\mu_2 - \lambda_2)t} \right]^{-1}. \tag{5.2}$$

This is coupled with the differential equation for the PGF for a single particle of type X_1,

$$\phi_1' - \lambda_1 \phi_1^2 + (\lambda_1 + \mu_1 + \nu)\,\phi_1 = \mu_1 + \nu\phi_2. \tag{5.3}$$

Note that (5.3) contains factors of s_2 through ϕ_2. This means that ϕ_1 will contribute to the probability of seeing particles of type X_2, as expected from the reaction diagram.

ϕ_1 and ϕ_2 combine to give the full PGF of the process with i initial X_1 particles and j initial X_2 particles,

$$\begin{aligned}\phi_{i,j}(t) &= \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} s_1^n s_2^m \Pr\left(X_1(t) = n, X_2(t) = m \mid X_1(0) = i, X_2(0) = j\right) \\ &= \phi_1(t)^i \phi_2(t)^j. \end{aligned} \tag{5.4}$$

This follows from the independence of the clans produced from each initial particle.

(5.3) is a Riccati equation. There are a series of manipulations we can perform to transform the governing equation from being a nonlinear inhomogeneous first order ODE into a linear second order linear ODE. These manipulations are standard for Riccati equations [Inc26].

We begin with (5.3) and the substitution $v = \lambda_1\phi_1$. This results in a simpler Riccati equation

$$v' = v^2 - (\lambda_1 + \mu_1 + \nu)\,v + \lambda_1\,(\mu_1 + \nu\phi_2)\,. \tag{5.5}$$

We make another substitution $v = -u'/u$ to get

$$v' = -\left(\frac{u'}{u}\right)' = -\frac{u''}{u} + \left(\frac{u'}{u}\right)^2 = -\frac{u''}{u} + v^2.$$

Rearrange and use (5.5) to get

$$\begin{aligned}\frac{u''}{u} &= v^2 - v' = (\lambda_1 + \mu_1 + \nu)\,v - \lambda_1\,(\mu_1 + \nu\phi_2) \\ &= -(\lambda_1 + \mu_1 + \nu)\,\frac{u'}{u} - \lambda_1\,(\mu_1 + \nu\phi_2)\,.\end{aligned}$$

We multiply through by u and rearrange to get the final equation for u

$$u'' + (\lambda_1 + \mu_1 + \nu)\,u' + \lambda_1\,(\mu_1 + \nu\phi_2)\,u = 0 \tag{5.6}$$

where for clarity $\phi_1 = -u'/\lambda_1 u$. We also have the initial conditions

$$u(0) = 1 \qquad \text{and} \qquad u'(0) = -\lambda_1 s_1. \tag{5.7}$$

We will be working primarily with (5.6) as it is substantially simpler than (5.3).

5.2.2 Asymptotic expansion for u when $\nu \ll 1$

The following depends on the mutation rate ν being quite small compared to the other parameters. Under typical cell dynamics, it is of order $10^{-8} \times \lambda_1$. This makes ν an excellent candidate for forming an asymptotic expansion for u. Specifically we expand u in powers of ν,

$$u \sim u_0 + \nu u_1 + \nu^2 u_2 + ... \tag{5.8}$$

where $\sim$ denotes asymptotic equivalence as $\nu \to 0$.

5.2.3 Applications to expanding ϕ_1^i

Some preliminary facts about asymptotic expansions are in order. First, we will assume $u' \sim u_0' + \nu u_1' + ...$, which is not always true. Fortunately for the parameter values we care about, it does appear to be the case. See Figure 5.2 for a comparison of u' with the derivative of the first term in its expansion u_0'. Even the first term in the expansion is incredibly accurate for biologically relevant values of ν.

Second we note that if $f \sim f_0 + \nu f_1 + ...$ and $g \sim g_0 + \nu g_1 + ...$, then under mild conditions on the coefficients of the expansions ($g \neq 0$ and $g_n \neq 0$ for sufficiently large n) we have $f/g \sim (f_0 + \nu f_1 + ...) / (g_0 + \nu g_1 + ...)$. These two facts together can be used to simplify ϕ_1^i in (5.4). Using $\phi_1 = -u'/\lambda_1 u$ and keeping terms up to $O(\nu)$,

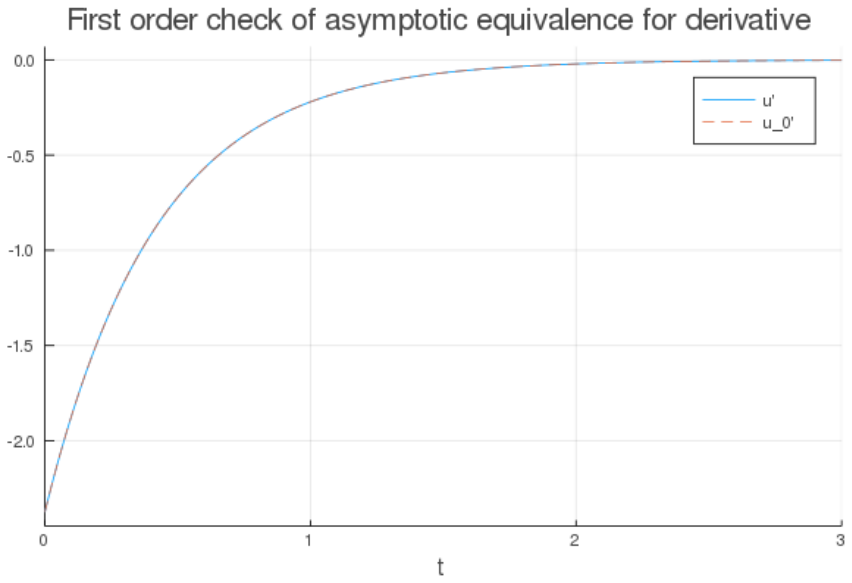

Figure 5.2: Comparisons of the numerical solution for u' from (5.6) against the derivative of the first term in the asymptotic expansion u_0'. An explicit formula for u_0' is given by the derivative of (5.11). We set $\nu = 10^{-8} \times \lambda_1$, so ν is significantly less than 1. The full parameters are given in Table 4.1.

$$\begin{aligned}
\phi_1^i &\sim \left(-\frac{u_0' + \nu u_1' + O(\nu^2)}{\lambda_1 \left(u_0 + \nu u_1 + O(\nu^2) \right)} \right)^i = \left(-\frac{1}{\lambda_1} \right)^i \frac{\left[u_0' + \nu u_1' + O(\nu^2) \right]^i}{\left[u_0 + \nu u_1 + O(\nu^2) \right]^i} \\
&= \left(-\frac{1}{\lambda_1} \right)^i \left(\frac{(u_0')^i + \nu i (u_0')^{i-1} u_1' + O(\nu^2)}{u_0^i + \nu i u_0^{i-1} u_1 + O(\nu^2)} \right) \\
&= \left(-\frac{u_0'}{\lambda_1 u_0} \right)^i \left(\frac{1 + \nu i u_1'/u_0' + O(\nu^2)}{1 + \nu i u_1/u_0 + O(\nu^2)} \right).
\end{aligned}$$

This is essentially the binomial approximation $(1+x)^\alpha \sim 1 + \alpha x$ for $|\alpha x| \ll 1$. Given that i can be truly enormous, this may prove to be a problem. This is a point where we could potentially have to use higher order terms such that x is of $O(\nu^2)$, which is a truly tiny number and can probably combat how large i can get. We shall see.

We now use the geometric expansion for small x, $\frac{1}{1+x} = 1 - x + x^2 - ...$ to expand the

denominator in powers of ν,

$$\begin{aligned}\phi_1^i &\sim \left(-\frac{u_0'}{\lambda_1 u_0}\right)^i \left(1+\nu i\frac{u_1'}{u_0'}+O(\nu^2)\right)\left(1-\nu i\frac{u_1}{u_0}+O(\nu^2)\right) \\ &= \left(-\frac{u_0'}{\lambda_1 u_0}\right)^i \left[1+\nu i\left(\frac{u_1'}{u_0'}-\frac{u_1}{u_0}\right)+O(\nu^2)\right] \end{aligned} \tag{5.9}$$

Note that every term in $O(\nu^2)$ will have incredibly small contributions to the transition probabilities once the PGF is either numerically inverted or expanded in a power series of s_1 and s_2. It would of course be possible to find terms that are multiplied by ν^2 to continue the asymptotic expansion; this is a straightforward application of the binomial theorem. I don't particularly see the point given how incredibly small ν^2 is, but it's doable.

We also note that the first term $-u_0'/\lambda_1 u_0$ can be replaced with $\tilde{\phi}_1$, which is the same equation (5.2) used for ϕ_2 with λ_2 and μ_2 swapped for λ_1 and μ_1. This is skipping ahead a bit; the section on deriving u_0 makes this connection more clear. Importantly, this shows that the dominant $O(1)$ contributions to ϕ_1^i are given by $\tilde{\phi}_1^i$, which is exactly the behavior of the branching process without mutation. This also mirrors our intuition that the probability contribution from the X_1 population to the X_2 population scales with the shift rate ν.

5.2.4 First term u_0

We begin with the simplest term u_0. This is obtained by plugging the expansion (5.8) into (5.6) and collecting terms that are of $O(1)$ (not multiplied by a factor of ν),

$$u_0'' + (\lambda_1+\mu_1)\,u_0' + \lambda_1\mu_1 u_0 = 0. \tag{5.10}$$

Using characteristic polynomials we find that

$$u_0(t) = c_1 e^{-\lambda_1 t} + c_2 e^{-\mu_1 t}$$

for constants c_1 and c_2 that satisfy BCs (5.7) such that

$$u_0(0) = 1 = c_1 + c_2$$

$$u_0'(0) = -\lambda_1 s_1 = -\lambda_1 c_1 - \mu_1 c_2$$

After some algebra we find that

$$u_0(t) = (1 - C_1)\, e^{-\lambda_1 t} + C_1 e^{-\mu_1 t} \tag{5.11}$$

where

$$C_1 = \frac{1 - s_1}{1 - \mu_1/\lambda_1}. \tag{5.12}$$

We make an aside regarding the second population X_2 and its single-particle PGF ϕ_2. Think of what would happen to (5.6) if we set $\nu = 0$ and swapped the parameters λ_1 and μ_1 for λ_2 and μ_2. We would get (5.10). Explicitly, if $\tilde{u} = u_0$ with its parameters swapped, then $\phi_2 = -\tilde{u}'/\lambda_2\tilde{u}$. This gives the observation about the $O(1)$ behavior of ϕ_1^i noted in the previous section for (5.9).

5.2.5 Second term u_1

We return to the expansion (5.8) plugged into (5.6), this time keeping terms that are $O(\nu)$,

$$\nu u_1'' + (\lambda_1 + \mu_1)\, \nu u_1' + \nu u_0' + \lambda_1\mu_1\nu u_1 + \nu\phi_2 u_0 = 0.$$

This gives the following equation for u_1 in terms of ϕ_2 and u_0,

$$u_1'' + (\lambda_1 + \mu_1)\, u_1' + \lambda_1\mu_1 u_1 = -u_0' - \lambda_1\phi_2 u_0. \tag{5.13}$$

Since the BCs (5.7) have been satisfied by u_0, we get the following BCs for u_1,

$$u_1(0) = 0 \qquad \text{and} \qquad u_1'(0) = 0. \tag{5.14}$$

(5.13) is a more complicated inhomogeneous ODE. However, it can be readily solved in terms of a Green's function. See Bender and Orszag's (B&O) "Advanced Mathematical Methods" section 1.5 for a general description of Green's functions for second order linear ODEs [BO13].

To begin, let $y_1(t) = e^{-\lambda_1 t}$ and $y_2(t) = e^{-\mu_1 t}$. These functions are the general solutions to the homogeneous problem

$$y'' + (\lambda_1 + \mu_1)\, y' + \lambda_1\mu_1 y = 0$$

as seen in the previous discussion about u_0. Given these general solutions, if we can find a particular solution u_p to equation (5.13), then the full solution is given by

$$u_1(t) = c_1 y_1(t) + c_2 y_2(t) + u_p(t).$$

This follows from a standard theorem on inhomogeneous ODEs. We will choose $u_p(0) = u_p'(0) = 0$ in a minute. This means that c_1 and c_2 are chosen such that the BCs (5.14) are satisfied, which conveniently sets $c_1 = c_2 = 0$. Therefore $u_1(t) = u_p(t)$.

The particular solution u_p to (5.13) is given by the integral of the Green's function G times the inhomogeneous portion,

$$u_p(t) = \int_0^\infty G(t,\tau)\left[-u_0'(\tau) - \lambda_1\phi_2(\tau)u_0(\tau)\right] d\tau. \tag{5.15}$$

$G(t,\tau)$ is given by

$$\frac{d^2}{dt^2}G + (\lambda_1 + \mu_1)\frac{d}{dt}G + \lambda_1\mu_1 G = \delta(t-\tau) \tag{5.16}$$

where $\delta(t-\tau)$ is the Dirac delta function. G also comes with the constraints of continuity at the point $t = \tau$ and that the derivative jumps by 1 over the point $t = \tau$, standard practice for Green's functions. We also choose the BCs $G(t=0,\tau) = \frac{d}{dt}G(t=0,\tau) = 0$, which is how we guarantee that $u_p(0) = u_p'(0) = 0$.

We will skip over the derivation of G, again see [BO13] section 1.5 for details. It is ultimately given as a function of the general solutions y_1 and y_2,

$$G(t,\tau) = \begin{cases} 0 & 0 < t < \tau \\ \frac{y_1(\tau)y_2(t) - y_1(t)y_2(\tau)}{y_1(\tau)y_2'(\tau) - y_1'(\tau)y_2(\tau)} & \tau < t \end{cases}.$$

G can be simplified,

$$\begin{aligned} \frac{y_1(\tau)y_2(t) - y_1(t)y_2(\tau)}{y_1(\tau)y_2'(\tau) - y_1'(\tau)y_2(\tau)} &= \frac{e^{-\lambda_1\tau-\mu_1 t} - e^{-\lambda_1 t-\mu_1\tau}}{(\lambda_1-\mu_1)e^{-(\lambda_1+\mu_1)\tau}} \\ &= \frac{1}{\lambda_1-\mu_1}\left(e^{\mu_1(\tau-t)} - e^{\lambda_1(\tau-t)}\right). \end{aligned}$$

Therefore

$$G(t,\tau) = \begin{cases} 0 & 0 < t < \tau \\ \frac{1}{\lambda_1-\mu_1}\left(e^{\mu_1(\tau-t)} - e^{\lambda_1(\tau-t)}\right) & \tau < t \end{cases} \tag{5.17}$$

and the integral (5.15) for u_p is given by plugging in G and u_0,

$$\begin{aligned}
u_p(t) &= \int_0^t \left(\frac{e^{\mu_1(\tau-t)} - e^{\lambda_1(\tau-t)}}{\lambda_1 - \mu_1}\right) \left[-u_0'(\tau) - \lambda_1\phi_2(\tau)u_0(\tau)\right] d\tau \\
&= \int_0^t \left(\frac{e^{\mu_1(\tau-t)} - e^{\lambda_1(\tau-t)}}{\lambda_1 - \mu_1}\right) \Big[\lambda_1(1-C_1)e^{-\lambda_1\tau} + \mu_1 C_1 e^{-\mu_1\tau} \\
&\qquad - \lambda_1\phi_2(\tau)(1-C_1)e^{-\lambda_1\tau} - \lambda_1\phi_2(\tau)C_1e^{-\mu_1\tau}\Big] d\tau \\
&= \int_0^t \left(\frac{e^{\mu_1(\tau-t)} - e^{\lambda_1(\tau-t)}}{\lambda_1 - \mu_1}\right) \Big[\lambda_1(1-\phi_2(\tau))(1-C_1)e^{-\lambda_1\tau} \\
&\qquad + \left(\mu_1 - \lambda_1\phi_2(\tau)\right) C_1 e^{-\mu_1\tau}\Big] d\tau \\
&= \int_0^t \left[\frac{\lambda_1(1-\phi_2(\tau))(1-C_1)}{\lambda_1 - \mu_1}\left(e^{-(\lambda_1-\mu_1)\tau-\mu_1 t} - e^{-\lambda_1 t}\right)\right. \\
&\qquad \left. + \frac{\left(\mu_1 - \lambda_1\phi_2(\tau)\right) C_1}{\lambda_1 - \mu_1}\left(e^{-\mu_1 t} - e^{(\lambda_1-\mu_1)\tau-\lambda_1 t}\right)\right] d\tau. \qquad (5.18)
\end{aligned}$$

Now we split the integral (5.18) up into four separate parts and solve each individually. Fortunately the integrals can be paired together into two similar types, which we will now discuss.

5.2.5.1 First Pair of Integrals, $\int \phi_2(\tau)d\tau$

We will do the more straightforward pair of integrals first. We take as an example

$$\begin{aligned}
&\int_0^t -\frac{\lambda_1(1-\phi_2(\tau))(1-C_1)}{\lambda_1 - \mu_1} e^{-\lambda_1 t} d\tau \\
&= -\frac{\lambda_1(1-C_1)}{\lambda_1 - \mu_1} e^{-\lambda_1 t} \int_0^t \frac{-1}{\frac{\lambda_2}{\mu_2-\lambda_2} + \left(\frac{1}{s_2-1} + \frac{\lambda_2}{\lambda_2-\mu_2}\right) e^{(\mu_2-\lambda_2)\tau}} d\tau \\
&= \frac{\lambda_1(1-C_1)}{\lambda_1 - \mu_1} e^{-\lambda_1 t} \int_0^t \frac{\mu_2/\lambda_2 - 1}{1 + \left(\frac{\mu_2/\lambda_2 - 1}{s_2 - 1} - 1\right) e^{(\mu_2-\lambda_2)\tau}} d\tau \\
&= \frac{\lambda_1(1-C_1)(\mu_2/\lambda_2 - 1)}{\lambda_1 - \mu_1} e^{-\lambda_1 t} \left[\tau - \frac{\ln\left[(C_2-1)e^{(\mu_2-\lambda_2)\tau} + 1\right]}{\mu_2 - \lambda_2}\right]_{\tau=0}^{\tau=t}
\end{aligned}$$

where

$$C_2 = \frac{1 - \mu_2/\lambda_2}{1 - s_2}. \qquad (5.19)$$

Continuing with the integral,

$$
\begin{aligned}
&= \frac{\lambda_1(\mu_2-\lambda_2)}{\lambda_2(\lambda_1-\mu_1)}(1-C_1)e^{-\lambda_1 t}\left[t - \frac{\ln\left[(C_2-1)e^{(\mu_2-\lambda_2)t}+1\right]}{\mu_2-\lambda_2} + \frac{\ln(C_2)}{\mu_2-\lambda_2}\right] \\
&= \frac{\lambda_1(1-C_1)}{\lambda_2(\lambda_1-\mu_1)}e^{-\lambda_1 t}\left[(\mu_2-\lambda_2)t - \ln\left[(C_2-1)e^{(\mu_2-\lambda_2)t}+1\right] + \ln(C_2)\right] \\
&= \frac{\lambda_1(1-C_1)}{\lambda_2(\lambda_1-\mu_1)}e^{-\lambda_1 t}\left[\ln\left(e^{(\mu_2-\lambda_2)t}\right) + \ln\left[\frac{C_2}{(C_2-1)e^{(\mu_2-\lambda_2)t}+1}\right]\right] \\
&= \frac{\lambda_1(1-C_1)}{\lambda_2(\lambda_1-\mu_1)}e^{-\lambda_1 t}\ln\left[\frac{C_2 e^{(\mu_2-\lambda_2)t}}{(C_2-1)e^{(\mu_2-\lambda_2)t}+1}\right] \\
&= \frac{\lambda_1(1-C_1)}{\lambda_2(\lambda_1-\mu_1)}e^{-\lambda_1 t}\ln\left[\frac{C_2}{C_2-1+e^{(\lambda_2-\mu_2)t}}\right]. \qquad (5.20)
\end{aligned}
$$

This is not particularly pleasant, but it is at least tractable. I have also confirmed numerically that this expression is correct. Expanding the logarithm in powers of s_2 should be fun.

We can do the same manipulations with

$$
\begin{aligned}
&\int_0^t \frac{(\mu_1-\lambda_1\phi_2(\tau))\,C_1}{\lambda_1-\mu_1}e^{-\mu_1 t}d\tau = \frac{\lambda_1 C_1}{\lambda_1-\mu_1}e^{-\mu_1 t}\int_0^t\left(\frac{\mu_1}{\lambda_1}-\phi_2(\tau)\right)d\tau \\
&= \frac{\lambda_1 C_1}{\lambda_1-\mu_1}e^{-\mu_1 t}\int_0^t\left[\frac{\mu_1}{\lambda_1} - 1 - \frac{\mu_2/\lambda_2-1}{1+\left(\frac{\mu_2/\lambda_2-1}{s_2-1}-1\right)e^{(\mu_2-\lambda_2)\tau}}\right]d\tau \qquad (5.21)
\end{aligned}
$$

This part we will save for a later date. It's a bit tedious, and I'd rather discuss the more interesting problem of the second pair of integrals.

5.2.5.2 Second Pair of Integrals, $\int \phi_2(\tau)e^{-x\tau}d\tau$ for large x

Now we consider the second pair of integrals. For example,

$$
\begin{aligned}
&\int_0^t \frac{\lambda_1(1-\phi_2(\tau))(1-C_1)}{\lambda_1-\mu_1}e^{-(\lambda_1-\mu_1)\tau-\mu_1 t}d\tau \\
&= \frac{\lambda_1(1-C_1)}{\lambda_1-\mu_1}e^{-\mu_1 t}\int_0^t(1-\phi_2(\tau))e^{-(\lambda_1-\mu_1)\tau}d\tau \\
&= \frac{\lambda_1(1-C_1)}{\lambda_1-\mu_1}e^{-\mu_1 t}\int_0^t -\frac{e^{-(\lambda_1-\mu_1)\tau}}{\frac{\lambda_2}{\mu_2-\lambda_2}+\left(\frac{1}{s_2-1}+\frac{\lambda_2}{\lambda_2-\mu_2}\right)e^{(\mu_2-\lambda_2)\tau}}d\tau \qquad (5.22)
\end{aligned}
$$

The integral $\int_0^t(1-\phi_2(\tau))e^{-(\lambda_1-\mu_1)\tau}d\tau$ appears to be intractable at first glance. Asking Mathematica to evaluate it returns a Gauss Hypergeometric function, which makes sense

given the previous literature. (5.3) has been solved in terms of a hypergeometric function by [AK11] and used by [BRA13] to study the evolution of cancer under combination drug therapy. We can solve these integrals using Integration by Parts. We shall leave the details for publication.

Finally, given that we've solved for $u \sim u_0 + \nu u_1$, we can plug this back into our expression for ϕ_{ij} to obtain the PGF. This allows us to skip the numerical integration step in chapter 4. Given that the numerical integration appears to be the computational bottleneck, this would improve our parameter searches for inference immensely. Additionally, it remains to expand ϕ_{ij} in powers of s_1 and s_2 to obtain approximate asymptotic expressions for the transition probabilities. This would circumvent performing the numerical inversion and would potentially provide analytical insights that we cannot obtain from numerical methods alone.

5.3 Additional Projects

In addition to these immediate extensions of my previous work, we have several alternative projects that we have been working on. The first involves computing the mean time to extinction (MTE) of cancer under immunotherapy, including quiescence. We have developed a stochastic predator-prey model for the interactions between the tumor cells, undergoing a birth-death process, and the immune cells preying on the tumor cells, resulting in the recruitment of more immune cells. This also includes reversible quiescence of the tumor cells to evade the immune system. This stochasticity is required to accurately model extinction phenonema. In particular, this model posses a *metastable* equilibrium that the process evolves around for a substantial length of time prior to absorption via extinction of the tumor. It is possible, using methods reviewed in [AM17], to use a WKB-style approximation to compute the MTE.

This may have significant clinical relevance, considering that if the mean extinction time is significantly large, this may give the existent tumor population time enough to evolve resistance to immunotherapy. It would be particularly interesting to discern how quiescence

changes the MTE. If, as suspected, quiescence dramatically increases the MTE, then it suggests that discovering a drug for removing these quiescent cells may be required to completely eradicate a tumor under immunotherapy. This could suggest a mechanism by which cancer, particularly melanoma, evolves long-term resistance to immunotherapy.

Much in the same vein as the previous problem of computing the MTE, it is possible to numerically compute numerous interesting properties of interacting particle systems. This relies on several mathematical techniques developed in statistical mechanics, notably in representing the transition probability function as a Feynman-style path integral over the space of possible trajectories of the system. This representation can be simplified by noting that the dominant contribution to the path integral comes from the contribution of the most likely path. This dominant contribution is controlled asymptotically by the small distance between lattice sites. Finding the dominant path requires solving a system of Hamilton-Jacobi PDEs, a task which is numerically feasible if analytically intractable. We can use this approach to compute extinction probabilities and detect critical phase transitions of the processes at hand.

[ACK06] Anne Auger, Philippe Chatelain, and Petros Koumoutsakos. "R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps." The Journal of Chemical Physics, **125**(8):084103, 2006.

[ACM02] Janis L Abkowitz, Sandra N Catlin, Monica T McCallie, and Peter Guttorp. "Evidence that the number of hematopoietic stem cells per animal is conserved in mammals." Blood, The Journal of the American Society of Hematology, **100**(7):2665–2667, 2002.

[AG07] Søren Asmussen and Peter W Glynn. Stochastic simulation: algorithms and analysis, volume 57. Springer Science & Business Media, 2007.

[AK11] Tibor Antal and PL Krapivsky. "Exact solution of a two-type branching process: models of tumor progression." Journal of Statistical Mechanics: Theory and Experiment, **2011**(08):P08018, 2011.

[AM17] Michael Assaf and Baruch Meerson. "WKB theory of large deviations in stochastic populations." Journal of Physics A: Mathematical and Theoretical, **50**(26):263001, 2017.

[And08] David F Anderson. "Incorporating postleap checks in tau-leaping." The Journal of Chemical Physics, **128**(5):054103, 2008.

[Bai90] Norman TJ Bailey. The elements of stochastic processes with applications to the natural sciences, volume 25. John Wiley & Sons, 1990.

[BEK17] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. "Julia: A fresh approach to numerical computing." SIAM Review, **59**(1):65–98, 2017.

[BF06] Michael GB Blum and Olivier François. "Which random processes describe the tree of life? A large-scale study of phylogenetic tree imbalance." Systematic Biology, **55**(4):685–691, 2006.

[BKL75] Alfred B Bortz, Malvin H Kalos, and Joel L Lebowitz. "A new algorithm for Monte Carlo simulation of Ising spin systems." Journal of Computational Physics, **17**(1):10–18, 1975.

[BO13] Carl M Bender and Steven A Orszag. Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory. Springer Science & Business Media, 2013.

[BRA13] Ivana Bozic, Johannes G Reiter, Benjamin Allen, Tibor Antal, Krishnendu Chatterjee, Preya Shah, Yo Sup Moon, Amin Yaqubie, Nicole Kelly, Dung T Le, et al. "Evolutionary dynamics of cancer in response to targeted combination therapy." elife, **2**:e00747, 2013.

[CGP06] Yang Cao, Daniel T Gillespie, and Linda R Petzold. "Efficient step size selection for the tau-leaping simulation method." The Journal of Chemical Physics, **124**(4):044109, 2006.

[CS12] Forrest W Crawford and Marc A Suchard. "Transition probabilities for general birth–death processes with applications in ecology, genetics, and evolution." Journal of mathematical biology, **65**(3):553–580, 2012.

[CSL16] Forrest W Crawford, Timothy C Stutz, and Kenneth Lange. "Coupling bounds for approximating birth–death processes by truncation." Statistics & probability letters, **109**:30–38, 2016.

[CV07] Abhijit Chatterjee and Dionisios G Vlachos. "An overview of spatial microscopic and accelerated kinetic Monte Carlo methods." Journal of Computer-aided Materials Design, **14**(2):253–308, 2007.

[DMP08] James W Demmel, Osni A Marques, Beresford N Parlett, and Christof Vömel. "Performance and accuracy of LAPACK's symmetric tridiagonal eigensolvers." SIAM Journal on Scientific Computing, **30**(3):1508–1526, 2008.

[DP04] Inderjit S Dhillon and Beresford N Parlett. "Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices." Linear Algebra and its Applications, **387**:1–28, 2004.

[Dur15] Richard Durrett. "Branching process models of cancer." In Branching process models of cancer, pp. 1–63. Springer, 2015.

[Fad97] Malcolm J Faddy. "Extended Poisson process modelling and analysis of count data." Biometrical Journal, **39**(4):431–440, 1997.

[Fel08] Willliam Feller. An introduction to probability theory and its applications, volume 2. John Wiley & Sons, 2008.

[FFB19] Alexander M Firsov, Maksim A Fomich, Andrei V Bekish, Olga L Sharko, Elena A Kotova, Harry J Saal, Dragoslav Vidovic, Vadim V Shmanai, Derek A Pratt, Yuri N Antonenko, et al. "Threshold protective effect of deuterated polyunsaturated fatty acids on peroxidation of lipid bilayers." The FEBS Journal, **286**(11):2099–2117, 2019.

[GHP13] Daniel T Gillespie, Andreas Hellander, and Linda R Petzold. "Perspective: Stochastic algorithms for chemical kinetics." The Journal of Chemical Physics, **138**(17):05B201_1, 2013.

[Gil77] Daniel T Gillespie. "Exact stochastic simulation of coupled chemical reactions." The Journal of Physical Chemistry, **81**(25):2340–2361, 1977.

[Gil01] Daniel T Gillespie. "Approximate accelerated stochastic simulation of chemically reacting systems." The Journal of Chemical Physics, **115**(4):1716–1733, 2001.

[Inc26] EL Ince. "Ordinary differential equations (London and New York.", 1926.

[Ken66] David G Kendall. "Branching processes since 1873." Journal of the London Mathematical Society, **1**(1):385–406, 1966.

[KPS17] Jakob Nikolas Kather, Jan Poleszczuk, Meggy Suarez-Carmona, Johannes Krisam, Pornpimol Charoentong, Nektarios A Valous, Cleo-Aron Weis, Luca Tavernar, Florian Leiss, Esther Herpel, et al. "In silico modeling of immunotherapy and stroma-targeting therapies in human colorectal cancer." Cancer Research, **77**(22):6442–6452, 2017.

[KRA07] Sonia Kéfi, Max Rietkerk, Concepción L Alados, Yolanda Pueyo, Vasilios P Papanastasis, Ahmed ElAich, and Peter C De Ruiter. "Spatial vegetation patterns and imminent desertification in Mediterranean arid ecosystems." Nature, **449**(7159):213, 2007.

[Lan82] Kenneth Lange. "Calculation of the equilibrium distribution for a deleterious gene by the finite Fourier transform." Biometrics, pp. 79–86, 1982.

[Lan10] Kenneth Lange. Applied probability. Springer Science & Business Media, 2010.

[Lig12] Thomas Milton Liggett. Interacting particle systems, volume 276. Springer Science & Business Media, 2012.

[Lig13] Thomas M Liggett. Stochastic interacting systems: contact, voter and exclusion processes, volume 324. Springer Science & Business Media, 2013.

[Lin02] Torgny Lindvall. Lectures on the coupling method. Courier Corporation, 2002.

[LJC17] Xiaoyang Lan, David J Jörg, Florence MG Cavalli, Laura M Richards, Long V Nguyen, Robert J Vanner, Paul Guilhamon, Lilian Lee, Michelle M Kushida, Davide Pellacani, et al. "Fate mapping of human glioblastoma reveals an invariant stem cell hierarchy." Nature, **549**(7671):227–232, 2017.

[LSK18] Alfonso Landeros, Timothy C Stutz, Kevin L Keys, Alexander Alekseyenko, Janet S Sinsheimer, Kenneth Lange, and Mary E Sehl. "BioSimulator. jl: Stochastic simulation in Julia." Computer <ethods and Programs in Biomedicine, **167**:23–35, 2018.

[MB07] Tatiana T Marquez-Lago and Kevin Burrage. "Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics." The Journal of Chemical Physics, **127**(10):09B603, 2007.

[MHI05] Franziska Michor, Timothy P Hughes, Yoh Iwasa, Susan Branford, Neil P Shah, Charles L Sawyers, and Martin A Nowak. "Dynamics of chronic myeloid leukaemia." Nature, **435**(7046):1267–1270, 2005.

[MPC06] James M McCollum, Gregory D Peterson, Chris D Cox, Michael L Simpson, and Nagiza F Samatova. "The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior." Computational Biology and Chemistry, **30**(1):39–49, 2006.

[NKK06] Artem S Novozhilov, Georgy P Karev, and Eugene V Koonin. "Biological applications of the theory of birth-and-death processes." Briefings in bioinformatics, **7**(1):70–85, 2006.

[NW14] Albert Nijenhuis and Herbert S Wilf. Combinatorial algorithms: for computers and calculators. Elsevier, 2014.

[PDV10] Janesh Pillay, Ineke Den Braber, Nienke Vrisekoop, Lydia M Kwast, Rob J De Boer, José AM Borghans, Kiki Tesselaar, and Leo Koenderman. "In vivo labeling with 2H2O reveals a human neutrophil lifespan of 5.4 days." Blood, The Journal of the American Society of Hematology, **116**(4):625–627, 2010.

[Per18] Jeffrey M Perkel. "Why Jupyter is data scientists' computational notebook of choice." Nature, **563**(7732):145–147, 2018.

[PP07] Imre Pázsit and Lénard Pál. Neutron fluctuations: A treatise on the physics of branching processes. Elsevier, 2007.

[PZF17] Amit A Patel, Yan Zhang, James N Fullerton, Lies Boelen, Anthony Rongvaux, Alexander A Maini, Venetia Bigley, Richard A Flavell, Derek W Gilroy, Becca Asquith, et al. "The fate and lifespan of human monocyte subsets in steady state and systemic inflammation." Journal of Experimental Medicine, **214**(7):1913–1923, 2017.

[RMF07] Tobias Reichenbach, Mauro Mobilia, and Erwin Frey. "Mobility promotes and jeopardizes biodiversity in rock–paper–scissors games." Nature, **448**(7157):1046, 2007.

[RN17] Christopher Rackauckas and Qing Nie. "Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia." Journal of Open Research Software, **5**(1), 2017.

[SAL09] Mary Sehl, Alexander V Alekseyenko, and Kenneth L Lange. "Accurate stochastic simulation via the step anticipation τ-leaping (SAL) algorithm." Journal of Computational Biology, **16**(9):1195–1208, 2009.

[WBP15] Bartlomiej Waclaw, Ivana Bozic, Meredith E Pittman, Ralph H Hruban, Bert Vogelstein, and Martin A Nowak. "A spatial model predicts that dispersal and cell turnover limit intratumour heterogeneity." Nature, **525**(7568):261, 2015.

[XM15] Jason Xu and Vladimir N Minin. "Efficient transition probability computation for continuous-time branching processes via compressed sensing." In Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence, volume 2015, p. 952. NIH Public Access, 2015.