Quantitative Approaches to Cancer and Cellular Differentiation

Quantitative Approaches to Cancer and Cellular Differentiation

A DISSERTATION PRESENTED
BY
JEREMY LLOYD FERLIC
TO
THE DEPARTMENT OF BIOSTATISTICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
BIOSTATISTICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
JULY 2019

Thesis advisor: Professor Franziska Michor Jeremy Lloyd Ferlic

Quantitative Approaches to Cancer and Cellular Differentiation

Abstract

In the past decade, advances in computational power have expanded the potential to analyze vastly complex systems. While this explosion of research has made great contributions in the fields of biology and human health, there remains a need to develop frameworks and tools to provide greater access for general scientists to these new methods. In this work, we describe a framework and two software packages to help further investigate cancer and cellular differentiation.

In the first chapter, we evaluate the effects of implementing screening for the precancerous state monoclonal gammopathy of undetermined significance (MGUS) in the progression to multiple myeloma (MM). Advances in medicine have discovered therapeutic and lifestyle interventions that potentially reduce the risk of progression from MGUS to MM. It remains an open question how best to implement a screening strategy and how to evaluate the effects of the new policy. We model the United States population containing high- and low- risk subgroups and compare screening regimens using MGUS and MM incidence and mortality measures.

Next, in the second chapter we present a computational tool, DIFFpop, an R package to

simulate the movement of cells through differentiation hierarchies. The software includes functionalities to simulate clonal evolution due to the emergence of driver mutations under the infinite-allele assumption as well as options for simulation and analysis of single cell barcoding and labeling data. The software uses the Gillespie Stochastic Simulation Algorithm and a modification of expanding or fixed-size stochastic process models expanded to a large number of cell types and scenarios.

Finally, in the third chapter, we develop a new method and tool to estimate rate parameters for and simulate continuous-time Markov branching processes. The software includes methods to simulate branching processes according to time-dependent rates as well as random distributions of offspring. ESTIpop uses the Gillespie Stochastic Simulation Algorithm with adaptive thinning. Parameter estimation is based on an extension of the Central Limit Theorem applied to multitype branching processes with ancestors of various types. The software is flexible and can be applied to any user-defined multitype branching processes.

Contents

1 Evaluation of Screening for Precancerous States **1**
- 1.1 Introduction . . . 1
- 1.2 Materials and Methods . . . 4
- 1.3 Results . . . 13
- 1.4 Discussion . . . 24

2 DIFFpop: Simulation of Differentiation Hierarchies **29**
- 2.1 Introduction . . . 29
- 2.2 Software Description . . . 32
- 2.3 Usage . . . 41
- 2.4 Applications . . . 49
- 2.5 Conclusion . . . 54

3 ESTIpop: Estimation of Continuous-time Markov Branching Processes **55**
- 3.1 Introduction . . . 55
- 3.2 Software Description . . . 58
- 3.3 Usage . . . 66
- 3.4 Application . . . 77
- 3.5 Conclusion . . . 88

Appendix A Analytical Approaches in Evaluation of Screening for Precancerous States **89**
- A.1 Analytical Markov Chain Model . . . 89
- A.2 Cumulative MM-specific mortality conditioned on MGUS detection . . . 94
- A.3 Evolving MGUS . . . 96

Appendix B DIFFpop Supplemental Materials **97**
- B.1 Application . . . 97
- B.2 Vignette 1: Branching Process . . . 105
- B.3 Vignette 2: Multi-type Moran Process . . . 121

Appendix C ESTIpop Supplemental Materials **136**
- C.1 Methods . . . 136
- C.2 Vignette 1: One-Type Birth-Death Process . . . 149
- C.3 Vignette 2: Two-Type Birth-Death-Mutation Process . . . 171

C.4 Vignette 3: Three-Type Process 187

REFERENCES **200**

Acknowledgments

I WOULD FIRST AND FOREMOST like to express my gratitude and thanks to my advisor, Professor Franziska Michor, who has been an excellent role model of not only a scientist, but also a human being. She has guided me through this journey and has been an invaluable source of knowledge and wisdom. She believed in me even when I did not believe in myself.

Next, I want to thank my committee members, Professors John Quackenbush and G.C. Yuan, who have always provided advice and a fresh perspective when looking at my work. In addition, I would like to thank Thomas "Ollie" McDonald, Philipp Altrock, and the rest of the Michor lab for all of the help over the years. To my classmates: I might not have seen much of you these past few years, but it gave me an incredible amount of comfort to know there were other people going through this with me.

Words cannot express how lucky I have been to have the support and love of my family. Mom, Dad, Michael, and Brett: thank you. Thank you to Momo and Zuko for being great writing buddies and always making sure my laptop was covered in your fur. Lastly, I need to thank Brian Zanghi. I could not have done this without you.

1

Evaluation of Screening for Precancerous States

1.1 Introduction

Accounting for 1.8% of new cancer cases and 2.1% of cancer deaths annually[3], multiple myeloma (MM) is the second most common hematologic malignancy in the United States. MM is a

plasma cell malignancy[33] in which patients show abnormal levels of the paraprotein M protein[63]. This increase in M protein concentration indicates a monoclonal cell population and end-organ damage such as lytic bone lesions[41]. Most patients who go on to develop MM have experienced progression from a precursor condition, monoclonal gammopathy of undetermined significance (MGUS), in which individuals only exhibit a spike in M protein concentration[41]. In the United States, MGUS has a prevalence of around 2% in the population over the age of 50[46], with evidence that men have higher age-adjusted incidence rates than women[44]. Race is also a significant risk factor, as MGUS prevalence at age 40 in African Americans is roughly equivalent to MGUS prevalence at age 50 in non-African Americans[45].

Advancements in MM therapeutics suggest that the progression rate from MGUS to MM can be modified[14,18]. In patients with type 2 diabetes, metformin-use is associated reduced progression rates potentially delaying the progression from MGUS to MM by 4 years[18]. Obesity is associated with higher progression rates to MM[76,17,13]. Regular aspirin use in the general population has also been associated with reduced progression risk[14]. While further study is required to establish causal relationships and determine the exact molecular mechanisms that undermine those relationships, these advances suggest that various interventions have the potential to reduce the risk of progression from MGUS to MM, ultimately decreasing the mortality burden due to this disease. Thus, it is of particular interest to study the effects of screening the population for MGUS, especially in the high-risk African American population, with the goal of detecting MGUS earlier. As a result of early detection and interventions to reduce the progression risk from MGUS to MM, overall MM prevalence and mortality can be reduced.

Independent of intervention-based risk reduction, knowledge of the precursor state can affect mortality and comorbidity in patient cohorts. One study found that patients with MM who had prior knowledge of MGUS had overall improved survival (median, 2.8 years) when compared against MM patients without prior knowledge of MGUS (median, 2.11 years)[70]. The authors concluded that as a result of prior knowledge, earlier treatment of MM leads to better survival, although this result is potentially conflicted by lead bias. Further conclusions include that clinical follow-up in cases of accidental MGUS detection may be important regardless of risk type[70] and follow-up preceding the diagnosis of MGUS-related malignancy may lead to improved survival[29]. In light of these findings, specifically screening for MGUS has additional merit given that currently less that 10% of MM diagnoses are knowingly associated with preexisting MGUS[70,29].

Here we designed a computational model that describes the incidence of MGUS and progression to MM under varying screening regimens and therapeutic or lifestyle interventions. The model is based on epidemiological data of MGUS and MM incidence and progression rates, which can depend on genetic background, sex, and age[75,47] and correlate with ethnicity[78]. Using both simulated and analytic results, we assessed whether a given intervention with an associated reduction in progression risk following a positive MGUS screen could both reduce MM prevalence and lead to an overall reduction in MM-specific mortality. Although presented here in the context of MGUS and MM, the general framework remains applicable to various precancerous and cancerous states and can also be useful in the reexamining of screening guidelines as future interventions are developed.

1.2 Materials and Methods

To investigate the effect of implementing a screening procedure in the development cycle of MM, we developed a Markov chain model (Figure 1.1A) of this system in the general United States population. In the model, healthy members of the population can develop undetected MGUS according to age and risk group-dependent rates. Those undetected MGUS individuals can then transition to a detected MGUS state upon successful screening. Individuals harboring MGUS progress at a constant rate towards the full MM state; however, those with a successful MGUS screen progress at a reduced rate (Figures 1.1B & 1.1C). At any point in the process individuals my die according to age-specific death rates, but mortality rates are higher for those who have developed MM. We then performed stochastic simulation of this model and derived an analytic framework to assess mortality due to MM and the assess the reduction in prevalence of MM after implementation of the screening regimen.

Model Inputs and Outputs

In our interest to model the United State population, we encountered a mixed population in terms of lifetime risk of developing MGUS. Prior studies show that the African American population has a roughly two-fold increase in lifetime risk of MGUS[47,48] compared to their non-African American counterparts. Thus, we consider the non-African American population and African American population as the low-risk (baseline) and high-risk groups respectively. We employed a crude birth rate for all individuals in the population as well as national life

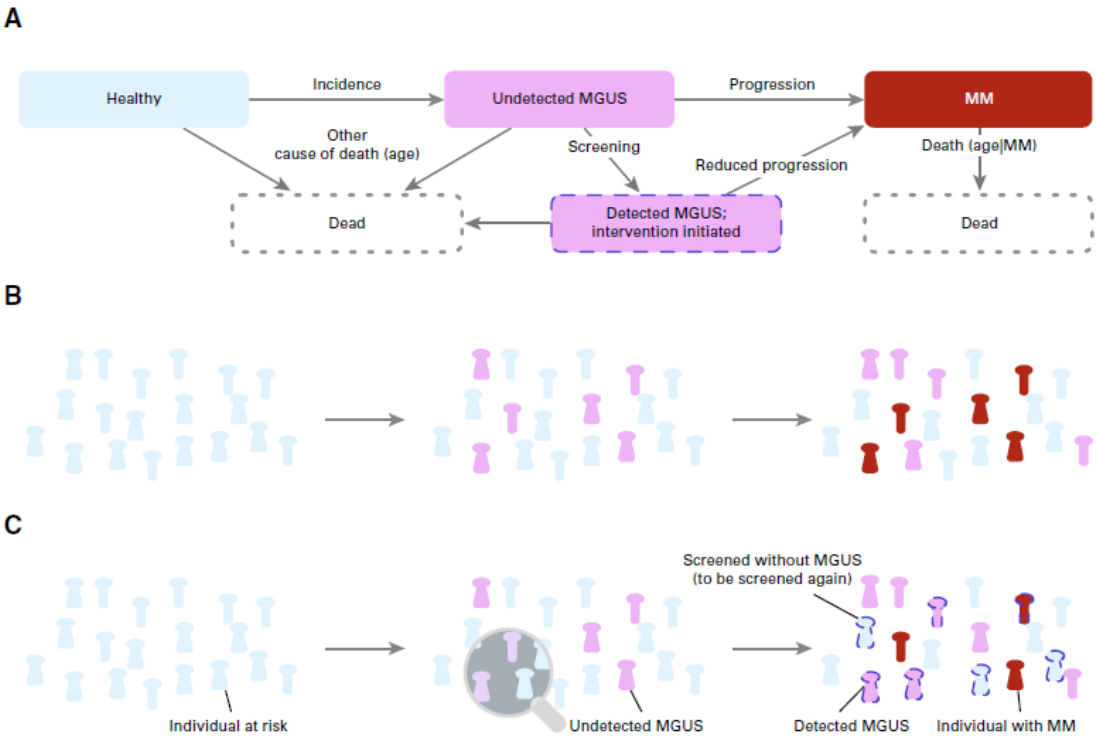

Figure 1.1: Schematic model of the disease trajectory for individuals with monoclonal gammopathy of undetermined significance (MGUS) as well as those with multiple myeloma (MM). (A) The transitions in the disease trajectory can be modeled using a Markov chain. The four possible states are healthy (blue), undetected MGUS (pink), detected MGUS (pink with dashed outline), and MM (red). (B) In the absence of screening, some individuals develop MGUS and some of those progress to MM. (C) In the presence of screening and effective intervention that reduces MGUS to MM progression. Individuals are screened and, if identified as having MGUS, receive a preventive treatment. This leads to a lower number of MM cases, although individuals may still progress to MM even at a reduced rate..

tables to determine death events for healthy individuals and individuals harboring MGUS, either undetected or detected. For individuals with MM, we used MM-specific death rates and assumed a fixed progression rate from MGUS to MM for individuals with undetected MGUS. Each screening regimen can be summarized using three parameters: the age at which screening is initiated (a_0), the average length of time between screens (Δa), and the reduction in progression risk following a successful screen (r) (Table 1.1).

As outputs of our model, we were primarily interested in the effect that various screening scenarios, as defined by the aforementioned parameters, have on MM-specific mortality after MGUS detection and on the fraction of individuals that progress to MM across all ages. All simulations were initiated according to the age distributions in the United States according to information from the 2013 census[7,4]. Although the fraction of African Americans in the United States is roughly 13%[7], we assumed a 20% mixture of high-risk individuals under the assumption that the genetic diversity in the United States would further contribute to the high-risk population.

Table 1.1: Parameters Used for Computational and Mathematic Modeling

Parameter	Description	Range or Value	Reference
a	Age	0-100 years	[7]
$d(a)$	Probability of dying as a result of any cause at age a	0-1, age-dependent	[3,7]
d_{MM}	Probability of dying as a result of MM	0.1295 per patient with MM per year	[61,42]
$m(a)$	Incidence rate of MGUS	0-1 per person per year, age-dependent, risk-group-dependent	[75]
p	Probability of progression from MGUS to MM	0-0.15 per person per year, depending on progression model and disease evolution	[83,64,66]
a_0	Age at first MGUS screen	20-50 years	This work
Δa	Average interval between screens	1-15 years	This work
r	Reduction in progression rate, conditional on MGUS detection	0-1; for example, if $r = 0.5$, then $p = 0.5 \times 0.01 = 0.005$ per patient with MGUS per year	[14,18]

Stochastic Model

We simulated the Markov chain model (Figure 1.1A) by using a crude birth rate[5], age-dependent death rates for healthy individuals as well as those inflicted with MGUS[7], both undetected and detected, and a fixed death rate for patients who developed MM[35]. MGUS incidence rates for the low-risk population were adapted from Therneau et al.[75]. Using the baseline MGUS incidence rates, we calculated elevated incidence rates for the high-risk population such that the lifetime risk of developing MGUS in the African American population is approximately two-fold higher than that of the baseline population[47,50]. Based on the work by Zingone and Kuehl[83], progress from MGUS to MM was constant across risk groups and occurred at a rate of $p = 0.01$ per year in undetected MGUS individuals[64].

Beginning at age a_0, with probability $1/\Delta a$ an individual was selected for MGUS screening, meaning that on average, an individual was screened once every Δa years. This choice was meant to mimic the somewhat stochastic nature by which individuals visit the clinic and are chosen to undergo diagnostic bloodwork. Once positively screened, an individual with detected MGUS was presumed to progress to MM at the reduced rate of $r \times p$. Recent studies have shown that among aspirin users in the general population $r = 0.61$[14]. Our simulations track the number of healthy individuals across all risk and age groups. In addition, more specific information about individuals who develop MGUS was tracked, such as MGUS status, MGUS screening status, age at MGUS development and diagnosis, MM status and age at development, and MM-specific mortality. Using this data, we were able to calculate MGUS and

MM prevalence and mortality rates. We now discuss in greater detail particular aspects of the model.

Birth and Death Events

Using a discrete time, agent-based stochastic birth-death process, we simulated the development and progression of MGUS to MM in the United States population. Advancing our simulations in single year time units, we were able to roughly approximate past U.S. population growth with a fraction of MGUS individuals that reached stationarity. As net expected birth and death rates are linearly proportional to the total size of the population, this growth can be well-characterized by using exponential growth[59]. New births in the population for a single year were drawn from a binomial distribution in which the total number of trials was set to the existing population size and the probability of success, in this case a birth, was set to 0.015, the crude birth rate in the current United States population of about 15 newborns per 1000 individuals[6]. The death of healthy individuals and those with MGUS, but not MM occurred stochastically according to census-derived, age-dependent mortality rates[7], that is, an individual of age a dies in the current year with probability $d(a)$. The assumption that a positive MGUS state, whether detected via screening or not, without progression to MM does not affect the survival of an individual reflects current knowledge about the biological underpinnings of the relationship between MGUS, smoldering MM, and full-blown MM. The probability of death due to MM was derived from the median survival rate of MM patients. The current median survival time of 5 years[35] implies that an individual survives longer than 5 years with proba-

bility $\frac{1}{2} = (1 - d_{MM})^5$, where d_{MM} is the single year probability of death due to MM. Solving this equation results in $d_{MM} = 0.1295$. For older MM patients, the probability of dying from causes other than MM could potentially be higher than the probability of dying as a result of MM. Thus, the death of every individual with MM was probabilistically attributed to either MM or to other causes with the probability of being attributed to MM proportional to $\frac{d_{MM}}{d_{MM}+d(a)}$.

In initiating simulations, all individuals were considered healthy with the initial age and risk-group distributions obtained from the United States 2013 census[7]. After a transient simulation state lasting around 30 years, the fractions of MGUS and MM individuals in the population reached stationarity. It is noteworthy that the stationary distribution of the age-structured population is independent of the initial distribution of individuals[59] due to the independence of birth and death events and the lack of any assumed carrying capacity in the population. As such, the simulated U.S. population grew over the course of simulation time. A population of constant size would only occur if births are balanced by deaths and a shrinking population would be observed if the death events outweighed the births. With our selected birth and death parameter regimes, the stochastic birth events exceeded the net of effect of stochastic death events, whether due to MM or any other cause, thus leading to overall growth in the population in which the MGUS and MM fractions reach stationarity.

MGUS INCIDENCE

We considered situations in which MM was diagnosed immediately after progression from MGUS to MM. For the incidence of MGUS, we used four risk groups each having its own set of age-dependent MGUS incidence rates: low-risk, non-African American females and males, considered the baseline risk groups, as well as the high-risk, African-American females and males. The annual incidence of MGUS in the baseline population was estimated from previously reported prevalence data from Therneau et al[75]. To translate these prevalence results to annual incidence rates, we obtained an exponential MGUS incidence law that approximates the probability than an individual of age a and risk group i will develop MGUS. This law has the form $\mu_{0,i}\exp(s_i \times a)$, where $\mu_{0,i}$ is the baseline risk for risk group i. Treating time in discrete units of 1 year, we can relate the lifetime prevalences to our annual incidence rates using:

$$\text{lifetime-risk}(s_i) = \sum_{k=0}^{100} \mu_{0,i}\exp(s_i \times k)\prod_{l=0}^{k-1}(1-d(a,i))$$

where $d(a,i)$ is the risk-group specific all-cause mortality rate. This equation reveals that we can view the lifetime risk of developing MGUS as a sum of annual risks of developing MGUS each weighted by the probability of surviving up to that particular age[68]. Fitting this form to the data from Therneau et al., we find $\mu_{0,\text{low-risk female}} = 0.000026$ and $\mu_{0,\text{low-risk male}} = 0.000033$ with $s_{\text{low-risk female}} = 0.044446$ and $s_{\text{low-risk male}} = 0.046701$. For the high-risk population, we assumed similar baseline factors, that is, $\mu_{0,\text{low-risk female}} = \mu_{0,\text{high-risk female}}$

and $\mu_{0,\text{low-risk male}} = \mu_{0,\text{high-risk male}}$, but allowed the slope of the exponential increase to have an additional factor f such that $s_{\text{high-risk female}} = f \times s_{\text{low-risk female}}$ and $s_{\text{high-risk male}} = f \times s_{\text{low-risk male}}$. To calculate this factor, we used the previously reported result that the lifetime risk of developing MGUS in the high-risk population is approximately two-fold that of the lifetime risk in the baseline population and thus,

$$\frac{\text{lifetime-risk}(s_{\text{high-risk female}}) + \text{lifetime-risk}(s_{\text{high-risk male}})}{\text{lifetime-risk}(s_{\text{low-risk female}}) + \text{lifetime-risk}(s_{\text{low-risk male}})} = 2.$$

This equation was solved numerically to result in $f = 1.267$. Using these age- and risk group-specific MGUS incidence rates in simulation, we achieved equal MGUS prevalence between the low- and high-risk populations at ages 50 and 42 for the low and high risk groups respectively. This is in good agreement with current studies of the racial disparity in MGUS prevalence[46,45].

Progression to MM

Due to previous studies suggesting that the effects of heterogeneity in MM in the population occur at the level of MGUS incidence and not at progression from MGUS to MM, we assumed that MGUS progresses to MM at a constant rate[47]. We assumed a per person per year progression rate of $p = 0.01$ per year in individuals harboring MGUS in the absence of intervention[83,64], modeled as a stochastic event where on average, 1% of MGUS individuals without intervention progressed to the MM state per year. The progression rate for MGUS individ-

uals on a particular intervention were assumed to progress to the MM state at a reduced rate, $r \times p$, where r, the risk reduction factor, indicates the efficacy of the intervention, $r \in [0, 1]$. Birmann et al. have previously reported $r = 0.61$ for "persons who regularly use aspirin" as an intervention that could be applicable to the general population[14].

Screening

We adapted the screening approach previously introduced by Zelen[80], where individual screening commences at a specific age a_0 and following tests are performed on average in Δa intervals, that is, each year after age a_0, an individual will be screened with probability $\frac{1}{\Delta a}$. Screening continues throughout the lifetime of an individual until a positive screen is returned. We further assumed that the screening test has perfect sensitivity and specificity, which is warranted by results showing that sensitivity and specificity of serum MGUS screens are 100% and 99% respectively.

1.3 Results

Prevalence of MM When Screening for MGUS

To investigate the effects of various screening regimens on MGUS and MM prevalence and mortality, we performed stochastic simulations of the agent-based model described previously. As could be expected, the proportions of these numbers varied with the fraction of high-risk individuals in the population, where a higher initial proportion of high-risk individuals lead to

more MGUS incidence, leading to higher MM statistics even with constant progression from MGUS to MM across risk groups. Increasing the efficiency of the intervention after positive MGUS detection, lowering r, drastically diminishes the number of individuals who progress to MM, while increasing the number of individuals who remain in the MGUS state holding all other variables constant (Figure 1.2A). As a validation for our model, we compared results from our model to that of the study performed by Birmann et al[14]. In this study, a cohort of 163,810 men and women contained 82 individuals who progressed from MGUS to MM at the baseline progression rate, $r = 1$, and 44 individuals who were long-term aspirin users who progressed at the reduced rate, $r = 0.61$. Birmann et al reported that the reduction in patients with MM associated with long-term aspirin use, $r = 0.61$, is 40%. This agrees with the prediction of our model which also predicts a 40% reduction in MM prevalence (Figure 1.2A).

Changes in the screening parameters a_o and Δa similarly affect MM risk reduction (Figure 1.2B). As an example, consider the scenario in which screening is initiated at age 45 and occurs on average every 8 years, $r = 0.61$, $a_0 = 45$ years, and $\Delta a = 8$ years. This screening regimen results in roughly 77.2% of the total MM diagnoses compared to in the absence of any screening benefit, $r = 1.0$. By comparison, a screening regimen defined by $r = 0.61$, $a_0 = 65$ years, and $\Delta a = 2$ years results in roughly 78.6% of the total MM diagnoses compared to in the absence of any screening benefit. Even in the case of a near complete blockage of progression, meaning that r is very close to 0, and rare screening frequency, $\Delta a = 8$ years, $a_0 = 45$ years reduces cases of MM by 60% whereas $a_0 = 65$ years reduces cases of MM by 38%. In Figure 1.2c and 1.2D, we display the normalized violin plot of the age at MM diagnoses in our

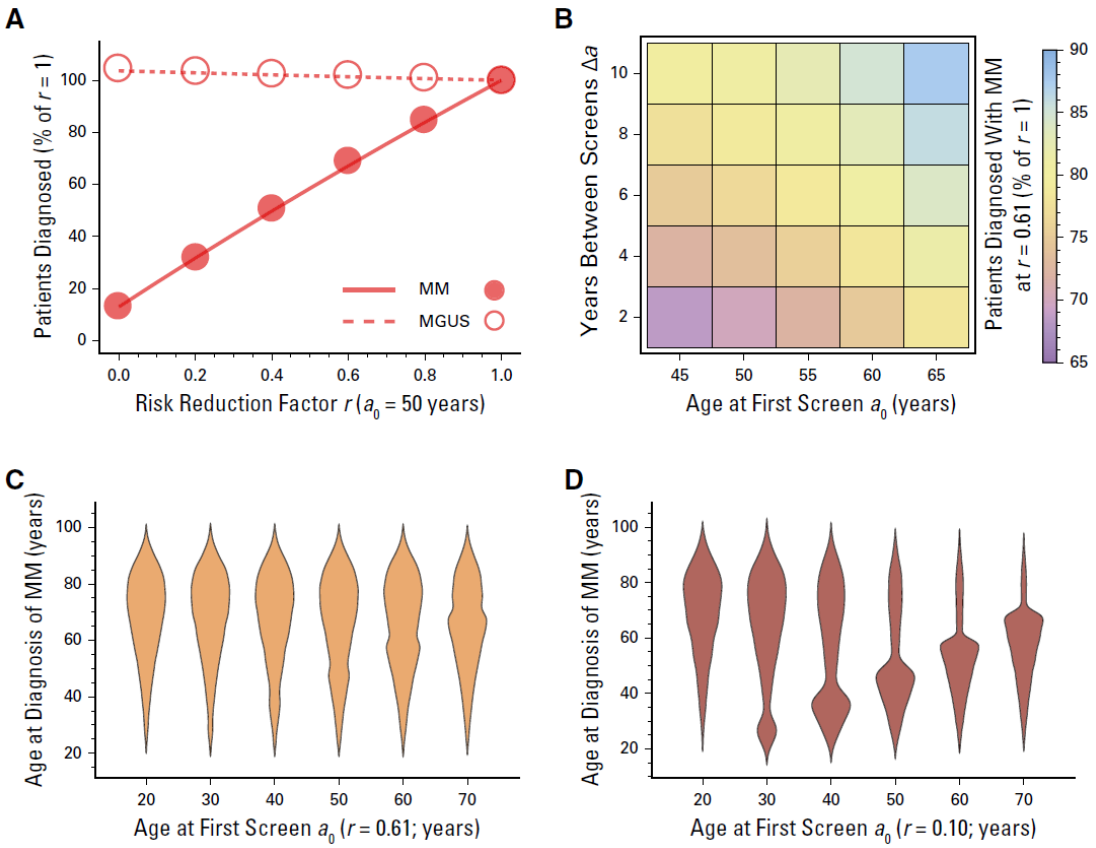

Figure 1.2: (A) When screening for MGUS was initiated, we measured the total number of patients diagnosed with MGUS (dashed line, open circles) and MM (solid line, filled circles) relative to the scenario in the absence of screening [r = 1] for various risk reduction factors (circles, simulations; lines, analytic model, See Analytical model). At r = 0.61, the MM fraction dropped to below 70% of its value in the absence of screening. (B) Variability in fraction of individuals with MM at r = 0.61 (risk reduction factor for general aspirin use), with respect to changes in a_0 and δa. (C, D) Distributions of age at MM diagnosis with yearly screening, varying a_0 and fixed r of either (C) 0.61 or (D) 0.1. The width in these plots is equal to probability of MM diagnosis at that age.

simulations for varying r, a_0, and Δa. These plots represent the relative probability of finding an individual with MM diagnosis at a specific age. In these plots, we observe a bottleneck occur at a_0, which becomes more pronounced as the intervention effectiveness increases. This bottleneck can be explained by noting that in ages earlier that a_0, these individuals exhibit no benefit in the screening regimen, but the reduction in MM diagnosis prevalence at the bottleneck can be explained as those individuals receive the benefit from initiating a screening regimen and subsequent intervention to slow progression from MGUS to MM. Thus, we have shown both that the number of cases of MM as well as the distribution of age at MM diagnosis is sensitive and can be influenced by the chosen screening parameters.

Lead-Time Bias and Cumulative MM-Specific Mortality

Screening can be a cause of lead-time bias. That is, the survival time after a positive screen is typically longer than the survival time following direct clinical presentation of a disease. The difference between these two times is called the lead-time bias[30,81]. Because of this perceived increase in survival time which might actually overshadow any true survival benefits, disease-specific mortality is a more appropriate measure for comparison[22]. In Figure 1.3A, we determined the expected lead-time bias by comparing survival times from disease diagnosis or screening in both a screened population and unscreened population with no screening benefit, $r = 1.0$. Median survival in the control group with no screening was 4 to 5 years, compared to 15 years in the group that was screened annually beginning at age 50, $a_0 = 50$ years and $\Delta a = 1$ year. Thus, the lead-time bias in this case would be around 10 years.

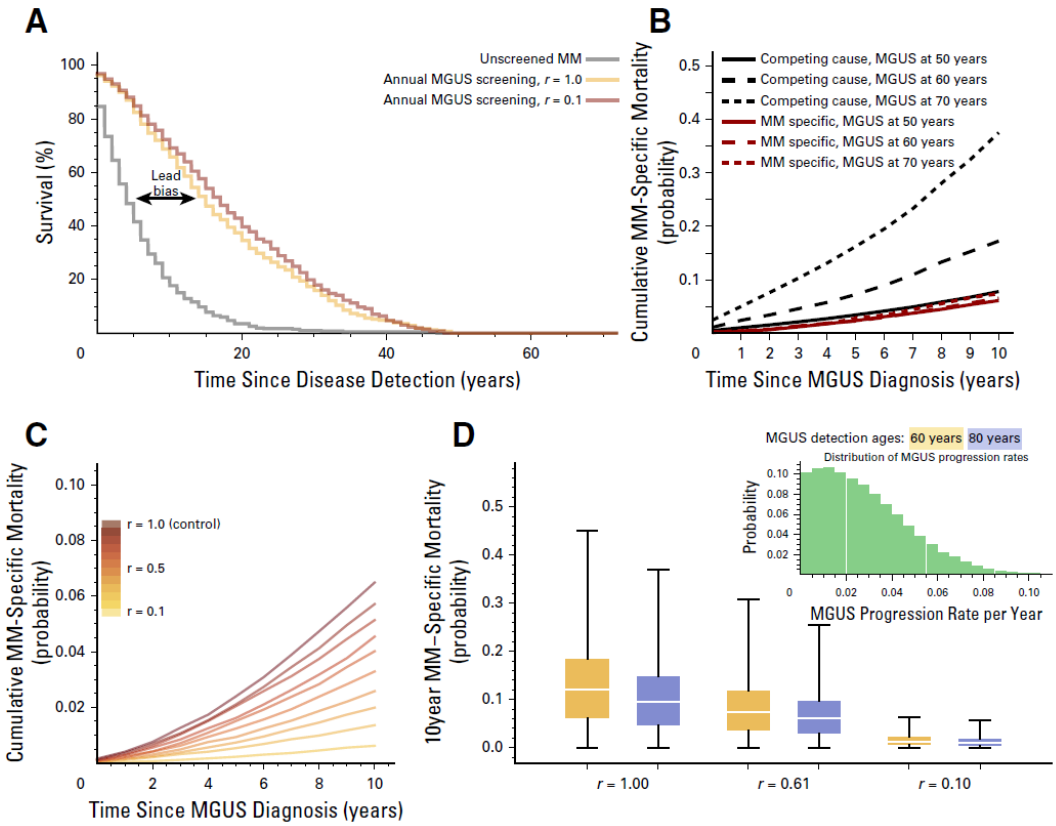

Figure 1.3: (A) Lead-time bias due to screening when comparing raw median-survival times for those who are not screened (grey: median survival, 4 years) and with screening with no progression reduction (gold: median survival, 15 years) and a significant progression reduction (red: median survival, 17 years after MGUS screen). (B) Cumulative MM-specific mortality in years after MGUS detection was measured for individuals of ages 50, 60, and 70 at MGUS detection (a_0 = 50 years, δa = 1, and r = 1). (C) MM-specific mortality varies with r (a_0 = 50 years, δa = 1), shown here for individuals diagnosed with MGUS at age 60 years, sampled from simulations. (D) MM-specific mortality is influenced by variability in progression rates from MGUS to MM (inset: MGUS progression rate distribution; truncated normal distribution with mean 0.01 and standard deviation 0.03). This effect is moderated by the risk reduction factor r.

Instead of raw survival, we calculated the cumulative MM-specific mortality after MGUS detection. This quantity is defined as the probability that an individual would die as a result of MM within a predetermined number of years after detection of MGUS at a fixed age[67]. Importantly, we separately distinguished deaths resulting from MM and deaths resulting from causes other than MM. In Figure 1.3B, we plotted the cumulative cause-specific mortalities for both MM and competing causes for MGUS detection at ages 50, 60, and 70 years. In the younger groups, the cumulative probability of dying due to MM was comparable to the chance of dying as a result of competing causes, whereas in the older cohorts, this later number increases with age. Once again, for these simulations, annual MGUS screening was initiated at age 50 with no screening benefit. As shown in Figure 1.3C, MM-specific mortality varied strongly with the intervention effectiveness *r*, with intuitively less cumulative MM-specific mortality with higher intervention effectiveness, lower *r*.

MGUS to MM Progression Variability and Evolving MGUS

Although in the previous simulations we assumed a constant rate of progression from MGUS to MM, the framework we developed allows of the assessment of the impact of individual variation in progression rates[83] as well as the impact of evolving progression rates in which the progression rate changes over time[66]. As shown in Figure 1.3D, variability in the MGUS progression parameter *p* can lead to large variation in the 10-year cumulative MM-specific mortality; however, this variability lessens as the risk reduction due to intervention increases.

As reported by Rosinol et al., patients with MGUS belong to either one of two groups:

the much larger group of individuals who experience progression from MGUS to MM at a constant rate or the smaller group of individuals who appear to experience progression from MGUS to MM at an increasing rate with age[66]. Of the 359 cases analyzed by this group, 330 (92%) were nonevolving and 29 (8%) were evolving[66]. To incorporate this evolving progression rate, we assumed that for each individual, the rate of progression exactly t years after MGUS incidence is given by the formula $\beta \times (1-\beta)^t$. Using the cumulative progression rates from Rosinol et al., we inferred that individuals with nonevolving MGUS progress at rate $\beta = 0.007$, which agrees well with our assumed constant progression rate $p = 0.01$ (Figure 1.4A). On the other hand, individuals with evolving MGUS progress at rate $\beta = 0.07$, a more than 10-fold higher value. In Figure 1.4B, we show that 10 year MM-specific mortality increases dramatically with increasing evolving MGUS parameter β, but this increase diminishes with increasing intervention efficacy (Figure 1.4C).

Effects of High-risk Immigration

Beyond population-based heterogeneity, global migration patterns into the United States could affect the use of screening[15]. Our modeling and simulation framework allowed us to analyze the effects of immigration on the distributions of high-risk and low-risk individuals in a system representative of the US population (Figure 1.5A). Two groups were added to comprise the immigrant population – immigrant females and males. Unlike population growth, immigrants are not introduced to the system by birth during every time interval of the simulation, but rather sampled from an assumed stationary age distribution typical of a west African coun-

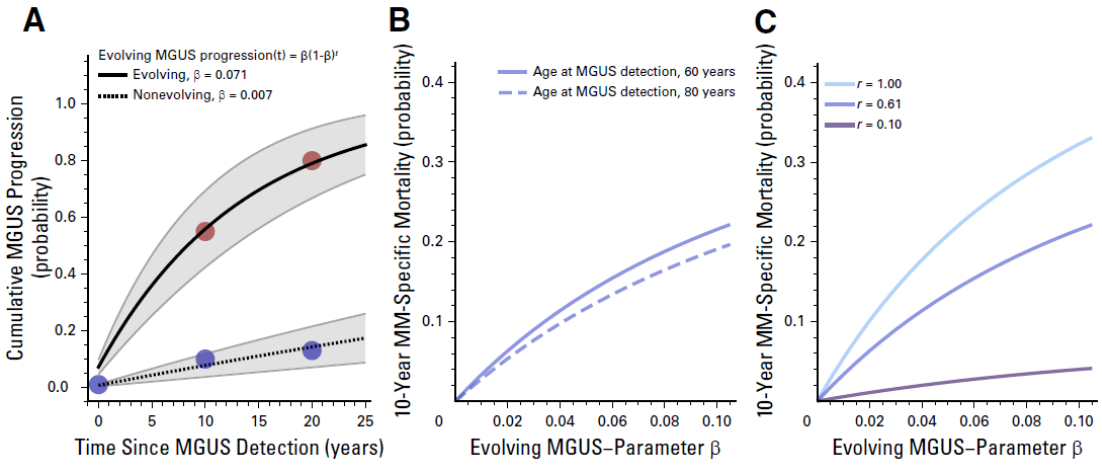

Figure 1.4: (A) Evolving MGUS progression rates fit to the functional form $\beta \times (1-\beta)^t$ using data from Rosiñol et al. (filled circles; nonevolving: 10% at 10 years, 13% at 20 years follow-up; evolving: 55% at 10 years, 80% at 20 years follow-up), and 95% CIs (shaded areas). A Nonevolving MGUS β value of (0.007; $R^2 = 0.996$) lends credibility to our constant progression risk p (Table 1.1). (B, C) Impacts of age at MGUS detection and progression risk reduction r on MM-specific mortality as a function of the evolving progression rate β where (B) $r = 0.61$ and (C) age at MGUS detection 60 years.

try, parameterized using data from Ghana[8], as this region was previously described by Landgren et al. and was found to be very similar to the African American population's lifetime risk of MGUS[48]. We used the 2013 immigration rate to the United States, which was roughly 1 million immigrants, and of those, about 1 in 10 are from Africa[1]. Tying this rate to the 2013 United States population yields an African immigration rate of 0.00032 legal immigrants per year per United States citizen. A multiplicative factor *imf*, where $imf = 1$ corresponds to the 2013 African immigration rate, was introduced to simulate scenarios with up to 50 times the current immigration rate from a region with a large fraction of MGUS high-risk individuals. Once the total number of African immigrants per year was computed based on the simulation population size, their ages were drawn from the Ghanaian age distribution in order to determine the

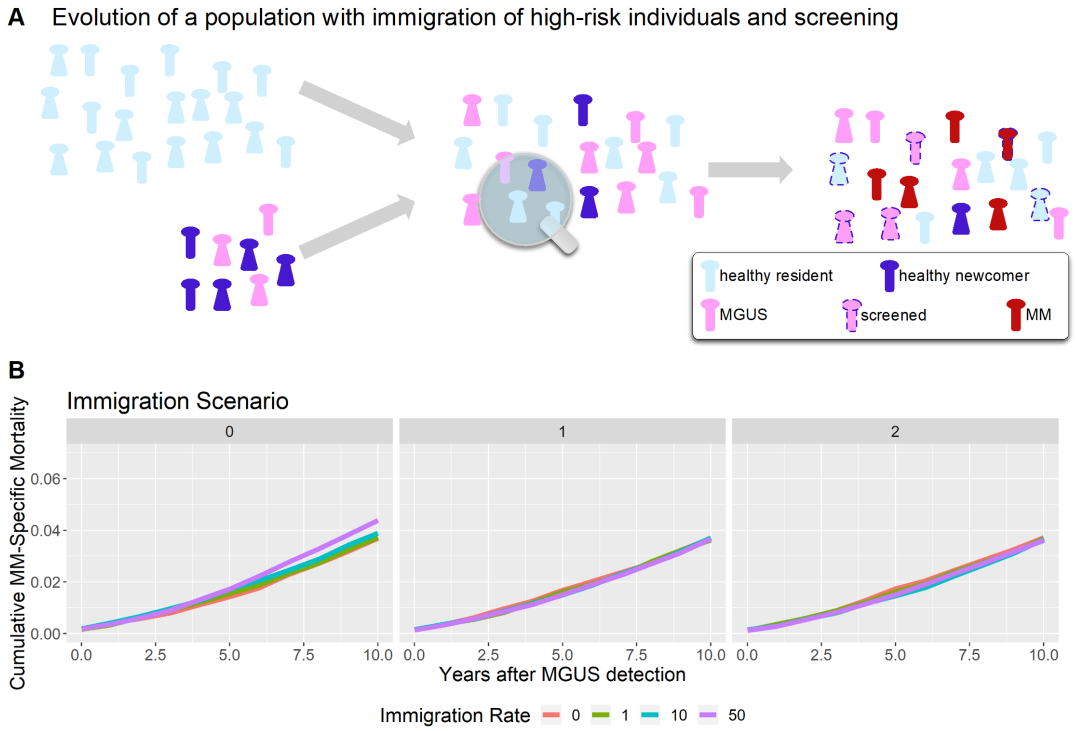

Figure 1.5: (A) Schematic model of immigration. (B) Cumulative MM-specific mortality following detection of MGUS for all age groups 50-70 in the entire population (residents and immigrants) for the three immigration scenarios (left to right). Stronger differences in mortality only became apparent when legal immigration was assumed to be 50 times higher than currently observed. The stationary fraction of high-risk African-Americans in the resident population was 20%.

age of each incoming immigrant. A portion was selected to arrive with MGUS according to age-specific MGUS prevalence rates, calculated via our analytical model to achieve the previously reported overall MGUS prevalence in the native Ghanaian population of 5.84%[48], using the high-risk African-American MGUS incidence rates.

Because Ghana-specific MGUS incidence rates are unknown, three scenarios were considered. In Scenario 0, immigrants experience death according to representative Ghanaian age-specific rates and MGUS incidence according to the U.S. high-risk rates. In Scenario 1,

immigrants experienced death rates according to the U.S. high-risk (i.e. African-American) death rates. In Scenario 2, we considered the extreme case where immigrants experienced death and MGUS incidence rates according to the U.S. low-risk population rates (non-African-Americans), both of which are lower than the high-risk or Ghanaian rates.

We analyzed how varying immigration rates affected the distribution of ages at MM diagnosis under the three immigration scenarios. To quantify the changes in this distribution with varying immigration rates, we used the KS test-statistic, comparing each distribution of ages at MM diagnosis under variable immigration rates ranging up to 50 times the 2013 immigration rate to the distribution without immigration. In cumulative MM-specific mortality, we observed meaningful mortality increases due to higher immigration rates only when immigrants followed their original region's death table and MGUS incidence rates, Scenario 0. Even then, the increase in cumulative MM-specific mortality was only appreciable for immigration rates around 50 times higher than the 2013 rate. For Scenarios 1 and 2 no such drastic increase was observed (Figure 1.5B). This observation suggests that, although immigration may change the overall distribution of age at MM diagnosis and cumulative MM-specific mortality for the United States, these changes are too minute to see meaningful changes in health outcomes at current immigration rates.

Equal MM Prevalence as a Criterion for Optimal Screening Frequency

As part of our investigation into the effects of screening for MGUS to reduce MM mortality, we sought to help health organizations or government entities identify ways in which to

implement screening procedures. In the scenario of limited funds and thus limited screening ability, one such criterion could be to distribute these screens between the high and low-risk populations with the goal of achieving equal fractions of MM in both populations, thereby eliminating the inherent increased mortality shown in the high-risk population. In this way, a fraction y of available screens could be given to the high-risk population with the remaining $1 - y$ fraction of screens given to the low-risk population, where, for some value y, the MM prevalence in both groups is equal.

In the absence of screening benefit, $r = 1$, no such point exists and all screening efforts should focus on members of the high-risk population (Figure 1.6A), $y = 1$. The current best widely applicable intervention of aspirin use, $r = 0.61$[14], also indicates that all screens should be given to the high-risk population $y = 1$. Lower values of r which could be the result of further intervention research could permit $y < 1$, for example, with $r = 0.1$, $y = 0.81$ suggesting that 81% of annual screens initiated at age 50 should be given to the high-risk population and the remaining 19% to the low-risk population to achieve equal MM prevalence in the two risk groups (Figure 1.6B).

Groups with Higher than Two-Fold Lifetime Risk

In addition to ethnic background, other factors have been determined to increase lifetime risk of developing MGUS, notably a prior family history of MM[69]. We reanalyzed the sensitivity of both MM prevalence and MM-specific mortality in these increased risk groups as a function of our screening parameters. Both risk reduction (r) and frequency of screenings (Δa)

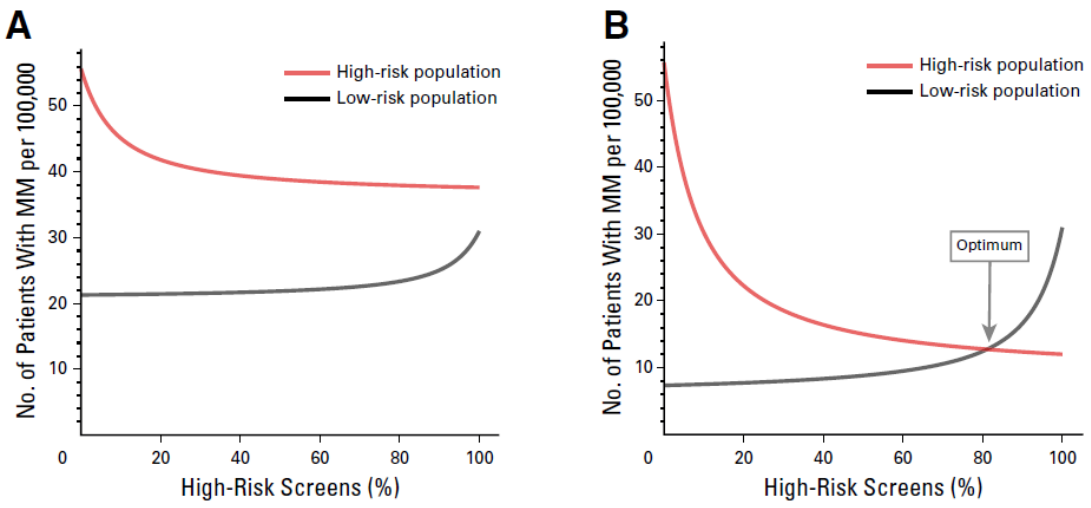

Figure 1.6: Using equal MM fractions as a criterion to distribute MGUS screening resources. (A, B) Comparing MM fractions in the high-risk and low-risk populations, with $a_0 = 50$ years and δa = 1 year, for different r. (A) For $r = 0.61$, the risk reduction factor associated with general aspirin usage, equality in MM fraction could not be observed for any percentage of screens distributed to the high-risk population. (B) For $r = 0.1$, equality was observed at approximately 81% high-risk screens. An optimal fraction of screens was defined as the point where the fractions of patients with MM in both subpopulations were the same.

have pronounced effects in these increased risk groups, but in those groups, steeper increases in mortality was found with decreasing screening frequency (Figure 1.7), suggesting that frequent screening is vital for those with increased risk of MGUS due to family history of MM.

1.4 Discussion

As no reliable cure for MM has been found, reducing mortality due to MM inevitably reduces down to reducing its prevalence[17]. Almost all patients who present with MM have progressed from a premalignant, asymptomatic stage known as MGUS[49]. Given that nearly perfectly sensitive and specific diagnostic tests exist for MGUS, early identification of individuals

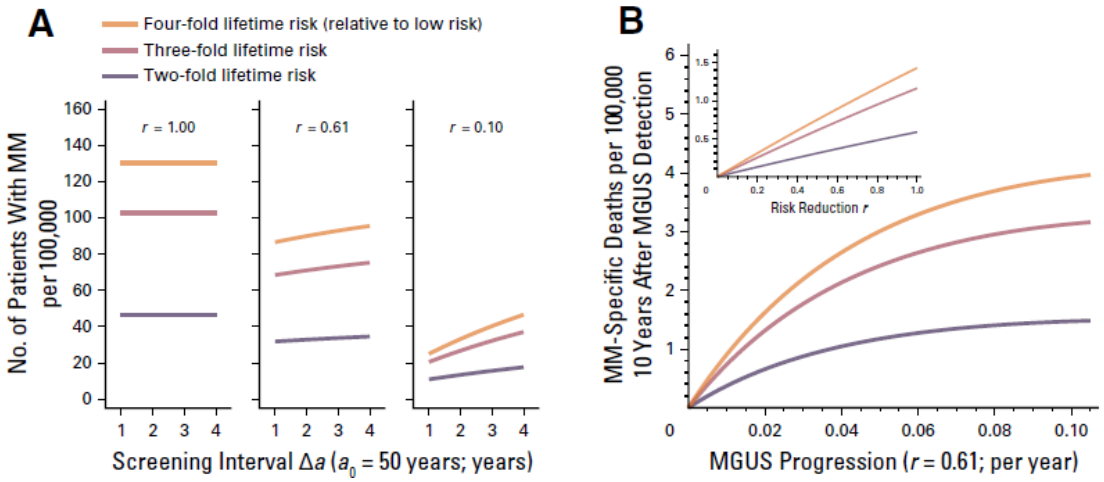

Figure 1.7: (A) The number of individuals with MM in risk groups with a lifetime risk higher than two-fold can be greatly influenced by the screening interval ($a_0 = 50$ years) and risk reduction factor r. (B) MM-specific deaths per 100,000 ($a_0 = 50$ years and δa = 1; age at MGUS detection, 60 years). The effects of both risk reduction and screening interval have more pronounced effects in groups with more than two-fold lifetime risk.

harboring MGUS could significantly contribute to reducing overall MM mortality by early intervention[39]. Because precise estimates of MGUS prevalence have changed over the past decade[46,44,45,43], we developed a framework to consider relative changes in prevalence. In this work, we evaluated various screening regimens including intervention efficacy, based on the current understanding that positive diagnosis of MGUS permits progression reduction via various interventions or modifiable risk factors such as aspirin, metformin, and exercise and diet changes[14,18,17,13,39].

As much of our current knowledge of MGUS comes from retrospective observational studies, any promise of early intervention in MGUS should be viewed cautiously. Our results do, however, suggest that research to identify better therapeutic agents to reduce the progression

from MGUS to MM are justified. Although not directly considered in this work, it will take considerable time to develop a more comprehensive understanding of the relationship between any possible therapeutic intervention and its potential adverse side effects on a wider scale than in preventing progression to MM. Patients positively screened for MGUS may experience similar psychological stress not dissimilar to that of an MM patient. The identification and disclosure therefore of any precancerous stage must accompanied by a discussion of how such a disclosure would affect the patient[20,56,54,38,55]. Efforts such as iStopMM[2], a long-term prospective three-armed randomized trial, are underway to evaluate MGUS screening with continuous follow-up preceding clinical manifestation of MM. Such efforts highlight the utility of predictive tools like ours to evaluate future therapeutic interventions.

The approach taken here allowed us to evaluate how effective an intervention must be in reducing the progression from MGUS to MM to result in specified reductions in MM prevalence and mortality, measured as fractions of their total values in the absence of screening. We used these measures of MM prevalence and mortality to compare screening regimens to avoid lead-time bias. In contrast, length-time bias was absent from most of our simulation studies by assumption, as we chose to characterize the progression from MGUS to MM as uniform across risk populations as well as across age groups, meaning someone who experiences MGUS in the high-risk population early in life will still progress to MM equally as quickly as someone who develops MGUS in the low-risk population relatively late in life. Thus, our study and its results are not confounded by these common epidemiological biases.

Using both a stochastic simulation framework as well as analytic tools, we measured MGUS

and MM prevalence as well as MM-specific mortality considering different risk-groups, varied screening regimens, and variation in progression risk after MGUS incidence and positive detection. Intuitively, to better reduce mortality due to MM, earlier and more frequent screening is needed. Additionally, improved interventions to effectively reduce the risk of progression from MGUS to MM would have a significant impact on both MM prevalence and mortality, as well as public health initiatives that may result. We found that these effects are even more important when considering individuals with evolving MGUS and those with a higher than two-fold lifetime risk of developing MGUS.

In this study we did not explicitly address screening toxicity nor did we choose to model smoldering MM, an intermediate stage between MGUS and MM that has been shown to have a much higher rate of progression to full MM at approximately 30% per individual per year, $p = 0.3$. We made this decision in part because it remains unclear whether or not smoldering MM is a requisite intermediate in the progression from MGUS to MM, but nonetheless, our framework can be easily adjusted to allow for this expansion.

Other efforts to assess screening and prevention in solid tumors such as prostate cancer have been controversial and overall lacking in positive evidence for screening in large prospective trials[73], especially the medicalization and treatment of asymptomatic conditions. While we share these skepticisms, the biology of MGUS and the robust laboratory screening tests demand careful evaluation of the role of screening and intervention. While there are noticeable similarities in the epidemiology of prostate cancer and MGUS, primarily that most low-grade lesions will not progress to advanced, lethal disease, major differences in the characteristics

of screening tests exist. For example, prostate-specific antigen tests for prostate cancer suffer from substantial false-positive and false negative rates (21-32% sensitivity, 85-91% specificity)[32]. On the other hand, the serum screening test for MGUS is straightforward, with a sensitivity close to 100% and specificity of 99%[37]. These differences cannot be ignored when considering the effectiveness and role of screening in MGUS and MM. We have shown that the reduction of the number of cases of MM as well as MM-specific mortality can be reduced in both high- and low-risk populations; however, to mitigate the effect of being predisposed to have a higher lifetime MGUS risk, a drastic reduction in progression risk is needed. Until such highly effective interventions are found, identification, screening, and possible intervention of high-risk individuals would be most effective in reducing MM mortality. As such agents are developed, our framework remains an applicable tool to assess the effect of screening in this area.

2

DIFFpop: Simulation of Differentiation Hierarchies

2.1 Introduction

Differentiation is a complex cellular process necessary for multicellular organisms to develop and maintain their tissue systems[58]. Cell populations throughout differentiation hierarchies

have been characterized by increased clonality driven by stochasticity and selection[36,71,10]. Branching processes are a class of stochastic processes that can be used to model the growth and composition of reproducing populations based on growth parameters specified for the individuals that compose those populations[31]. Branching processes are used to investigate the dynamics of cancer evolution and questions regarding pre-existing versus newly acquired resistance using high complexity barcoding libraries, in which each single cell is tagged with a unique genetic barcode[12]. Contrasting the growing populations in a branching process, a stochastic process model known as the Moran model describes populations of strictly constant size in which cell proliferation events are balanced by cell death events[57]. Simulation of complex processes such as cellular differentiation can be implemented using the Gillespie Stochastic Simulation Algorithm (SSA)[27].

DIFFpop uses the branching process, Moran process, and Gillespie Algorithm to simulate cellular differentiation, where each barcode or cellular clone and its progeny are tracked over time. The process instantiates all populations using user-specified proliferation, death and differentiation parameters. Throughout a simulation, cellular ancestry can be tracked in each population of the hierarchy using individual barcodes. To simulate exponentially growing populations, DIFFpop uses the direct Stochastic Simulation Algorithm[27] to advance the simulation by first determining the time until the next event followed by a stochastic choice of the type of event taking place. For fixed-size populations, DIFFpop simulates a multitype modified Moran model using tau-leaping[28] with the introduction of differentiation events, whereby events are coupled together to maintain fixed population sizes; for instance, a mitosis event

generating an additional cell is followed by a differentiation or apoptosis event to eliminate a cell.

Selection is introduced to the system by choosing cells for proliferation according to their fitness. During a mitosis event, one daughter cell may harbor a new mutation with a specified probability, giving rise to a new clone. In such situations, new clones are formed according to the infinite allele assumption[60]. This new cell inherits the fitness of its parent plus an additional change in fitness chosen from a user-defined probability distribution. As a default, fitness changes are drawn from a normal distribution such that the lower bound for the fitness of any clone is 0.

The flexible nature of the package allows the user to customize the process, easily change the underlying differentiation structure, parameters, and distributions, and achieve updated results. The hierarchical structure, population types and attributes, and event rates are specified using functions in R, allowing the user to quickly create multiple possible trees and implement simulations of each. Users may also vary the selective pressures at work in the cell populations by specifying population-level mutation rates and the distribution from which fitness changes of mutated cells are drawn. Setting the mutation probabilities to zero results in a process in which no new clones appear. Allowing for a positive mutation probability but setting the passenger probability, the probability that mutation does not affect a clone's fitness, to 1 simulates the infinite-allele process where mutations are recorded, but due to a lack of variability in fitness are selectively neutral[52]. After simulation initiation, no new barcodes are created, and therefore the maximum total number of barcodes is set at the initiation of the simulation,

allowing for the calculation of diversity indices to compare populations with different model parameterizations.

Our package is designed to work in tandem with experiments using cell labeling and barcoding in complex differentiation systems[72,16]. Results from simulations using DIFFpop can then be compared to experimental data to eliminate sets of parameters that result in findings not compatible with available data.

2.2 Software Description

Software design and class structures

DIFFpop is designed as an R package that interfaces with C++ for efficiency gains using Rcpp[24]. As such, the source code makes use of standard object-oriented programming concepts, defining several software classes. The software classes implemented in DIFFpop are described below and summarized in Table 2.1.

GrowingPop

A GrowingPop is the class used to designate the various cell types throughout a differentiation tree. A GrowingPop contains a list of cell states, functions to enact cellular events on those cell states, and event rates at which to perform those functions. The hierarchical structure is maintained by pointers to upstream and downstream cell populations. The GrowingPop class serves as the base class for both the FixedPop and DiffTriangle, whose own implementations

simply modify various member functions of the GrowingPop class.

FixedPop

A FixedPop is a class derived from a GrowingPop. In order to maintain a constant population size, cellular events are coupled together. For example, a mitosis event generating an additional cell is immediately followed by a differentiation or death event. Similarly, if the number of cells in the FixedPop population increases by one from upstream differentiation, a differentiation or death event of its own is immediately enacted to maintain the population level.

DiffTriangle

A DiffTriangle cell type is used to represent the downstream fully differentiated cells. Cells are arranged in a triangle formation. Cells enter the population on the highest level of the triangle, experiencing further differentiation and division to progress down the triangle. When a new cell enters the DiffTriangle population, it causes an already existing cell on the highest level to divide and further differentiate to the next level of the triangle. Those two cells each displace a existing cell, causing them to divide and differentiate, thus generating four newly displaced cells, which in turn displace cells that further displace cells until reaching the lowest level of the triangle. A displaced cell from the last row of the triangle can either be passed on to an offspring cell type if there are further cell types in the hierarchy or die out. Importantly, DiffTriangle structures will not initiate any cellular events of their own, as differentiation waves throughout a triangle are only initiated when receiving a new cell from an

Table 2.1: Software classes available in DIFFPop

Class Type	Population Size	Usage
Growing Pop	Dynamic	dynamically sized population with exponentially-distributed times between events
FixedPop	Constant	constant size homogeneous population
DiffTriangle	Constant	constant size population with z levels of maturation

upstream population.

The population size of a DiffTriangle is specified by two parameters; the first, z, is the number of cell divisions until full maturation or the number of levels in the triangle. The second is the *mfactor*, which is the number of cells that exist in the first stage of maturation, i.e. the number of cells in the first level of the DiffTriangle. If *mfactor* is greater than 1, then a cell entering the DiffTriangle population simply chooses at random which specific triangle to enter.

NodeList

The cells of each population are maintained by a NodeList. A NodeList is a doubly-linked list of nodes. Each node keeps track of a particular cell state, defined by the combination of a barcode, mutation status, and fitness value, as well as a count of how many cells belong to that particular state. In addition, DiffTriangle nodes also contain data to record in which triangle and at which level a cell resides. Mutation status is a string listing which mutations have occurred in that state. Information about particular mutations can be found in the mutation information file. The NodeList keeps track of the total number of cells in the list as well as the sum of the fitness values for the cells. This sum of fitness values is used to weight a

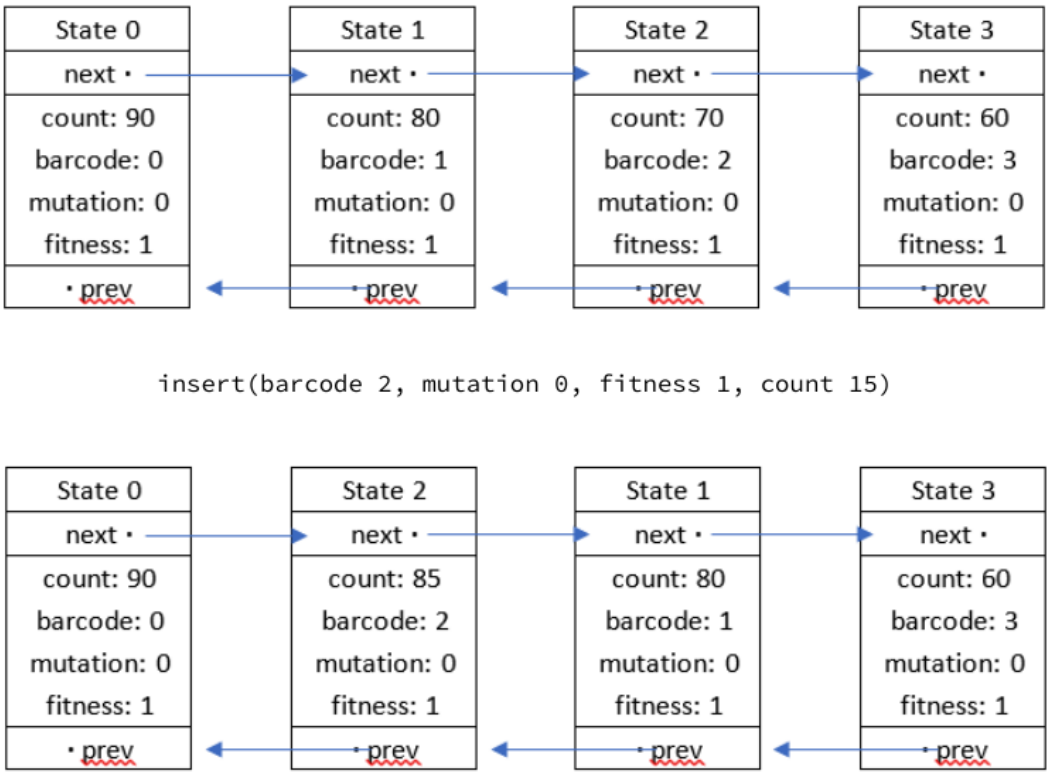

Figure 2.1: A NodeList maintains the order of states by count, allowing for faster indexing as dominant clones arise. This order is updated after every insertion or deletion from the list. Each node state consists of a unique combination of barcode, mutation history, and fitness. The number of cells in each state are kept as a count. Nodes are linked together to form a list using the pointers *next* and *prev*. Initially, the states are stored in order 0, 1, 2, and then 3. After the insertion of 15 cells to state 2, it is moved to the left of state 1 as the count of cells in state 2 is now greater than the count of cells in state 1.

uniform[0,1) random variable to select cells for mitosis by fitness. Individual cells may be inserted or removed from the list, maintaining a left-balanced list by cell count – that is, cell states with higher cell counts are found to the left of the list. Figure 2.1 shows an example of this left-balancing after the addition of cells to a particular state. This approach allows for more efficient indexing by cell count, particularly as diversity in the compartment decreases as dominant clones arise.

Table 2.2: Cellular events available in DIFFPop

Rate Parameter	Variable type	Description
α (alpha)	double	mitotic self-renewal rate (birth event)
β (beta)	double	asymmetric differentiation to downstream cell type
γ_1 (gamma1)	double	mitosis-independent (one-to-one) differentiation rate to downstream cell type
γ_2 (gamma2)	double	mitosis-dependent (one-to-two) differentiation rate to downstream cell type
ζ (zeta)	double	de-differentiation rate to upstream cell type
δ (delta)	double	apoptosis rate (death event)
μ (mu)	double	probability of mutation per mitotic event

Event types and parameters

Cellular events in DIFFpop are enacted according to their accompanying parameter rates given in units of number of events per cell per time unit. Figure 2.2 and Table 2.2 display the events implemented in DIFFPop, including the appropriate "type" parameter for the `addEdge` function. Note that population-specific parameters are specified using subscripts, i.e. $\alpha_{(LT-HSC)}$ is the mitotic self renewal rate for the LT-HSC population and $\gamma_{1(LT-HSC,ST-HSC)}$ is the mitosis-independent differentiation rate from the LT-HSC population to the ST-HSC population. These subscripts are occasionally simplified or dropped when the populations are understood.

Events can be classified into three categories based on how they affect the population size of the compartment: events resulting in a one-cell deficit include differentiation (γ_1/γ_2), de-differentiation (ζ), and apoptosis (δ); events that maintain the population size include asymmetric differentiation (β); events that result in a one-cell surplus include mitosis (α).

Mutations in DIFFpop occur only during mitosis events. Each mitosis event results in a

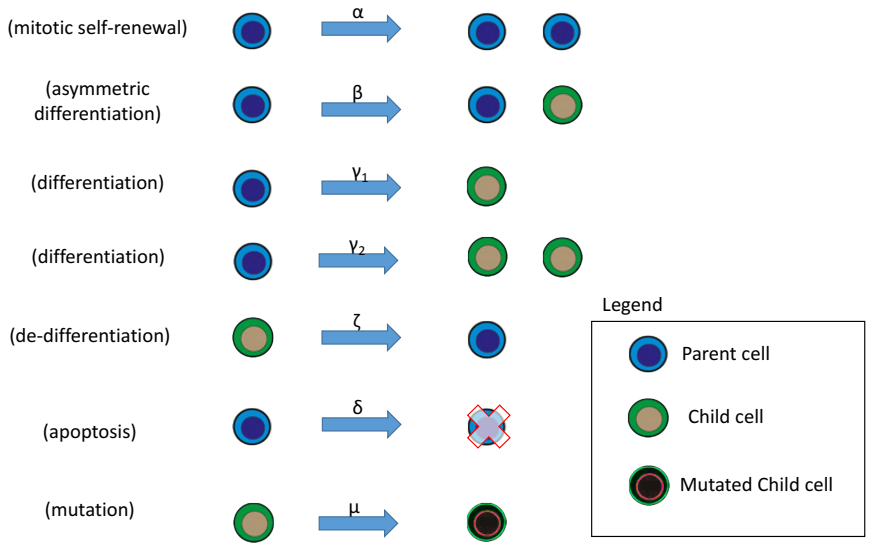

Figure 2.2: Events implemented in DIFFpop include apoptosis, asymmetric differentiation, mitosis-independent differentiation, mitosis-dependent differentiation, de-differentiation, apoptosis, and mutation. The lowercase Greek letter above each event describes the rate at which the event occurs in units of number of events per cell per time unit.

new mutation with probability μ. Therefore in a general population A, the rate of mitosis events resulting in a new mutation is $\alpha_{(A)}\mu_{(A)}$ and the rate of mitosis events resulting in no mutation is $\alpha_{(A)}(1-\mu_{(A)})$. Mutations accumulate according to the infinite allele assumption, that is, every new mutation leads to a new allele that has yet to be seen in the population.

In the FixedPop setting, all de-differentiation events result in an apoptosis event in the upstream population. This consideration is necessary to avoid circular equations when calculating adjustments to net proliferation in order to maintain a constant population size.

Maintaining a constant population size

In order to maintain a constant population size during Moran-process simulation using FixedPops and DiffTriangles, a relationship must exist between the event rates of the various cell types. Specifically, those event rates that result in an excess of cells in a compartment must be balanced with those event rates that result in a deficit of cells. Let $\alpha_{(x)}$ denote the mitotic self-renewal (α) rate of population x. Let $\gamma_{1(x,y)}$ denote the one-to-one differentiation (γ_1) rate from population x to population y. Let $\gamma_{2(x,y)}$ denote the one-to-two differentiation (γ_2) rate from population x to population y. Let $\beta_{(x,y)}$ denote the asymmetric differentiation (β) rate from population x to population y. Let $\zeta_{(x,y)}$ denote the one-to-one de-differentiation (ζ) rate from population x to population y. Let $\delta_{(x)}$ denote the cell death (δ) rate of population x. Let $n_{(x)}$ be the size of population x.

Then, for any compartment or population A, we have

$$n_{(A)}\alpha_{(A)} + \sum_{\text{pop } i \neq A} n_{(i)} \left(\gamma_{1(i,A)} + 2\gamma_{2(i,A)} + \beta_{(i,A)} + \zeta_{(i,A)}\right) = n_{(A)} \left[\delta_{(A)} + \sum_{\text{pop } i \neq A} \left(\gamma_{1(A,i)} + \gamma_{2(A,i)} + \zeta_{(A,i)}\right)\right] \tag{2.1}$$

and therefore, solving (2.1) for the death rate of population A,

$$\delta_{(A)} = \frac{n_{(A)} \left[\alpha_{(A)} - \sum_{\text{pop } i \neq A} \left(\gamma_{1(A,i)} + \gamma_{2(A,i)} + \zeta_{(A,i)}\right)\right] + \sum_{\text{pop } i \neq A} n_{(i)} \left(\gamma_{1(i,A)} + 2\gamma_{2(i,A)} + \beta_{(i,A)} + \zeta_{(i,A)}\right)}{n_{(A)}}. \tag{2.2}$$

For the population size of population A to remain constant, (2.1) must hold. That is, events that increase the population size (self-renewal and influx from other populations) must be balanced by events that decrease the population (cell death and differentiation). In the modified Moran Process, we force this to hold by automatically calculating delta, the death rate, for each population. If this calculated delta value is positive, we simply set the effective death rate equal to this value. If this calculated delta value is negative, we increase the self-renewal, alpha rate of the population by this value. Although DIFFpop uses (2.2) to adjust net proliferation, a user could use (2.1) to calculate the rate for any other event type given that the other event rates are known or can be estimated.

Table 2.3: Description of parameters used to specify the distribution from which changes in fitness are drawn upon mutation

Parameter	Variable type	Description
fitness_distribution	string	Random distribution to draw from. ["double-exp", "normal", "uniform"]
alpha_fitness	double	alpha parameter for the fitness distribution
beta_fitness	double	beta parameter for the fitness distribution
pass_prob	double	probability that mutation does not incur a change in fitness
upper_fitness	double	upper bound on fitness values
lower_fitness	double	lower bound on fitness values

Fitness distribution

Throughout the differentiation hierarchy, whenever a new clone arises due to mutation, a change in fitness can be drawn from a random distribution. The parameters of that distribution can be specified by the user in R. Table 3 shows the parameters used to specify the fitness distribution.

If the distribution function selected is normal, fitness additions are drawn from a $N(alpha\_fitness, beta\_fitness)$ distribution. If the distribution function selected is uniform, fitness additions are drawn from a $U(alpha\_fitness, beta\_fitness)$ distribution. If the distribution function selected is double exponential, alpha_fitness refers to the rate parameter of an exponential distribution for the positive range and beta_fitness refers to the rate parameter of an exponential distribution for the negative range.

2.3 Usage

Overview

The first step to utilize DIFFpop for simulation of a differentiation hierarchy is to specify the populations of the hierarchy. Populations of cells are created using functions that correspond to a specific software class, i.e. GrowingPop, FixedPop, or DiffTriangle. Users must give each population a unique name as well as an initial population size. Optionally, users may specify an initial cell barcoding frequency, which represents the proportion of initial cells that receive a unique barcode. If this parameter is not set, no unique barcodes will be created for the population.

The next step is to specify the transitions between populations. For that purpose, the `addEdge` function is used, along with the correct parameters: the initiating population, the receiving population, event type as a string (either "alpha", "beta", "gamma1", "gamma2", "delta", "zeta", or "mu"), and event rate. For events involving only one population ("alpha", "delta", or "mu"), users set that population as both the initiating and receiving population.

The last specification step is to specify which population is the root of the differentiation hierarchy, that is, which population is the furthest upstream, using the `setRoot` function in R.

The `simulateTree` function is then used to initiate the simulation.

Table 2.4: Description of simulation parameters

Parameter	Variable Type	Description
fixed	boolean	TRUE, if simulating using FixedPops and/or DiffTriangles; FALSE, if simulating using GrowingPop
time	integer	number of time units to simulate
census	integer	time interval to output full population census
indir	string	directory location for input files
outdir	string	directory location to write output files
seed	numeric	random number generator seed (optional)

Simulation parameters

Table 2.4 describes the parameters of the simulation. Observations are made and output files updated at every integer time unit through *nObs*. In addition, full outputs of the cell states in each population are made every *census* time unit(s). The *indir* directory informs the C++ backend where the input files for the differentiation hierarchy are located and *outdir* specifies a particular directory in which to place all output files. Optionally, the user can specify a numeric *seed* for the GSL random number generator used throughout the simulation.

Birth-Death Example

Towards learning how to utilize DIFFpop to simulate cellular differentiation, we present the following birth-death process. Using this simple example as a starting-off point, we then show how by the addition of relatively few lines, this model can be modified and expanded.

The following script creates the example:

```
library(diffpop)

# Create an empty DiffTree object
tree1 = DiffTree()

# Create a population "pop1" with 100 unlabeled cells
GrowingPop(tree = tree1, name = "pop1", size = 100, label = 0.0)

# Add cell birth event to pop1
addEdge(tree = tree1, parent = "pop1", child = "pop1", type = "alpha",
        rate = 0.5)
# Add cell death event to pop1
addEdge(tree = tree1, parent = "pop1", child = "pop1", type = "delta",
        rate = 0.3)

# Set the root of tree1 to "pop1"
setRoot(tree = tree1, popName = "pop1")

# Simulate tree1 for 10 time units writing results to ./output/
simulateTree(tree = tree1, fixed = FALSE, time = 20,
        1indir = "./input/", outdir = "./output/")
```

After installing DIFFpop using the steps outlined above, the library must first be loaded into the current R session. In order to begin creating a differentiation hierarchy, we must create an empty DiffTree object we call "tree1". We then begin by creating a single population named "pop1", and initializing it to contain 100 unlabeled cells, and add it to the tree. Because we want to model a population whose size fluctuates over time, we use a GrowingPop to model this population.

We then add transitions to this population. We start by adding a self-renewal event, which occurs at a rate of 0.5 events per cell per time unit. We also add the death event, which occurs at a rate of 0.3 events per cell per time unit.

The last remaining steps are to set "pop1" as the root of the differentiation tree and start our simulation. We simulate this tree for 20 units of time, writing all of the inputs to the input folder and all output files to the output folder. We also set the simulation parameter fixed to FALSE, notifying DIFFpop we are modeling the system using GrowingPops.

Before expanding our basic process, let us take a look at some of the output files from this simulation. Each simulation is given a unique file prefix. For example, the output files from the last run were all prefixed with "out_11-11-2018-202313_54045", letting us know the simulation was initiated at 8:23 PM on November 11, 2018. Because we had completely unlabeled cells and did not allow for mutation, the only system statistics of interest are the population sizes, which are output to the "out_11-11-2018-202313_54045_pop.csv" file each time unit. The resulting plot is shown in Figure 2.3. The following code creates the plot:

```
popfiles = list.files("./output/", pattern="^out.*_pop.csv$", full.names=T)

# Read in population sizes file
pop = read.csv(popfiles[length(popfiles)])

# Plot the population size of "pop1" vs. simulation
plot(pop$time, pop$pop1, xlab = "Time", ylab = "Population Size",
        main = "Simple Birth-Death Process")
```

Extending the Birth-Death Example

In our next example, let us look at exploring some other features of DIFFpop beyond a simple birth-death process by labeling 50% of cells in "pop1", adding an additional population

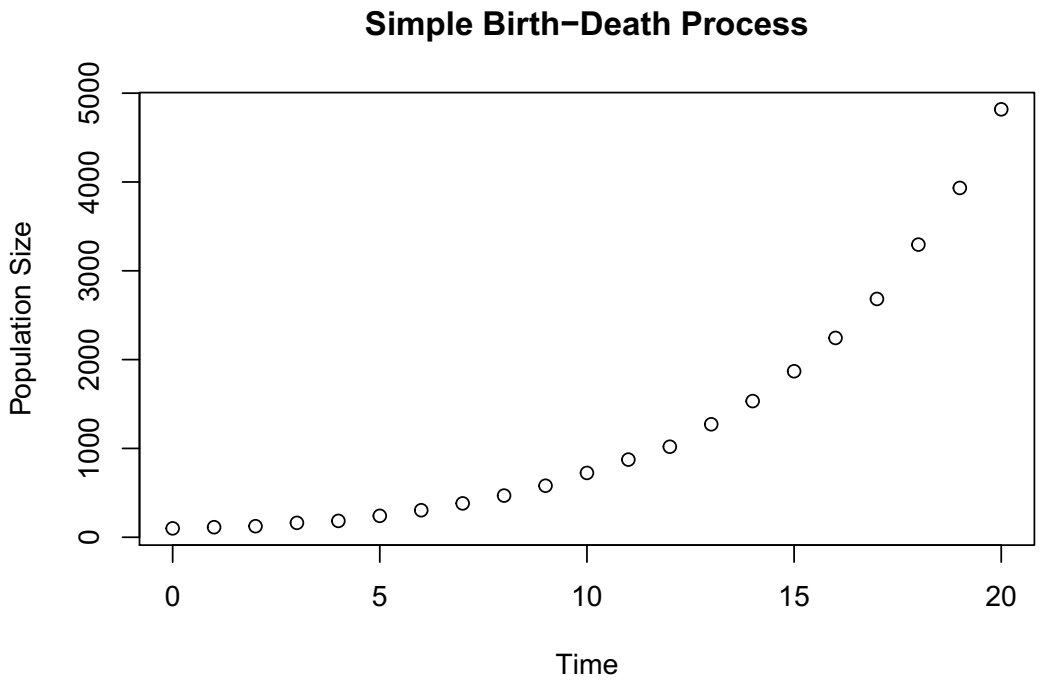

Figure 2.3: Simple birth-death process population across the course of 20 time units, $birth = 0.5$ and $death = 0.3$. Event rates in units of number of events per individual per time unit.

"pop2", adding differentiation from "pop1" to "pop2", and allowing for mutations to occur in "pop1". We store this updated tree as "tree2". Whenever a mutation occurs, we introduce a new fitness change to be drawn from a standard normal distribution. The necessary R code to simulate this model is shown below:

```
library(diffpop)

# Create an empty DiffTree object
tree2 = DiffTree()

# Create two populations, labeling on average 50% of cells in pop1
GrowingPop(tree = tree2, name = "pop1", size = 100, label = 0.50)
GrowingPop(tree = tree2, name = "pop2", size = 50, label = 0.0)

# Add cell birth/death events to pop1
addEdge(tree = tree2, parent = "pop1", child = "pop1", type = "alpha",
        rate = 0.4)
addEdge(tree = tree2, parent = "pop1", child = "pop1", type = "delta",
        rate = 0.3)

# Add cell birth/death event to pop2
addEdge(tree = tree2, parent = "pop2", child = "pop2", type = "alpha",
        rate = 0.35)
addEdge(tree = tree2, parent = "pop2", child = "pop2", type = "delta",
        rate = 0.4)

# Add differentiation from pop1 to pop2
addEdge(tree = tree2, parent = "pop1", child = "pop2", type = "gamma1",
        rate = 0.05)

# Add mutation in pop1, occurs during mitosis event with probability 1e-4
addEdge(tree = tree2, parent = "pop1", child = "pop1", type = "mu",
        rate = 1e-4)

# Set fitness change distribution when a new mutant arises
setFitnessDistribution(tree = tree2,
```

```
        distribution = "normal",
        alpha_fitness = 0,
        beta_fitness = 1,
        pass_prob = 0,
        upper_fitness = NA,
        lower_fitness = 0)

# Set the root of tree1 to "pop1"
setRoot(tree = tree2, popName = "pop1")

# Simulate tree1 for 50 time units writing results to ./output2/
simulateTree(tree = tree2, fixed = FALSE, time = 100,
indir = "./input2/",  outdir = "./output2/")
```

The population sizes and label frequencies for the two populations across the course of simulation time are shown in Figure 2.4. For ”pop1”, we expect the population to grow, as the self-renewal rate is greater than the sum of the death rate and differentiation rate downstream to ”pop2”. Although the net proliferative ability of ”pop2” is negative, that is its death rate exceeds its self-renewal rate, we observe growth in ”pop2” due to influx from the upstream ”pop1” cells differentiating (Figure 2.4A). We initially labeled 50% of the ”pop1” cells and 0% of the ”pop2” cells. As the simulation progresses, the label is gradually taken up by the ”pop2” cells as labeled cells from ”pop1” undergo differentiation. Towards the end of the simulation, we see a decline in the label frequencies in both populations, due to an unlabeled cell undergoing a mutation to become more fit than its neighbors, dominating the populations (Figure 2.4B). For more detailed usage examples, please see Appendix B.2 and B.3.

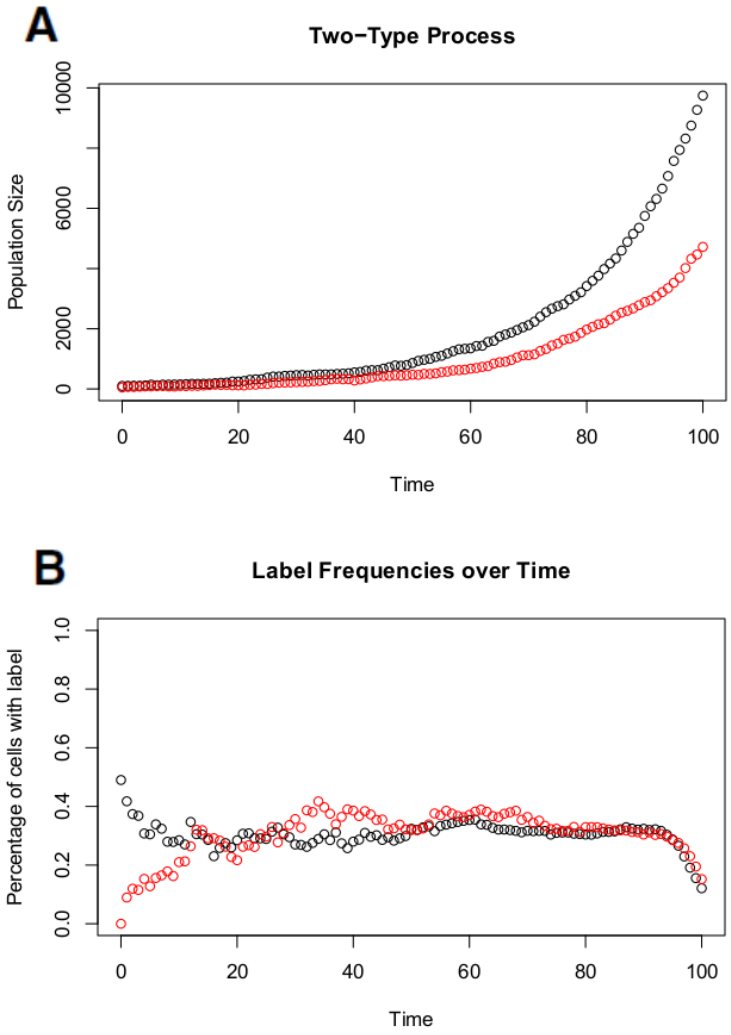

Figure 2.4: (A) Population sizes for two cell type model with differentiation from pop1 (black points) to pop2 (red points). Each cell type has its own cell birth and death events. Differentiation from pop1 to pop2 proceeds via mitosis-independent differentiation, that is, one cell from pop1 differentiates without division to one cell of pop2. (B) Label frequencies over time for pop1 (black dots) and pop2 (red dots). The initial frequency of the binary label was 50% in pop1 and 0% in pop2. As simulation progresses, the label is taken up by the pop2 population as cells from pop1 differentiate to pop2.

2.4 Applications

Our software package uses simulations to explore and test hypotheses in tandem with experimental barcoding or labeling data. The simulation outputs include statistics related to the population size, barcode diversity, event rates, mutation events, and the fraction of labeled cells. Additionally, the user can specify how often to output a census of the entire system to longitudinally track clonal dynamics throughout the hierarchy. Users may then draw repeated samples from this population census to compare against data generated from single cell barcoding or cell labeling experiments. In the next sections, we describe several possible applications of DIFFpop to various experimental procedures.

Binary Cell Labeling

DIFFpop is capable of tracking the uptake and progression of a binary label throughout a differentiation hierarchy. In experimental settings, these are often fluorescent labels that allow the researchers to sort samples into cell types and then quantify proportion of cells that express the label.

Simulating a binary labeling scheme in DIFFpop can be achieved in two ways. In the first, the user manually enters the number of initial cells that belong to the unlabeled and labeled populations. In the second, the user specifies with what probability a cell will gain the label upon simulation initiation. The proportion of labeled cells can be tracked over time in the label and census output file.

As an example of applying DIFFpop to this type of data, we have included a vignette replicating the results from introducing a yellow fluorescent protein (YFP) reporter into the hematopoietic cells of the bone marrow[16]. After validating that our stochastic simulations closely match the results from the in-vivo experiments, we could further investigate the system using DIFFpop, including inferring hematopoietic clonal dynamics from the system if the investigators introduced unique barcode labeling.

Confetti-style Labeling

A confetti-style labeling scheme is one in which one particular label from amongst a series of possible labels is expressed through random segregation and reintegration into the host cells genome. As an example system, 4 possible colored reports, labeled green, blue, yellow, and red, are added side-by-side in the host cell's genome. Upon induction by CRE recombinase, these label sections are random spliced out of the genome and reintegrated, with ultimately only one label color being expressed. This same label is expressed in all daughter cells and can be traced as cells replicate and differentiate.

Simulating a confetti-style labeling scheme can be achieved in DIFFpop by simply specifying the initial number of cells to express each particular label. The census files can then be analyzed upon simulation completion to track the changes in label expression throughout the system over time.

As an example of this type of experimental procedure, we point the reader to a confetti-style labeling scheme implemented in the hematopoietic system of mice[26]. Such a labeling

experiment could be easily simulated using DIFFpop assuming the proper population sizes and transition rates were known.

Unique Cell Barcoding

In addition to simulating fluorescent cell labels, DIFFpop can also be used in combination with unique cell barcoding experiments. Unique cell labeling can be achieved by introducing a mobile transposon into the genome. Upon induction of labeling with tamoxifen, this transposon is spliced from the genome, and randomly reintegrated at some point in the host genome. Assuming the probability that the transposon randomly integrating into the same location in two cells is negligible, each cell now contains the transposon in a unique genomic position. The transposon in this location will then be passed to all offspring cells and be maintained through replication and differentiation events. At the end of an experiment, cell populations can be sorted and then sequenced for the presence or absence of these barcodes. Alternatively, a sample of cells can be sent off for single cell sequencing, allowing for not only the presence or absence of a particular barcode, but also an estimate of the size of a particular barcode-defined clone.

Simulating unique cell barcoding in DIFFpop can easily be achieved by simply specifying the proportion of cells to be successfully labeling upon system initialization. The census files can then be analyzed upon simulation completion to track the barcode frequencies in the system over time. One can even simulate a single cell barcoding experiment by randomly sampling from the barcode population.

As an example of a unique barcoding population, we point the reader to an experimental procedure in which cells of the hematopoietic system are labeled in-vivo and analyzed in native hematopoiesis, not requiring the use of cell transplantation[65].

Example Application

To illustrate a possible application to experimental data, DIFFpop simulations were run for a mouse model of the hematopoietic system in which a fraction of cells contain a fluorescent protein label. Parameters for the model were determined using the data and methods from a previous study[16]. Using DIFFpop, we performed 1,000 simulations of the model (Figure 2.5A) and recorded the median trajectory along with 25th and 75th quantile confidence bands for each cell population along with the experimental data from the mouse model (Figure 2.5B). We found that the simulated trajectories demonstrated good agreement with experimental results, including for data points from older mice that were not used in the determination of the simulation parameters. In addition to comparing model results to experimental data, other features of DIFFpop, such as simulations including barcoded cells, can be used to investigate cellular diversity in the hematopoietic system over time (Figure 2.5C). For full application code, see Appendix B.1.

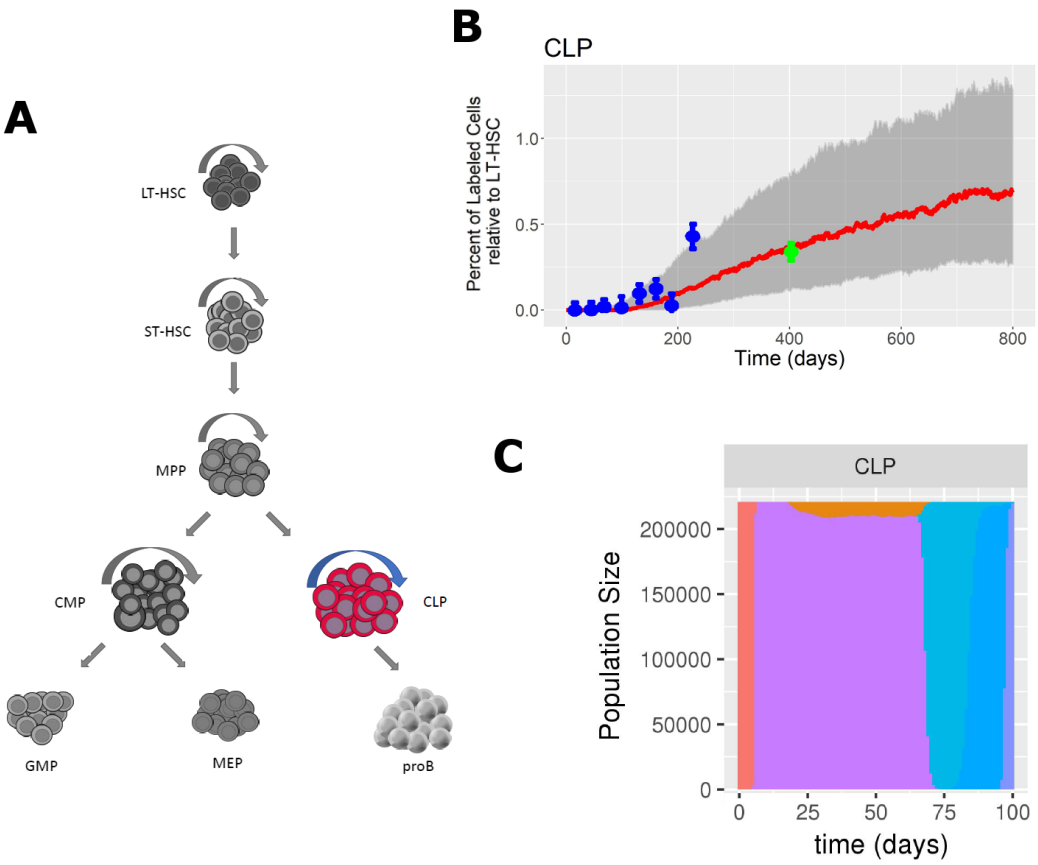

Figure 2.5: Visualization of DIFFpop outputs. (A) A schematic representation of the hematopoietic system. The common lymphoid progenitor (CLP) population is the initial population in the lymphoid branch of the hematopoietic system and the focus of panels B and C. Abbreviations: Long-term hematopoietic stem cell (LT), short-term hematopoietic stem cell (ST), multi-potent progenitor (MPP), common myeloid progenitor (CMP), common lymphoid progenitor (CLP), granulocyte-macrophage progenitor (GMP), megakaryocyte-erythroid progenitor (MEP), pro-B cell (proB). (B) Experimental label progression results from Busch et al. (blue points) and DIFFpop simulated trajectories (red lines, median trajectory; grey bands, 25th and 75th percentiles) for the CLP population. Experimental data points from beyond 400 days (green points) were not used during parameter estimation but are correctly predicted using simulated results. (C) Bar plot of clone sizes denoted by different colors over the first 100 days of simulation of the CLP population.

2.5 Conclusion

DIFFpop simulates cellular differentiation including single cell barcoding and mutation acquisition under the infinite-allele assumption, tracking evolutionary dynamics and other model outputs. Estimation methods for complex differentiation systems, including multitype branching processes and Moran models, quickly become intractable as the model complexity increases. Simulation methods such as DIFFpop provide an alternative method for investigation of these systems and can be performed quickly on a cluster.

3

ESTIpop: Estimation of Continuous-time Markov Branching Processes

3.1 Introduction

Understanding the effects of stochastic systems has played a central role in many fields throughout the last century. Branching processes, a subclass of stochastic processes, have been used

extensively to model the growth and composition of reproducing populations[31]. Parameterized by rates that specify the growth and death of and transition between various cell types, branching processes can model complex cellular systems and hierarchies[23]. Branching process models have been employed to investigate the evolutionary dynamics of cancer, where differences in the fitness of cells, mutations conferring resistance or other traits, and competition between clones and cell types affect the trajectory of a population of cells[53]. Recently, branching processes were used to investigate the dynamics of pre-existing versus newly acquired resistance using high complexity barcoding libraries, in which each single cell is tagged with a unique genetic barcode at the beginning of an experiment[12]. In this way, each barcoded cell can be viewed as an ancestor to its own branching process whose progeny can then be traced over time and across samples.

A continuous-time branching process describes a system of independently reproducing individuals belonging to various types who live for an exponentially distributed, type-specific lifetime before generating offspring according to a type-specific distribution. Branching processes can be used to model common biological phenomena such as cell division, cell death, mutation, and differentiation using various types and offspring distributions.

ESTIpop is an R package designed to work in tandem with experimental data to estimate the rate parameters and simulate continuous-time branching processes. Based on the Central Limit Theorem (CLT) and due to the fact that individuals in a branching process are independent from each other, a multitype branching process can be viewed as a sum of processes initiating with a number of independent ancestors of various types. As this number of ancestors

of a single type tends to infinity, the number of individuals of all types present at any specific time is approximately normally distributed[79]. In Appendix C.1 we show that the CLT holds for ancestors of different types. This is a noteworthy contribution, as there are cases in which the ancestral population is composed of various types, such as preexisting drug-resistant clones present in a tumor of otherwise sensitive cells. The mean and variance of the asymptotic distribution can be calculated using the properties of branching processes and are ultimately functions of the lifetime parameters and offspring distributions. Thus, the asymptotic likelihood is a function of the data and the counts of each cell type, which can then be maximized over the parameter space to estimate the rates at which the model events – such as birth and death – occur. As branching process models increase in complexity, analytical approaches become intractable and research on inference methods for branching processes are ongoing, even for seemingly simple processes such as the linear birth-death process[74]. The use of an asymptotic likelihood function with basic assumptions provides a method to perform estimation in these scenarios.

Simulation in ESTIpop is implemented using both exact and approximate methods. In addition to fixed transitions (ex. a cell division event giving rise to two daughter cells, mutation, death, etc.), ESTIpop allows the user to specify reproduction distributions from standard distributions, such as the Poisson distribution. Such reactions may prove useful for modeling viral dynamics, in which one infected cell is capable of infecting a random number of new cells. A further extension of the simulation capabilities of ESTIpop is the addition of time-dependent rates, such as a cellular growth rate modulated according to the circadian rhythm or

time-dependent concentrations of a drug. This is achieved by adapting the Gillespie Stochastic Simulation Algorithm (SSA)[27] by use of adaptive thinning[51]. Simulation in ESTIpop can proceed in either an exact manner, using the Gillespie SSA, or as an approximation using the asymptotic distribution derived in Appendix C.1 for significantly faster speed. Although Gillespie's SSA returns exact simulation results, it suffers from the drawback that for large systems or for large simulation times, it becomes computationally expensive. To improve upon this issue, based on the asymptotic distribution derived from the Central Limit Theorem, ESTIpop provides methods for approximate simulations with significantly improved simulation speed. Results regarding execution times are provided in Appendix C.2.

3.2 Software Description

ESTIpop is designed as an R package that interfaces with C++ for efficiency gains using Rcpp[24]. As such, the source code makes use of standard object-oriented programming concepts and defines several software classes. The software classes implemented in ESTIpop are described below.

TransitionList

A TransitionList is used to specify the structure of the model by listing the transitions that can occur between the various types. Each transition consists of a parent population, rate, and update vector, although for estimation, a rate is not required to be supplied. The parent

population is the population that initiates the transition. Populations should be named using 0-indexed integers. The rate is specified in terms of the number of events per individual of the parent type per unit of time and can be constant throughout the course of the simulation. The update vector is a k-length vector that is added to the system when the transition is enacted after removing one individual from the parent population enacting that transition. The update vector may be either fixed or random in nature, demanding the use of either the FixedTransition object or RandomTransition object (see FixedTransition and RandomTransition).

FixedTransition

A FixedTransition is a transition in which the update vector is the fixed across multiple enactments of the same transition. As an example, the death and removal of an individual can be represented by using the 0 vector. Whenever a transition is selected to be enacted, an individual from the parent population is first removed from the system. This is why the vector of 0s represents a death event in any type. As another example, in a two-type process, the update vector $(2, 0)$ enacted from parent population 0 would represent a net increase of 1 in the 0 population. A single enactment of this FixedTransition is demonstrated in Figure 3.1. Table 3.1 displays the parameters for a FixedTransition.

RandomTransition

A RandomTransition is a transition in which the update vector is determined by draws from a random distribution. First, a total number of offspring is drawn from the `oDist` distribution

Table 3.1: Parameters for a FixedTransition

Parameter	Variable Type	Description
parent	integer	specifies which population is capable of enacting the transition; populations are named using zero-indexed integers
rate	numeric or Rate object	specifies the rate per individual per unit time that the transition occurs
fixed	integer vector	a k-length vector that is added to the system

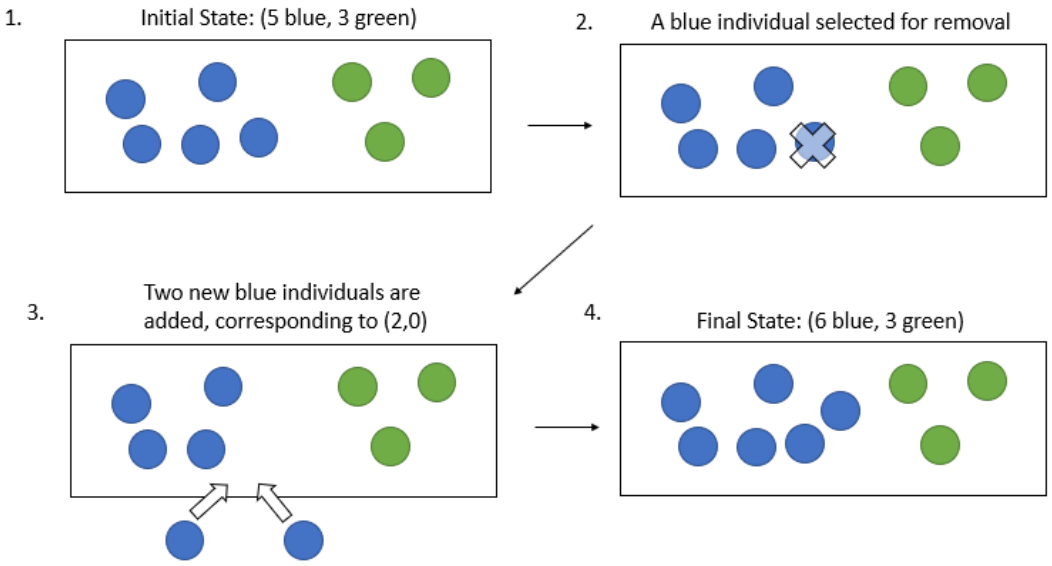

Figure 3.1: Example of FixedTransition(parent = 0 (blue), fixed = c(2, 0)). The blue circles represent individuals of type 0 and the green circles represent individuals of type 1.

Table 3.2: Parameters for a RandomTransition

Parameter	Variable Type	Description
parent	integer	specifies which population is capable of enacting the transition; populations are named using zero-indexed integers
rate	numeric or Rate object	specifies the rate per individual per unit time that the transition occurs
oDist	string	distribution of the total number of offspring ["poisson"]
oParams	numeric vector	parameters for the oDist offspring distribution
oVec	numeric vector	a k length vector that specifies the probabilities according to which the total offspring will be distributed amongst the types

with given parameters `oParams`. Then, this number of offspring is distributed to the various types of the system according to a multinomial distribution with probabilities proportional to the `oVec` vector. The parameters for a RandomTransition are shown in Table 3.2.

Rates

ESTIpop is capable of simulating general multitype branching processes with both constant transition rates as well as time-dependent transition rates. This can be done by specifying a Rate object as a part of a FixedTransition or RandomTransition. If only a numeric value is given, a constant rate is assumed. To ease the burden of using time-dependent rates, we have provided various templates for common time-dependent structures, although a completely custom function of time may also be used. If a time-dependent rate is used, the appropriate simulation function is `branchTD`. The parameters for a Rate are shown in Table 3.3.

Table 3.3: Parameters for a Rate

Parameter	Variable Type	Description
type	integer	specifies which rate template is being used (see)
params	vector	specifies the parameters for the rate template (see)

Table 3.4: Available Rate Templates in ESTIpop

type	Description	Parameters
0	Constant Rate	(i) rate per individual per unit time, constant across simulation time
1	Linear Rate	(i) intercept and (ii) slope of rate function across simulation time
2	Switch Rate	(i) pre-switch rate, (ii) post-switch rate, (iii) and time of switch for rate across simulation time
3	Custom Rate	(i) path to .dll file and (ii) function name within .dll

RATE TEMPLATES Rate templates have been included in ESTIpop to provide access to common time-dependent rates. Each is specific using the `type` parameter of a Rate and the accompanying parameters. As an example, the `Rate(type = 1, params = c(0, 1))` template will create a rate that starts at 0 at time 0 and increases linearly in time with slope 1. The templates and their parameters are shown in Table 3.4.

STOPLIST

Similar to a TransitionList, a StopList is a collection of StopCriteria. An alternative to simply simulating a process for a set length in time, a StopCriterion specifies a certain condition upon which the simulation should halt, such as a particular type reaching a set threshold. After each elementary step in the simulation, all of the specified StopCriteria in the StopList are checked

Table 3.5: Parameters for a StopCriterion

Parameter	Variable Type	Description
indices	integer vector	specifies the set of populations whose sizes should be added; specified using 0-indexed populations
inequality	string	one of "<", ">", "<=", ">=", or "="
value	numeric	specifies the value to which the specified system population sizes is compared

and if any StopCriterion is met, the simulation stops.

StopCriteria

A StopCriterion is composed of three elements: a set of indices, an inequality, and a value for comparison. During evaluation, the size of each type specified in the set of indices are added together and then compared against the value using the specified inequality. If the statement evaluates true, then the simulation halts. As an example, a `StopCriterion(indices = c(0,1), inequality = ">", value = 5000)` will halt the simulation once the combined population sizes of type 0 and type 1 exceed $5,000$. The parameters for a StopCriterion are shown in Table 3.5.

Likelihood Functions

The `estimateBP` function will perform optimization on the log-likelihood function derived from the Central Limit Theorem applied to general multitype branching processes. For more information on the derivation of the likelihood function, see C.1. By default, the optimizer

Table 3.6: Parameters for `bploglikelihood`

Parameter	Usage
data	A matrix of the type counts, with columns defining the types and each row defining an observation from a particular time point
time	a numeric or vector of timepoints from which the observed data were collected
N	a *k*-length vector or matrix of the initial population counts at time 0. If a matrix, each row specifies the initial conditions for the same row in the data matrix
transitionList	A TransitionList object to specify the model form. See TransitionList.

will use the "L-BFGS-B" method[82] with lower bounds on the rate estimates around 1e-10 and upper bounds around 4, which have provided good results for estimating rate parameters for biological processes given in units of number of events per individual per day. If the user wishes to more finely tune the optimization scheme or set differing bounds, we also provide access to the log-likelihood functions, which may be maximized over the rate parameter space to provide a maximum likelihood estimate. The likelihood function, `bploglikelihood` has parameters listed in Table 3.6.

Additional Parameters

In addition to the software classes and functions described in the previous sections, ESTIpop also requires users to specify other quantities for either estimation or simulation. These additional parameters are described in the following subsections.

Data

Data from which to estimate the rate parameters is supplied to the estimation functions as a matrix, where each row is a particular observation of the type counts, which are defined by the columns. Data is provided as an $N \times k$ matrix for N observations of a k-type system.

Time

For estimation, time is a numeric or N-length vector of time points for each data observation under the assumption that the process was initiated at time 0 with the specified initial vector (see Initial Population Vector). If all observations come from the same time point, a single numeric may be used.

For simulation, time is the number of time units for which a simulated trajectory is run. A simulation initiates at time 0 and outputs the system population counts at each integer time until the end time is reached.

Initial Population Vector (N)

The initial population vector (N) is a k-length vector of the type counts at time 0. This must be supplied for both estimation and simulation.

Initial Estimate

When estimating rate parameters using optimization, the user must specify an initial estimate for each rate parameter in the model. This is provided to ESTIpop in the form of a numeric vector.

Known Parameters

It can sometimes be the case that some parameters have been previously characterized either by experimentation or literature results. In those cases, it might be necessary to fix a rate parameter to a particular value. This can be accomplished by using the `known` parameter, which is a boolean vector the same length as the TransitionList object, where `TRUE` designates that the parameter is fixed at the initial estimate value and will not be estimated and, `FALSE` designates that the rate parameter is to be estimated.

3.3 Usage

Estimation Overview

General estimation of the rate parameters for continuous-time Markov branching processes in ESTIpop can be performed using the `estimateBP` function with parameters described in the next sections. If more specific optimization parameters are needed, we also provide different forms of log-likelihood functions, which can be used with any standard optimizer to find a

Table 3.7: Parameters for `estimateBP`

Parameter	Usage
data	A matrix of the type counts, with columns defining the types and each row defining an observation from a particular time point
time	a numeric or vector of timepoints from which the observed data were collected
N	a k-length vector or matrix of the initial population counts at time 0. If a matrix, each row specifies the initial conditions for the same row in the data matrix
transitionList	A TransitionList object to specify the model form. See TransitionList.
initial	vector of initial rate parameter estimates

maximum likelihood estimator for the rate parameters. The required parameters for estimation using `estimateBP` are shown in Table 3.7.

Birth-Death Example

As an introduction to estimation via ESTIpop, let us start with the one-type birth-death model shown in Figure 3.2. In this model, a population of a single type experiences birth events, in which an individual from the population is chosen to replicate, and death events, in which an individual from the population is chosen for removal. To test our estimation procedure, we begin by simulating data using functions available in ESTIpop. We initiate the population with size 100 and allow it expand for 5 units of time with birth parameter 1 and death parameter 0.7. Using the following code, we generate 1,000 samples from this process.

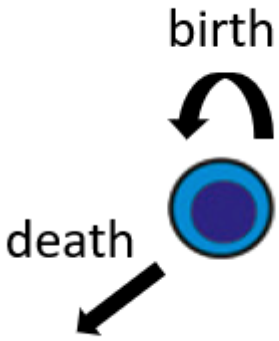

Figure 3.2: One-type birth-death model in which members of the population live for an exponentially-distributed time with parameter $1/(birth + death)$. At the end of an individual's lifetime, an individual gives birth to two new individuals with probability $birth/(birth + death)$ for a net population increase of 1 and will dies with probability $death/(birth + death)$ for a net population decrease of 1.

```
library(estipop)

# Specify how many units of time to simulate
time = 5

# Initiate with a single type with size 100
initial = c(100)

# Specify two fixed transitions, birth and death
transitionList = TransitionList(FixedTransition(population = 0,
                                                rate = 1.0,
                                                fixed = c(2)),
                                FixedTransition(population = 0,
                                                rate = 0.7,
                                                fixed = c(0)))

# No other stops beyond time
stopList = StopList()

# Simulation 100 trials
ntrials = 1000
```

```
full_res = matrix(ncol = 2)

# Run simulations and store results into res
for(i in 1:ntrials){
        res = branch(time, initial, transitionList, stopList,
                silent = TRUE)
        full_res = rbind(full_res, as.matrix(res))
}

full_res = na.omit(full_res)
```

The "full_res" matrix now contains the size of the population output at each integer time for each of the 1,000 trials. Here, we only retain the results from timepoint 5. We can now use this simulated data to estimate the birth and death rate parameters. The following code performs the desired estimation.

```
# Keep only the population size for timepoint 5
data = as.matrix(full_res[full_res[,1] == 5,2])

# Set up our estimation parameters

N = c(100)

time = 5

# Specify two fixed transitions, birth and death
transitionList = TransitionList(FixedTransition(population = 0,
                                        fixed = c(2)),
                                FixedTransition(population = 0,
                                        fixed = c(0)))

initial = c(1, 0.5)

# Estimate using the estimateBP function
```

```
estimates = estimateBP(time = time,
        N = N,
        transitionList = transitionList,
        data = data,
        initial = initial)
```

The results from `estimateBP` are those that are returned from the `optim` function, using the default values from ESTIpop. Our estimated birth rate is 0.953 and our estimated death rate is 0.653. Our true birth rate parameter is 1.0 and our true death rate parameter is 10.7. These estimated results differ slightly from the true values and repeating this procedure with new simulated data results in different estimates, akin to a sampling distribution.

Simulation Overview

To simulate a general multitype branching process using ESTIpop requires specifying four main components: (1) simulation time, (2) initial population vector, (3) a TransitionList, and (4) a StopList. These are passed as arguments to either the `branch` or `branchTD` function.

Two-Type Birth-Death-Mutation Example

In our simulation example, let us look beyond a simple birth-death process by adding an additional cell type. In this new model shown in Figure 3.3, there is a parent population, called type 1, which experiences birth and death events. During each birth event, with some probability, a mutation event occurs that gives rise to a second type, type 2. This second type also has birth and death events. We assume that the mutation bestows some extremely beneficial func-

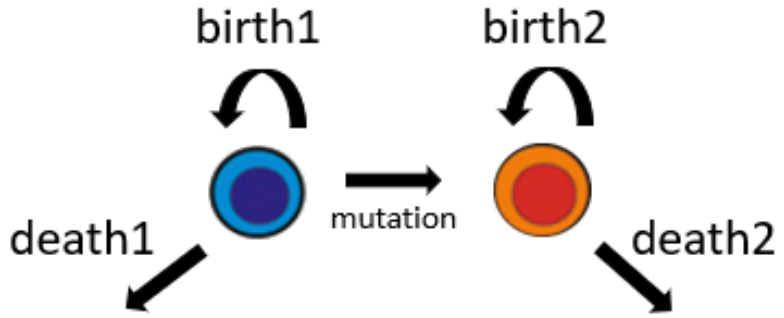

Figure 3.3: A birth-death-mutation model is a model comprised of two types, sensitive (blue) and resistant (red) cells. Each type has its own birth and death rates. Additionally, individual cells from the sensitive type may undergo a mutation event, which confers resistance.

tionality to individuals of that type, and thus the growth rate of this type will be many times higher than that of the parental type. We create a StopList to halt the simulation once either population reaches 5,000. To this end, we use the following code:

```
library(estipop)

# Specify how many units of time to simulate
time = 25

# Initiate first type with size 100 and second type size 0
initial = c(100, 0)

# Define variables for rate parameters
birth1 = 1.0
death1 = 0.7
mutation1 = 0.001
birth2 = 5
death2 = 3
```

```
# Specify transitions
transitionList = TransitionList(
        FixedTransition(population = 0,
                        rate = birth1 * (1 - mutation1),
                        fixed = c(2, 0)),
        FixedTransition(population = 0,
                        rate = death1,
                        fixed = c(0, 0)),
        FixedTransition(population = 0,
                        rate = birth1 * mutation1,
                        fixed = c(1,1)),
        FixedTransition(population = 1,
                        rate = birth2,
                        fixed = c(0, 2)),
        FixedTransition(population = 1,
                        rate = death2,
                        fixed = c(0, 0)))

# Specify to stop simulation once the population exceeds 1000
stopList = StopList(
                StopCriterion(indices = c(0),
                        inequality = ">=",
                        value = 5000),
                StopCriterion(indices = c(1),
                        inequality = ">=",
                        value = 5000))

# Run simulation and store results into res
res = branch(time, initial, transitionList, stopList, silent = TRUE)
```

Let us plot the population sizes over time. We expect to observe zero cells of type 2 until a mutation event from the type 1 population occurs. After this moment, we see rapid growth of this type 2 population. In Figure 3.4, we see that the type 2 population remains at zero until time 10 and then experiences tremendous growth in the next few time units to reach 5,000 before the type 1 population.

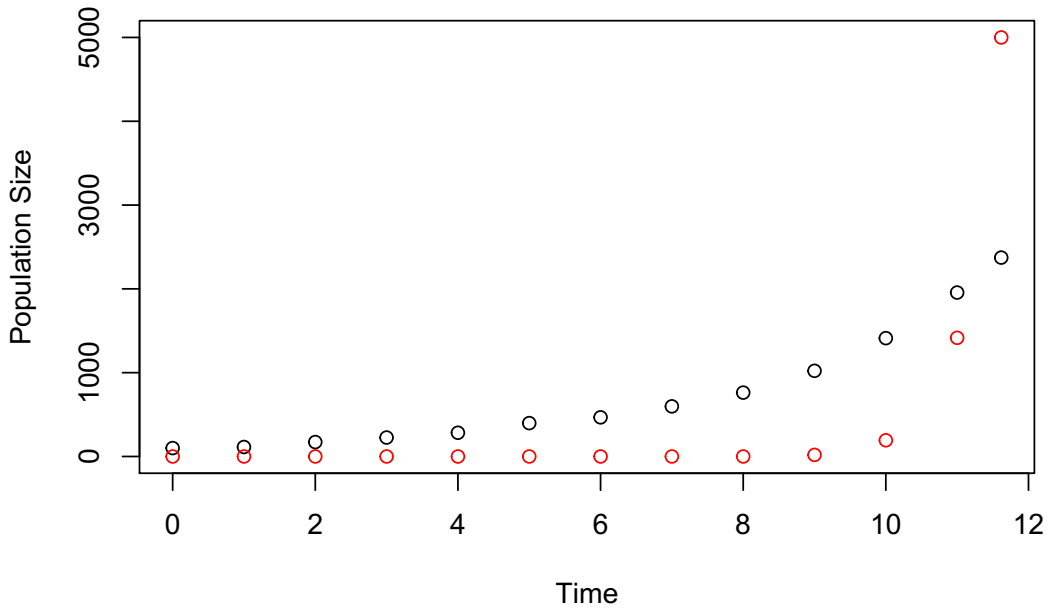

Figure 3.4: Population size of a two-type birth-death-mutation process shown over time. The first population (black circles) grows according to its birth and death parameters. The second population (red circles) remains at 0 until a mutation event generates one individual, at which time the population experiences tremendous growth.

Time-Dependent Rate Example

We now further extend the previous example to include a time-dependent rate. Here, we initiate the system with 100 type 1 individuals and 500 type 2 individuals. The type 1 individuals have the same growth patterns as shown in the first example; however, the type 2 individuals do not grow for the first 5 units of simulation, but experience slightly increased growth compared to type 1 individuals in subsequent times. We perform this simulation by using a Rate template of type 2, which designates a switch, where the rate is one value before the switch time and another value after the switch time. Importantly, because we now are simulating using time-dependent rates, we use the `branchTD` simulation function. We use the following code:

```
library(estipop)

# Specify how many units of time to simulate
time = 10

# Initiate first type with size 100 and second type with size 500
initial = c(100, 500)

# Specify some parameters for rates
birth1 = 1.0
death1 = 0.7
mutation1 = 0.001
birth2 = 1.1
death2 = 0.7

# Specify two fixed transitions, birth and death
transitionList = TransitionList(
```

```
        FixedTransition(population = 0,
                rate = birth1 * (1 - mutation1),
                fixed = c(2, 0)),
        FixedTransition(population = 0,
                rate = death1,
                fixed = c(0, 0)),
        FixedTransition(population = 0,
                rate = birth1 * mutation1,
                fixed = c(1,1)),
        FixedTransition(population = 1,
                rate = Rate(type = 2,
                        params = c(0, birth2, 5.0)),
                fixed = c(0, 2)),
        FixedTransition(population = 1,
                rate = Rate(type = 2,
                        params = c(0, death2, 5.0)),
                fixed = c(0, 0)))

# Sepcify to stop simualtion once the population exceeds 1000
stopList = StopList(
        StopCriterion(indices = c(0),
                inequality = ">=",
                value = 10000),
        StopCriterion(indices = c(1),
                inequality = ">=",
                value = 5000))

# Run simulation and store results into res
res = branchTD(time, initial, transitionList, stopList)
```

We use the following code to create a simple plot of the population sizes over time:

```
# Plot the population size of "pop1" vs. simulation time
plot(res$time, res$V2, col = "black",
xlab = "Time", ylab = "Population Size", ylim = c(0, max(res)))
points(res$time, res$V3, col = "red")
```

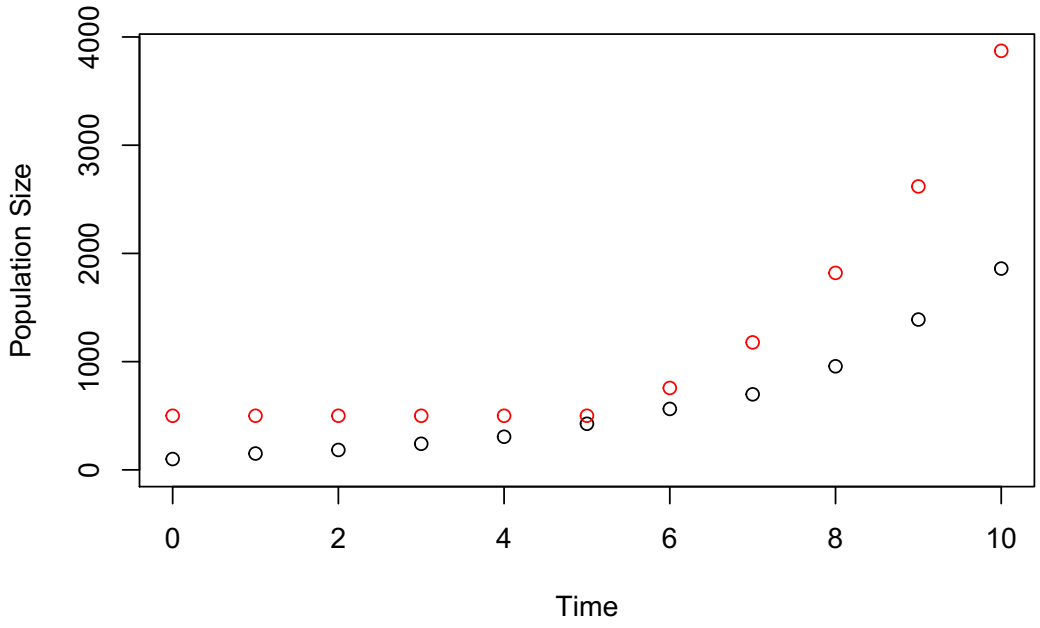

Figure 3.5: Population size of a two-type birth-death-mutation process shown over time. The first population (black circles) grows according to its birth and death parameters. The second population (red circles) remains at its initial value of 500 for 5 units of time and then experiences growth thereafter.

The resulting plot is shown in Figure 3.5, where the type 2 population remains constant for 5 units of time according to the switch rate, and then experiences growth following time unit 5.

3.4 Application

Introduction

Targeted cancer treatments often initially result in impressive responses; however, frequently, tumors develop resistance to these therapies[34]. Generally, this resistance is thought to occur through the acquisition of a de novo mutation which confers resistance to previously sensitive cell phenotype[77,9,62]. A second possible explanation is that in the heterogeneous tumor mass, even before treatment, resistant phenotypes exist[21]. If this is the case, current diagnostic and therapeutic strategies would need to be changed in order to better detect and target these pre-existing resistant clones.

Bhang et al.[12] sought to address this question using high-complexity barcoding. In particular, one portion of the study focused on using the KCL-22 cell line, a cell line derived from a chronic myeloid leukemia (CML) patient, to examine resistance mechanisms to ABL1 inhibitors such as imatinib, nilotinib, and GNF-2.

Once a therapy is determined to be effective, determining the proper dosing schedule can be a tough challenge. There are possibly an unlimited number of schedules to choose from and it is unethical and prohibitively expensive to test each one using a randomized clinical trial. One approach is to develop a model of the tumor-therapy system and then run a series of in-silico clinical trials to determine which schedules to advance to randomized clinical trials[40,19]. Developing this model often includes multiple steps, including specifying the structure of the

model, parameterizing the model, and validating the model with external data. It is important to be able to parameterize models using experimental data to ensure that the resulting model is predictive of true biological behavior.

Model

As a model of tumor growth in the presence or development of resistance, we used the two-type birth-death-mutation model shown in Figure 3.3. The two types of the model are the sensitive and resistant cells. Each type proliferates according to its own birth and death rates. There is also a mutation event from the sensitive population to the resistant population, whereby during each birth or mitosis event in the sensitive population, with small probability a mutation occurs that confers the resistant phenotype.

Data

There are two data sets used to estimate the rate parameters in a birth-death-mutation model (Figure 3.3). First, five replicates of a mixture of the sensitive and resistant populations containing nearly 100 million cells were seeded in plates containing GNF-2 treatment. Over the course of 21 days, these replicates initially shrink, as the sensitive cells are killed in response to treatment, but rebound as the resistant population expands following exponential growth. During this time, on certain days, two replicates were randomly selected from the 5 for measurement of the total number of viable cells, which consists of both sensitive and resistant cells. On the 21st day, all replicates were harvested and measured resulting in a data set con-

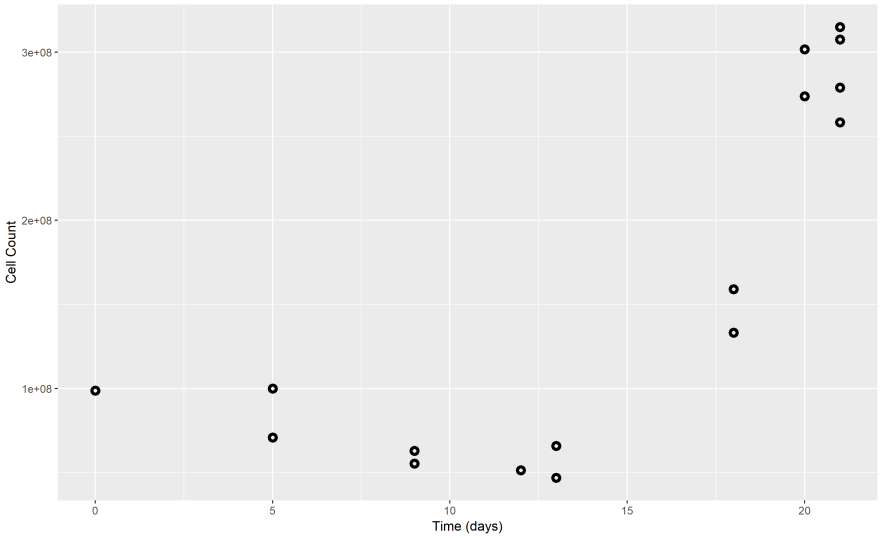

Figure 3.6: Combined sensitive and resistant cell growth. Five replicates, each initially seeded with around 100,000,000 cells were randomly measured over the course of 21 days.

sisting of 21 observations. These data are shown in Figure 3.6.

The initial experiments possibly containing a combination of sensitive and resistant cells were run until resistance emerged. At this point in time, four different clones were isolated and extracted from the resistant population. Each of these four clones was expanded to around half a million cells and then plated in GNF-2 treatment. These resistant clones were then allowed to further expand in treatment for 6 days, during which time they were assayed 4 additional times for both viable and dead cells. These cells are assumed to only contain resistant cells, although each clone could and likely does harbor a different resistance mechanism. These data are shown in Figure 3.7.

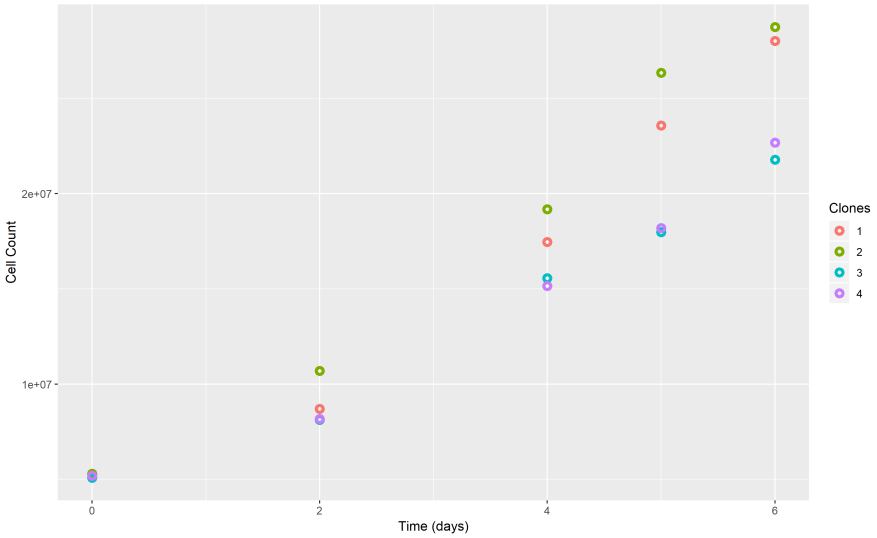

Figure 3.7: Growth assay for four resistant clones (color). Each clone was measured four times after an initial seeding across 6 days. Clones 1 & 2 carried an A337V mutation, conferring resistance to GNF-2. Clone 3 harbored the T315I gatekeeper mutation, rendering it resistant to imatinib and nilotinib. Clone 4 was found to be resistant to all 3 drugs.

Methods

To estimate the rate parameters for both the sensitive and resistant cells, as well as a mutation rate from the sensitive phenotype to the resistant phenotype, we use the maximum likelihood estimators from ESTIpop with an adapted likelihood. We cannot simply use the likelihood from a two-type birth-death-mutation model as shown in the supplemental materials and vignettes due to the fact that in the mixture data, we observe only the sum of the two types, not the individual counts for each type.

We adapt the likelihood by noting that individually, the processes for both the sensitive and resistant cell types are asymptotically normally distributed. Let $X(t)$ be the process for the sensitive cells and $Y(t)$ be the process for the resistant cells. $X(t) \sim N(\mu_X(t), \sigma^2_X(t))$ and

$Y(t) \sim N(\mu_Y(t), \sigma_Y^2(t))$. An assumption derived from our model is that the two processes are not independent of one another due to the mutation from the sensitive phenotype to the resistant phenotype. Therefore, there is also a covariance $\sigma_{X,Y}(t)$ between $X(t)$ and $Y(t)$.

We only observe the combined total of the sensitive and resistant strains. That is, we observe $Z(t) = X(t) + Y(t)$. By the properties of normally-distributed random variables, $Z(t)$ is normally distributed with mean $\mu_X(t) + \mu_Y(t)$ and variance $\sigma_X^2(t) + \sigma_Y^2(t) + 2\sigma_{X,Y}(t)$. Thus, when calculating the likelihood for combined population $Z(t)$, we use this mean and variance, which are functions of the individual means, variances, and covariance for the two types.

To be able to distinguish between the birth and death rates of the sensitive and resistant populations, we first estimate these rate parameters separately using data of the resistant clones. This is a one-type birth-death estimation procedure which was extensively explored in Vignette 1. After estimating the birth and death rates of the resistant population, we use those values in the two-type birth-death-mutation model with the modified likelihood.

Another parameter for calculating the likelihood is the initial population size. Here, we have a two-type model, but we only observe their sum. We use a separate parameter, ρ, which denotes the initial fraction of resistant cells. We use ρ to specify the initial population sizes: $N = [(1-\rho)N_0, \rho N_0]$ where N_0 is the total number of cells initially plated in the experiment. As this initial fraction of resistant cells is unknown, we estimate the rate parameters separately for varying values of ρ.

Table 3.8: Resistant clone birth and death rate estimates

Clone	Estimated birth rate	Estimated death
1	0.297	0.007
2	0.330	0.019
3	0.270	0.016
4	0.271	0.020

Results

Resistant Cell Estimation

Due to the fact that each clone may harbor its own unique resistant mechanism, we estimated the birth and death parameters for each clone separately. The results are shown in Table 3.8. These results show good agreement with rates estimated by Bhang et al[12] in their data supplement.

Combined Sensitive-Resistant Cell Estimation

After estimating the birth and death rates of the resistant clones, we used the average of the birth and death rates from the clones that appeared to be resistant to GNF-2 (clones 1 and 2) for estimating the sensitive birth rate, sensitive death rate, and mutation rate. In Table 3.9, we present the results of estimation over a range of values for ρ. In general, if the initial fraction of resistant cells is low, small ρ, then the mutation rate from the sensitive population to the resistant population is higher. As ρ increases above a certain point, our estimated values suggest that no mutation is necessary to account for the cell dynamics observed.

Table 3.9: Birth-death-mutation model rate estimates, holding the resistant birth and death rates fixed

ρ	Estimated sensitive birth rate	Estimated sensitive death rate	Estimated mutation rate	Estimated resistant birth rate	Estimated resistant death rate
0e+00	0.93747	1	0.00250	0.29791	0.01614
1e-05	0.93747	1	0.00249	0.29791	0.01614
5e-05	0.93747	1	0.00248	0.29791	0.01614
1e-04	0.93746	1	0.00246	0.29791	0.01614
1e-03	0.93737	1	0.00216	0.29791	0.01614
5e-03	0.93700	1	0.00079	0.29791	0.01614
1e-02	0.92845	1	0.00000	0.29791	0.01614
2e-02	0.90869	1	0.00000	0.29791	0.01614
3e-02	0.89429	1	0.00000	0.29791	0.01614
4e-02	0.88178	1	0.00000	0.29791	0.01614
5e-02	0.86998	1	0.00000	0.29791	0.01614

In the results in Table 3.9, we notice that the estimated death rate is always 1, which was the upper boundary value for our optimizer. This result means that we likely do not have enough data to accurately distinguish birth and death from net growth ($birth - death$).

Simulated Trajectories

To verify that our estimated parameter schemes were able to recapitulate the experimental data, we simulated trajectories using the estimated rate parameters for each value of ρ. These trajectories, along with the original experimental data, are shown in Figure 3.8. These results suggest an initial resistant fraction less than 0.01, which is in agreement with the conclusion drawn in the paper that $\rho \approx 0.0002 - 0.0003$.

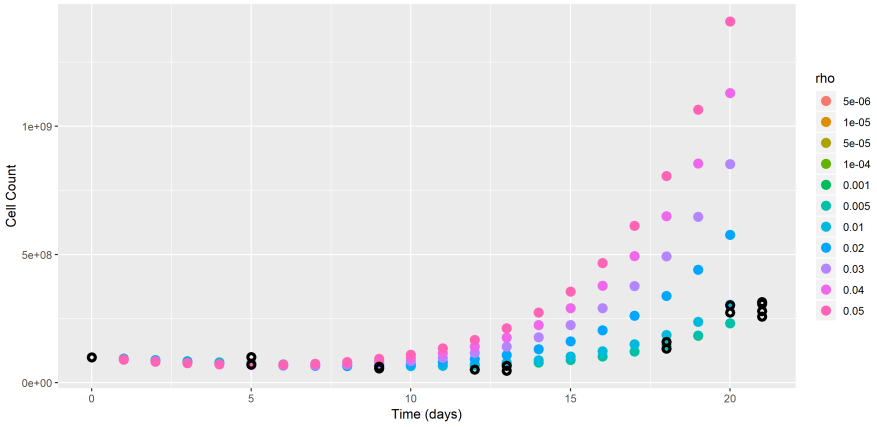

Figure 3.8: Simulated trajectories using the estimated rates from the experimental data for differing values of the initial resistance fraction, ρ (colored points). Experimental data are shown as black circles.

Sample Size Calculation

We then performed a sample size analysis to evaluate the level of confidence in our estimates obtained from the above procedures. In the case of the combined sensitive-resistant cell estimation, we can use simulation as a method to determine how many samples we would need to collect to be able to accurately distinguish the sensitive birth and death rates.

Resistant Cell Simulation

For the resistant clones, we simulated a birth-death model for various samples sizes under the ground truth that birth rate $= 0.298$ and death rate $= 0.016$ per cell per day. We simulated 1,000 replicates per samples size and the results are shown in Figure 3.9.

Even at small samples sizes close to 10 total samples, our estimates of the birth and death rates are within a few hundredths of the true rates; however, in practice, the variance of our

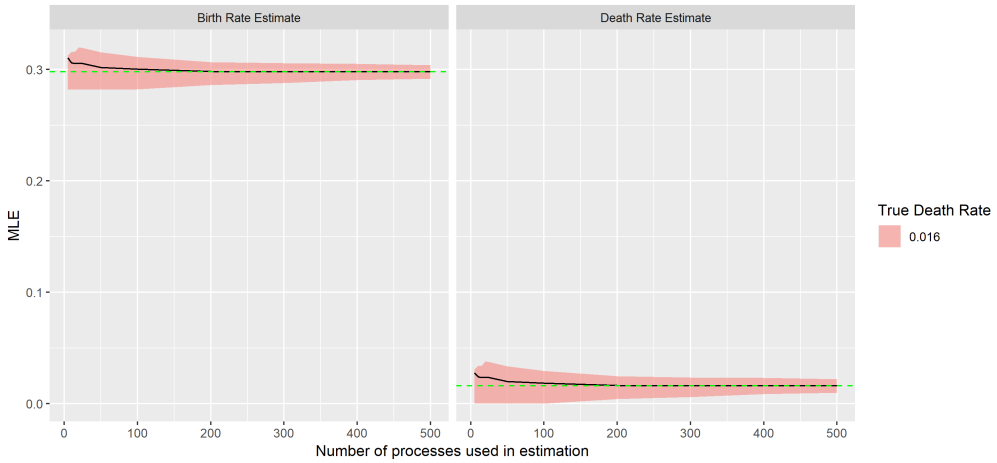

Figure 3.9: Mean estimates (black line) and 2.5th and 97.5th percentiles (shaded area) for the birth and death rate parameters in the one-type birth-death model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines, true birth rate $= 0.298$ and true death rate $= 0.016$.

estimates would also be affected by experimental conditions, such as additional noise or measurement error and ancestors of the process not being truly independent, which could be the result of space limitations, see Vignette 1.

Combined Sensitive-Resistant Cell Simulation

For the combination of sensitive and resistant cells, we simulated a birth-death-mutation model for various samples sizes under the ground truth that for the sensitive population, $birth_{sens} = 0.237$ and $death_{sens} = 0.300$, for the resistant population, $birth_{res} = 0.298$ and $death_{res} = 0.016$, and mutation rate, $\mu = 0.0025$. We assume that the initial population is comprised of 100,000,000 sensitive cells, $\rho = 0$. We simulated 100 replicates per sample size with the additional assumption that our resistant birth and death parameters were known. For each data point, we sampled from the time points of the original data. The results are shown in Figure

3.10.

Here, we observe results that are similar to those presented in Vignette 2. Given that the resistant birth and death rates are fixed, the mutation rate is estimated precisely with very small variance, even at low sample sizes. On the other hand, having to estimate both the birth and death rate for the sensitive population results in highly variant estimates. We would require thousands of samples in order to be within a few hundredths of the true rate with high confidence (Figure 3.10).

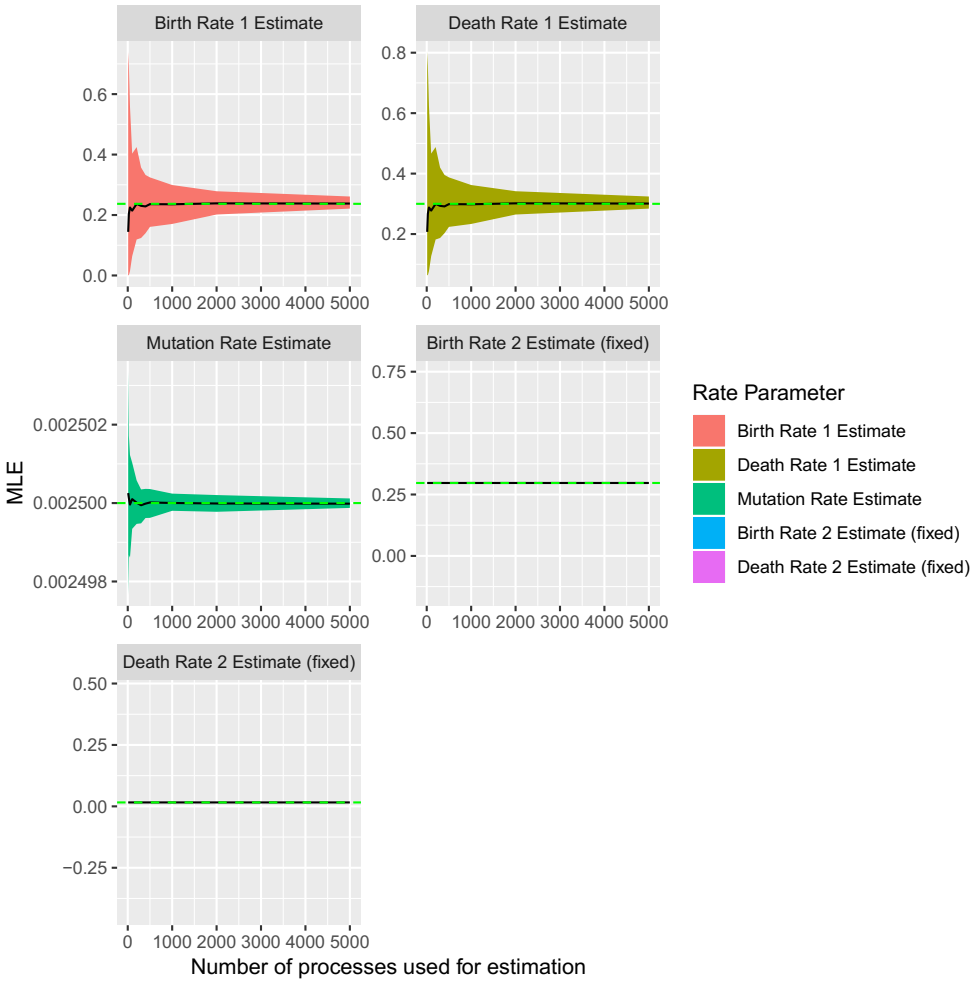

Figure 3.10: Mean estimates (black line) and 2.5th and 97.5th percentiles (shaded area) for the rate parameters in the one-type birth-death model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines. The birth 2 and death 2 rates were held fixed at their true values.

3.5 Conclusion

ESTIpop provides methods to simulate and estimate parameters from complex continuous-time branching processes with wide applications for analyzing biological processes as well as in non-biological fields such as economics[25]. Exact estimation methods for complex systems quickly become intractable as the model complexity increases, but estimation can still be a viable approach using the asymptotic distribution presented herein. Exact simulation, while computationally more expensive, is a useful tool to investigate stochasticity in these systems. Approximate simulation can provide a quick alternative for systems involving extremely large population sizes or lengthy simulation times. Both simulation and estimation can be performed efficiently using parallelization.

A

Analytical Approaches in Evaluation of Screening for Precancerous States

A.1 Analytical Markov Chain Model

A Markov chain is a discrete-time stochastic process in which the transition rates between all possible states of the system are fixed. We used a Markov chain model to calculate the age

distributions and mortality rates of individuals harboring MGUS or diagnosed with MM. We previously used simulation as a tool to analyze such a Markov chain, as simulations have the benefit of directly yielding these distributions once stationarity has been reached. When studying optimization problems, however, simulations become computationally expensive, as an entire simulation must be run to investigate a single instance of the parameter space. Therefore, analytical models can be useful as an analytical model is built on the same parameter space as the simulation, but sometimes allows for an exact analytical solution and numerical methods to quickly explore the parameter space for optimization. For our analytical model, we used a stationary age distribution in the healthy population as input, whereas in the simulation framework, we stochastically simulated individual births and deaths, which then stabilized to a stationary age distribution as simulation time progressed.

In the following section, we show how to calculate the fractions of MGUS and MM individuals in the population as a function of age. Using the resulting analytical expressions, we analyzed MGUS and MM prevalence under varying screening parameter schemes. As in the simulation framework, we assumed that time progresses in discrete steps of 1 year.

Let us define $H_{a,i}(t)$ as the initial fraction of healthy individuals in risk group i at age a. We denote $M_{a,i}(t)$ as the fraction of unscreened individuals harboring MGUS in risk group i at age a, $T_{a,i}(t)$ as the fraction of screening MGUS individuals in risk group i at age a, and $N_{a,i}(t)$ as the fraction of individuals with MM in risk group i at age a. As time progresses, the fraction of healthy individuals at risk, $X_{a,i}(t) = H_{a,i}(t) - M_{a,i}(t) - T_{a,i}(t) - N_{a,i}(t)$, changes until it reaches stationarity. We can think of $X_{a,i}(t)$ as the probability of selecting a healthy individual

of age a and risk group i out of the population.

It is of interest to track $M_{a,i}(t)$, $T_{a,i}(t)$, and $N_{a,i}(t)$. Towards that end, let us further define $d_{a,i}$ as the probability to die at age a for an individual of risk group i, $x_{a,i}$ as the probability to be screened at age a for an individual of risk group i, and $m_{a,i}$ as the probability to develop MGUS at age a for an individual of risk group i. Furthermore, let p be the probability to progress from MGUS to MM per individual per year and r be the risk reduction factor for those individuals who have been positively screened for MGUS. Then, the fraction of undetected MGUS individuals changes over time according to:

$$\begin{aligned} M_{a,i}(t+1) - M_{a,i}(t) = {} & M_{a-1,i}(t)(1-d_{a-1,i})(1-x_{a-1,i})(1-p) \\ & + X_{a-1,i}(t)(1-d_{a-1.i})m_{a-1,i} \\ & - M_{a,i} \end{aligned}$$

From this equation, we can see that the change in the fraction of undetected MGUS individuals is due to two components: inflow from healthy, at risk individuals in the previous age who develop MGUS without dying and those who remain in the undetected MGUS fraction from the previous age who neither die, nor successfully transfer to the screened MGUS population, nor progress to the MM state. To reach stationarity, the condition $M_{a,i}(t+1) - M_{a,i}(t) \stackrel{def}{=} 0$, which leads to a recursion for the fraction of undetected MGUS individuals at age a and risk group i:

$$M_{a,i} = (1 - d_{a-1,i})(M_{a-1,i}(1 - x_{a-1,i})(1 - p) + X_{a-1,i} m_{a-1,i})$$

with boundary condition $M_{0,i} = 0$, as no individuals are born with MGUS. In a similar way, we calculate the change in the fraction of the screened MGUS population at age a in risk group i:

$$\begin{aligned} T_{a,i}(t+1) - T_{a,i}(t) = {} & T_{a-1,i}(t)(1 - d_{a-1,i})(1 - rp) \\ & + M_{a-1,i}(1 - d_{a-1,i})(1 - p) x_{a-1,i} \\ & - T_{a,i}(t) \end{aligned}$$

where inflow comes from those individuals with unscreened MGUS who are successfully screened during the previous age and those who remain in the compartment from the previous age who neither die nor progress to MM. This leads to a recursion

$$T_{a,i} = (1 - d_{a-1,i}) \left(M_{a-1,i} x_{a-1,i}(1 - p) + T_{a-1,i}(1 - rp) \right)$$

with boundary condition $T_{0,i} = 0$ for the fraction of individuals who are screened positively for MGUS. For those individuals with MM, the general law that shapes age-dependent survival is unknown. Thus, we assumed the probability of death due to MM was a constant rate, $d_{MM} = 0.12945$ per individual per year. This yearly probability accurately recovers that

for MM patients, the median survival time post-diagnosis is 5 years. The fluctuations of MM patients at age a in risk group i over time is given by

$$\begin{aligned} N_{a,i}(t+1) - N_{a,i}(t) = {} & N_{a-1,i}(t)(1-d_{MM}) \\ & + M_{a-1,i}(t)(1-d_{a-1,i})(1-x_{a,i})p \\ & + M_{a-1,i}(t)(1-d_{a-1,i})x_{a,i}rp \\ & + T_{a-1,i}(t)(1-d_{a-1,i})rp \\ & - N_{a,i}(t) \end{aligned}$$

where inflow to the MM compartment comes from unscreened MGUS individuals, who progress to MM or are successfully screened and progress at the reduced rate in the same year, screened MGUS individuals who progressed to MM at the reduced rate, and individuals who had already developed MM in previous ages, but had not yet succumbed to the disease. Due to no individuals are initialized with MM, $N_{0,i}(0) = 0$, we obtain the following recursion for the fraction of individuals with MM at age a in risk group i

$$N_{a,i} = (1-d_{a-1,i})(M_{a-1,i}p(1-x_{a,i}+rx_{a,i}) + T_{a-1,i}rp) + (1-d_{MM})N_{a-1,i}$$

with boundary condition $N_{0,i} = 0$. These recursions can be solved iteratively in order to calculate the fractions of unscreened MGUS individuals, screened MGUS individuals, and

MM individuals. Calculating prevalences from these fractions can be achieved by multiplying the fractions by the total population size.

A.2 Cumulative MM-specific mortality conditioned on MGUS detection

As previously discussed, cumulative disease-specific mortality is a better measure to use in screening situations in which lead-time bias can exaggerate the benefits to individuals who are successfully screened. Towards analytically calculating the cumulative MM-specific mortality, denote $c_{a,x}$ as the probability of death as a result of MM between the age of $a+x$ and $a+x+1$ after a positive screen at age a. Let d_a denote the probability of dying due to all other causes between the ages of a and $a+1$ and d_{MM} denote the probability of dying as a result of MM, which again, is assumed to be independent of age and time since progression from MGUS to MM. Let us introduce the notation $D_a = (1-d_a)(1-q)$ as the probability of progression-free survival at age a, where $q = rp$ for progression rate p and reduction factor r. Since MM-specific mortality is conditioned on being positively screened for MGUS, we only consider the reduced progression rate rp. Now consider an individual who is positively screened for MGUS at age a. The probability that such an individual dies the exact same year as MGUS detection is given by

$$c_{a,0} = qd_{MM},$$

where the individual immediately progresses to MM and subsequently dies. The probability of dying exactly one post-MGUS detection is given by

$$c_{a,1} = q(1 - d_{MM})d_{MM} + D_a q d_{MM},$$

where the individual either progresses to MM during their first year of exposure and subsequently dies in the next year (first term) or they survive progression-free during the first year of exposure and then progress to MM and die in the same year (second term). The probability of dying in during the second year post-MGUS detection is given by

$$c_{a,2} = q(1 - d_{MM})^2 d_{MM} + D_a q(1 - d_{MM})d_{MM} + D_a D_{a+1} q d_{MM},$$

where the individual either progresses during the initial, first, or second year as shown in the respective terms for $c_{a,2}$. Continuing this pattern, we arrive at the formula for $c_{a,x}$ as given by

$$c_{a,x} = q d_{MM} \left[(1 - d_{MM})^x + \sum_{k=0}^{x-1} (1 - d_{MM})^k \prod_{l=0}^{x-k-1} D_{a+l} \right].$$

We can then calculate the cumulative MM-specific mortality, $C_{a,\alpha} = \sum_{x=0}^{\alpha} c_{a,x}$. We note that as expected, $C_{a,\alpha}$ does not depend on any screening parameters, as we have already stated

that cumulative MM-specific mortality is calculated conditional on already having a positive screen; however, cumulative MM-specific mortality does depend on the progression rate p, risk reduction r, as well as the death probabilities due to MM and all other causes, d_{MM} and d_a respectively.

A.3 Evolving MGUS

As previously discussed, the rate of progression from MGUS to MM might not be independent of time, but rather be some function of time since MGUS incidence. In particular, let $p(t)$ be the risk of progression from MGUS to MM for an individual who is t years post MGUS incidence. One possible parametric form for this function could be $p(t) = (1-\beta)^t\beta$, where β is the risk of immediately transitioning from MGUS to MM upon MGUS incidence. We can thus modify our equation for the MM-specific mortality by letting $q(t) = rp(t)$ and $D_{a+l} = (1-d_{a+l})(1-q(l))$:

$$c_{a,x} = d_{MM}\left[q(0)(1-d_{MM})^x + \sum_{k=0}^{x-1} q(x-k)(1-d_{MM})^k \prod_{l=0}^{x-k-1} D_{a+l}\right].$$

As before, the single parameter β can be estimated using least squares regression fitting the cumulative progression rate, $1-(1+\beta)^{n+1}$, which is the probability to progress at any time before $n+1$ years after MGUS detection. Values for β were estimated using cumulative progression rates in evolving and non-evolving MGUS from Rosinol et al[66].

B

DIFFpop Supplemental Materials

B.1 Application

Background

To begin our example, let us consider a subtree of the hematopoietic system shown in Figure B.1. This system has been studied in mice by researchers who have developed a mouse model that introduces a fluorescent tag into certain cell population[16]. Once activated, this tag, inte-

grated into the genome of the cell, will continue to be present in the progeny of the initially labeled cell. The uptake and loss of the label can then be observed as it descends through the differentiation hierarchy. Assuming that the system has reached a steady state with no major population fluxes, Busch et al. employed an ordinary differential equation model to estimate the sizes of the compartments and rates of transitions between compartments. To validate our tool using the experimental results[16], we used the previously derived parameter estimates determined to provide the best fit to the data and predicted, using our tool, the cell numbers at time points not used for parameter estimation.

DIFFpop Simulation

The following script was used to run simulations of the hematopoietic system as modeled by Busch et al. in their 2015 Nature paper. To mimic the experimental procedure, we initially labeled only LT-HSC cells. Because the parameter estimates were calculated under the assumption that steady state hematopoiesis had been reached, we modeled the system using FixedPops, which maintain constant population sizes across the system.

```
library(foreach)
library(doParallel)

# Set up parallelization
cores = detectCores()
cl = makeCluster(cores[1]-1)
registerDoParallel(cl)
```

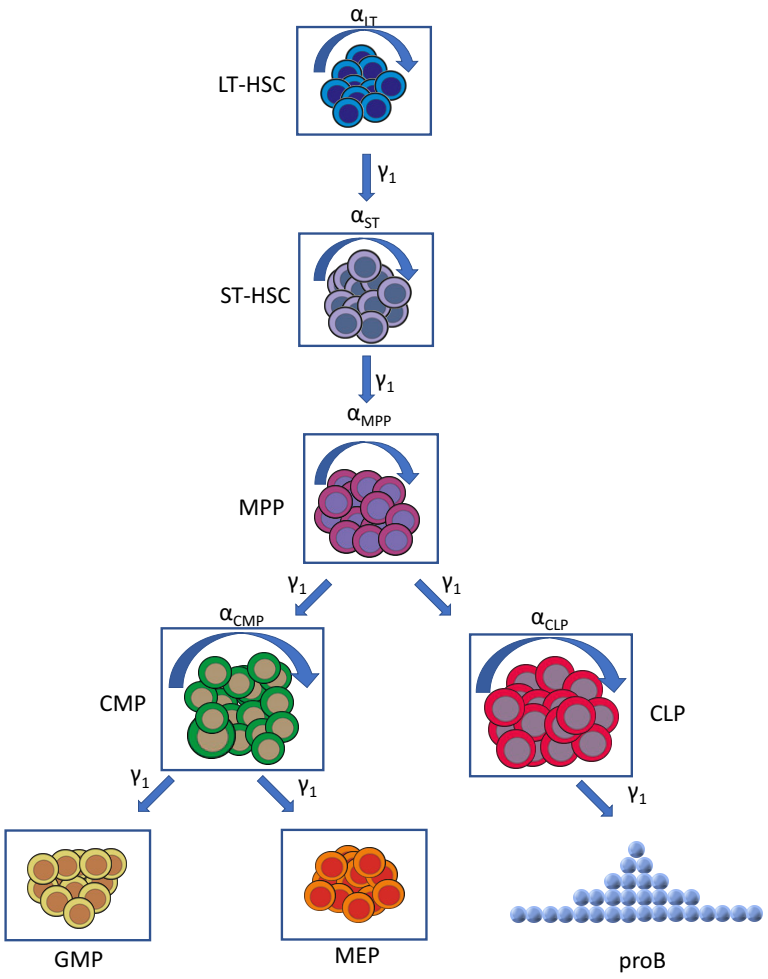

Figure B.1: Hematopoietic system modeled using FixedPops. Abbreviations: long-term hematopoietic stem cell (LT-HSC), short-term hematopoietic stem cell (ST-HSC), multi-potent progenitor (MPP), common myeloid progenitor (CMP), common lymphoid progenitor (CLP), granulocyte-macrophage progenitor (GMP), megakaryocyte-erythroid progenitor (MEP). Events between populations consist of mitosis (α) and mitosis-independent differentiation (γ_1).

```
ntrials = 1000

foreach(i_=1:ntrials) %dopar%{
print(i_)
library(diffpop)

# Simulation size and label parameter
nLT = 5000
LT_lbl = 0.01

# Blank DiffTree object
tree = DiffTree()

# Add all pops to tree using population
#                          sizes estimated from Busch et al.
FixedPop(tree, "LT", nLT, LT_lbl)
FixedPop(tree, "ST", as.integer(2.9*nLT), 0.0)
FixedPop(tree, "MPP", as.integer(9*nLT), 0.0)
FixedPop(tree, "CMP", as.integer(39*nLT), 0.0)
FixedPop(tree, "CLP", as.integer(13*nLT), 0.0)
FixedPop(tree, "GMP", as.integer(0.24*39*nLT), 0.0)
FixedPop(tree, "MEP", as.integer(0.39*39*nLT), 0.0)
FixedPop(tree, "proB", as.integer(108*13*nLT), 0.0)

# Add self-renewal/mitosis events
addEdge(tree, "LT", "LT", "alpha", 0.009)
addEdge(tree, "ST", "ST", "alpha", 0.042)
addEdge(tree, "MPP", "MPP", "alpha", 4)
addEdge(tree, "CLP", "CLP", "alpha", 3.00)
addEdge(tree, "CMP", "CMP", "alpha", 4)

# Add differentiation events
# Note: Busch et al. assume mitosis-independent differentiation
addEdge(tree, "LT", "ST", "gamma1", 0.009)
addEdge(tree, "ST", "MPP", "gamma1", 0.045)
addEdge(tree, "MPP", "CLP", "gamma1", 0.022)
addEdge(tree, "MPP", "CMP", "gamma1", 3.992)
addEdge(tree, "CLP", "proB", "gamma1", 2.000)
addEdge(tree, "CMP", "GMP", "gamma1", 2)
addEdge(tree, "CMP", "MEP", "gamma1", 3)
```

```
# Set LT population as root of tree
setRoot(tree, "LT")

# Simulate tree for 800 time units (days)
# Note: we use FixedPops here because parameters were
#       estimated for steady state hematopoiesis
simulateTree(tree = tree,
        fixed = TRUE,
        time = 800,
        indir = paste("input/", i_, sep = ""),
        outdir = "output/",
        census = -1)
}

stopCluster(cl)
```

The above script was run on a cluster with multiple simulation trajectories running in parallel.

Comparing Simulation Output to Experimental Data

The raw experimental data from the mouse model provides the percentage of cells in each population that express the label. These raw percentages are then normalized by the label percentage in the LT-HSC population. This same information can be calculated from the information in the simulation label output files (*prefix*_label.csv). We downloaded all of these label output files from the cluster and stored them in a local directory. We then ran the following script to load the data into R and transform the raw label percentages to percentages relative to the LT-HSC population.

```r
library(reshape2)
library(ggplot2)
library(grid)
library(gridExtra)


# Set working directory to that which contains our label output files
inDir = "C:/DFCI/Jeremy/flex/diffpop_review2/label_newpop2/"
setwd(inDir)

# Generate a list of all filenames
lblfiles = list.files(inDir, pattern="^out.*_label.csv$", full.names=F)

# Read in all of the label files into one dataframe
myMergedData <-
  do.call(rbind,
lapply(lblfiles, read.csv))

# Label each row with a corresponding file id
nfiles = length(lblfiles)
myMergedData$id = rep(1:nfiles, each = 801)

# Remove any data point where the LT label has died out
#        (necessary to divide by it in next step)
myMergedData = myMergedData[myMergedData$LT != 0.0,]
myMergedData[,3:9] = myMergedData[,3:9] / myMergedData[,2]
```

We summarized the 1,000 trajectories by calculating the median trajectory, as well as the 25th and 75th percentiles to act as confidence bounds. We then looped over all of the populations to plot. We also added in the data points at the best resolution possible that come from the Busch et al. paper (Figure 3 and Extended Data Figure 6)[16]. In the following plots, the blue data points were used by Busch et al. to fit the model parameters, whereas the green data points were used for validation.

```
# Melt the data set in order to plot using ggplot2
myMergedData = melt(myMergedData, id.vars = c("time", "id"))

# Rename some columns
names(myMergedData) = c("time", "id", "pop", "value")

# Experimental results directory
exp_dir = "C:/DFCI/Jeremy/flex/diffpop_review2/"

# Plotting function
plot_dat = function(pop, df){
        exp_pts = read.csv(paste(exp_dir, tolower(pop), "_fit.csv", sep = ""),
        header = F)
names(exp_pts) = c("time", "value")
        exp_pts$id = 1
        exp_pts$color = "blue"

if(file.exists(paste(exp_dir, tolower(pop), "_pred.csv", sep = ""))){
    pred_pts = read.csv(paste(exp_dir, tolower(pop), "_pred.csv", sep = ""),
    header = F)
    names(pred_pts) = c("time", "value")
    pred_pts$id = 1
    pred_pts$color = "green"

    exp_pts = rbind(exp_pts, pred_pts)
  }

  p <- ggplot(df[df$pop == pop,], aes(x = time, y = value, group = id))
  p = p +  stat_summary(aes(group = 1), geom = "ribbon",
        fun.ymin = function(x) quantile(x, 0.25),
        fun.ymax = function(x) quantile(x, 0.75),
        col = "grey80", alpha = 0.3) +
        stat_summary(aes(group = 1), geom = "line", fun.y = median,
        col = "red", size = 2) +
        geom_point(data = exp_pts, col = exp_pts$color, alpha = 1.0) +
        xlim(0, max(exp_pts$time + 50)) +
        ggtitle(pop) +
        ylab("")

  return(p)
```

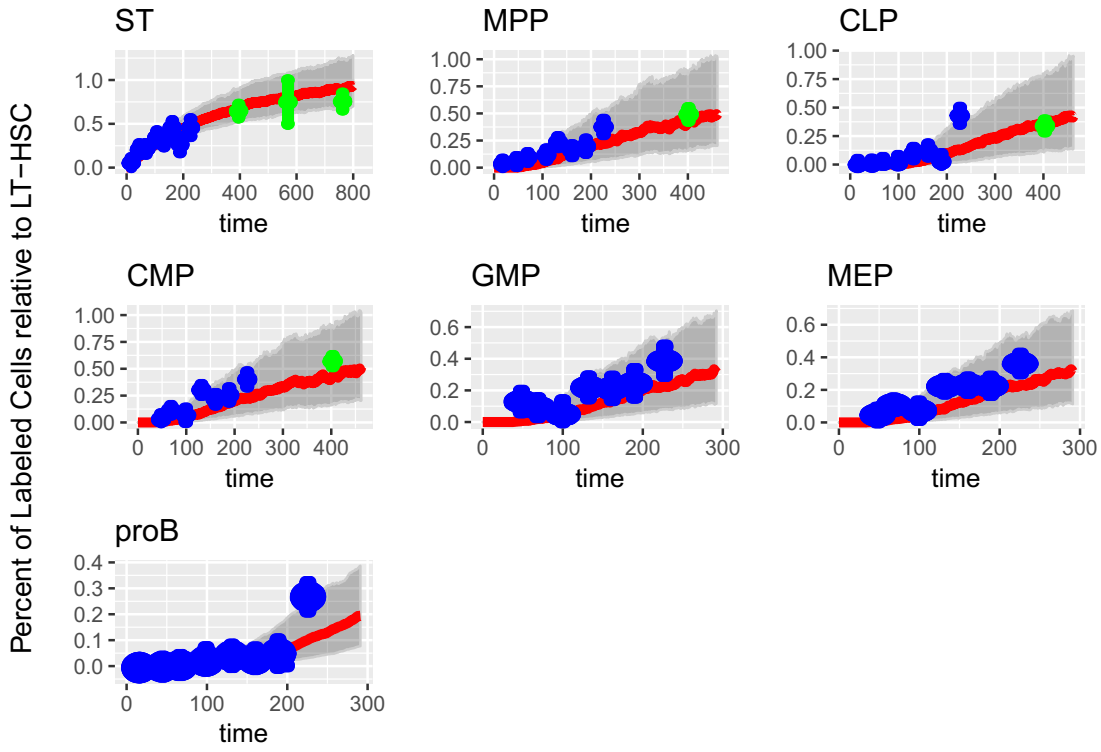

Figure B.2: Hematopoietic system modeled using FixedPops. Abbreviations: long-term hematopoietic stem cell (LT-HSC), short-term hematopoietic stem cell (ST-HSC), multi-potent progenitor (MPP), common myeloid progenitor (CMP), common lymphoid progenitor (CLP), granulocyte-macrophage progenitor (GMP), megakaryocyte-erythroid progenitor (MEP). Events between populations consist of mitosis (α) and mitosis-independent differentiation (γ_1).

```
}

# Apply plotting function to each population
plot.list = lapply(c("ST", "MPP", "CLP", "CMP", "GMP", "MEP", "proB"),
                   plot_dat, df = myMergedData)

# Arrange the plots in a nice grid
grid.arrange(grobs = plot.list,
left = textGrob("Percent of Labeled Cells relative to LT-HSC",
rot = 90, vjust = 1))
```

B.2 Vignette 1: Branching Process

Background

To begin our example, let us consider a subtree of the hematopoietic system shown in Figure B.3. This subsystem has been studied in mice by researchers who have developed a mouse model that introduces a fluorescent tag into certain cell population (Busch, 2015). Once activated, this tag, integrated into the genome of the cell, will continue to be present in the progeny of the initially labeled cell. The uptake and washout of the label can then be observed as it descends through the differentiation hierarchy. From this information and assuming that the system has reached a steady state with no population fluxes, Busch et al. used an ordinary differential equation model to estimate the sizes of the compartments and rates of transitions between compartments. Let us use those estimates as a starting point and see how perturbations in various parameters affect the system. In this model, we will only be using a subset of the event types available in DIFFpop. As an example, the events for the short-term hematopoietic stem cells (ST) are shown in Figure B.4.

Using DIFFpop in R

In R, we first specify the populations of the tree using the appropriate DIFFpop functions. To do this, we determine which of the three basic DIFFpop classes is appropriate, give the population a name, initial population size, and initial population barcoding efficiency, what

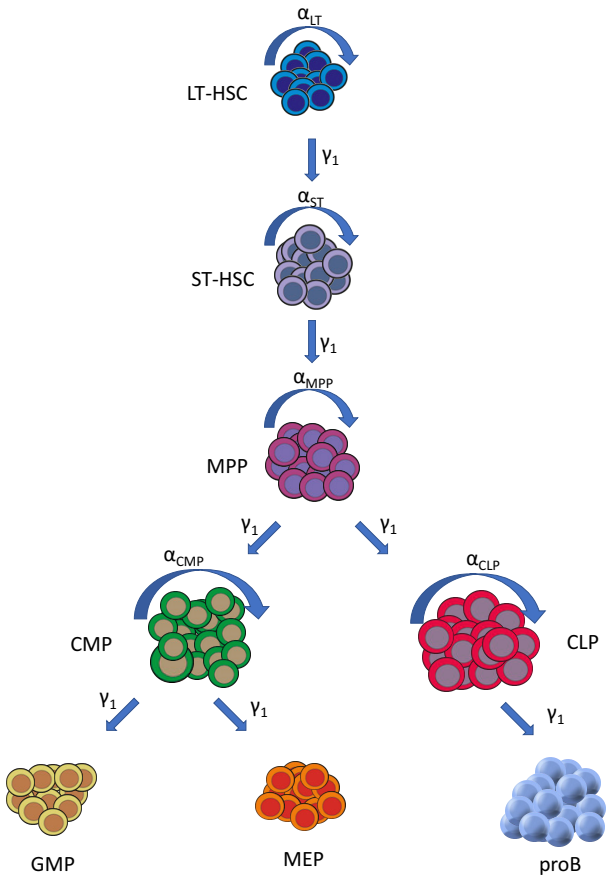

Figure B.3: Hematopoietic system modeled using GrowingPops. Abbreviations: long-term hematopoietic stem cell (LT-HSC), short-term hematopoietic stem cell (ST-HSC), multi-potent progenitor (MPP), common myeloid progenitor (CMP), common lymphoid progenitor (CLP), granulocyte-macrophage progenitor (GMP), megakaryocyte-erythroid progenitor (MEP). Events between populations consist of mitosis (α) and mitosis-independent differentiation (γ_1).

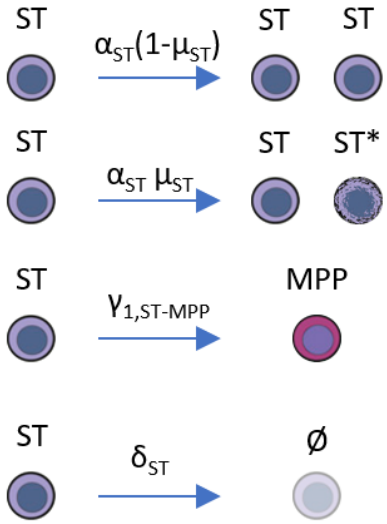

Figure B.4: The cellular events for the following simulations include mitosis with and without mutation, mitosis-independent differentiation, and cell death, each occurring according to the rates specified by lower-case Greek letters. Abbreviation: Short-term hematopoietic stem cell (ST).

proportion of cells are given a unique barcode during simulation initialization. GrowingPop should be used for systems with homogeneous populations that may grow or decline. FixedPop should be used for homogeneous populations that maintain constant sizes. DiffTriangle should be used for constant-sized populations with discrete levels of maturation. Here we initially uniquely barcode only LT-HSC cells. Also note that population sizes are given relative to the LT-HSC population size, allowing us to effectively scale our simulations by changing the number of LT-HSCs. The R code to perform the necessary population specification follows:

```
library(diffpop)

# All other population sizes will be based on nLT
nLT = 500

# Create DiffTree object
tree1 = DiffTree()

# Create cell types with sizes from Busch et al.
GrowingPop(tree1, "LT", nLT, 1.0)
GrowingPop(tree1, "ST", 2.9*nLT, 0.0)
GrowingPop(tree1, "MPP", 9*nLT, 0.0)
GrowingPop(tree1, "CLP", 13*nLT, 0.0)
GrowingPop(tree1, "CMP", 39*nLT, 0.0)
GrowingPop(tree1, "GMP", as.integer(0.24*39*nLT), 0.0)
GrowingPop(tree1, "MEP", as.integer(0.39*39*nLT), 0.0)
GrowingPop(tree1, "proB", as.integer(108*13*nLT), 0.0)
```

We can then use the ‘addEdge‘ function to specify links between the populations. We use the point estimates from (Busch, 2015), who parameterize a net proliferation rate, which is the death rate subtracted from the self-renewal/mitosis rate. Here, we set this value as the mitotic rate and take the cell death parameter for each population to be zero; however, we could add any positive value to both the alpha (mitosis) and delta (death) event rate to remain within the same parameterization. The R code used to specify these transitions is shown below:

```
# Add self-renewal events
addEdge(tree1, "LT", "LT", "alpha", 0.009)
addEdge(tree1, "ST", "ST", "alpha", 0.042)
addEdge(tree1, "MPP", "MPP", "alpha", 4)
addEdge(tree1, "CLP", "CLP", "alpha", 3.00)
addEdge(tree1, "CMP", "CMP", "alpha", 4)
```

```
# Add mitosis-independent differentiation events
addEdge(tree1, "LT", "ST", "gamma1", 0.009)
addEdge(tree1, "ST", "MPP", "gamma1", 0.045)
addEdge(tree1, "MPP", "CLP", "gamma1", 0.022)
addEdge(tree1, "MPP", "CMP", "gamma1", 3.992)
addEdge(tree1, "CLP", "proB", "gamma1", 2.000)
addEdge(tree1, "CMP", "GMP", "gamma1", 2)
addEdge(tree1, "CMP", "MEP", "gamma1", 3)

# Add cell death to terminal populations
addEdge(tree1, "CLP", "CLP", "delta", 1.015)
addEdge(tree1, "GMP", "GMP", "delta", 2*39/(0.24*39))
addEdge(tree1, "MEP", "MEP", "delta", 3*39/(0.39*39))
addEdge(tree1, "proB", "proB", "delta", 2*13/(108*13))
```

To initiate a simulation of the specified tree, the last steps are to first specify a population as the root of the tree, the population that is furthest upstream, and then start the simulation using the simulateTree function with accompanying simulation parameters. Note, for input and output directories, if an absolute path is not specified, DIFFpop will build the specified directory structure from the R working directory. Since we are simulating using GrowingPops, we will set the *fixed* parameter to FALSE.

```
# Set root and simulate the hierarchy
setRoot(tree1, "LT")
simulateTree(tree = tree1,
        fixed = FALSE,
        time = 700,
        indir = "example/",
        outdir = "example/")
```

Results

One output for the simulation is the size of each population after each unit of simulation time. Let us investigate how deviations in one parameter from steady-state parameter set influences the system in terms of population sizes. We varied the self-renewal (mitotic) rate for the LT-HSC population, (α_{LT}). Changing this rate at the top of our differentiation hierarchy has the potential to affect all downstream populations. For each realized value of α_{LT} we have performed 100 simulations and plotted individual simulation trajectories as the fainter trajectory cloud, as well as the mean trajectory as a bolded trajectory for each population over the various α_{LT}. As expected, the steady state α_{LT} value of 0.009 has produced stable population sizes. Increasing this parameter, we observe an increase in the LT population size over time, with the analogous decline in population size for decreased parameter values. We also observe this same trend carry throughout the populations of the hierarchy. Although the differentiation rate to the downstream ST-HSC population remained the same, an increase in LT population size results in an increase net number of cells progressing through the trees. The trajectories are shown in Figure B.5.

In addition to varying the α_{LT} rate, we also varied the differentiation rate from the LT-HSC population to the ST-HSC population, γ_{LT}. For γ_{LT} rates that exceed the stable γ_{LT} rate of 0.009, we see a decline in the LT-HSC population, as self-renewal is not able to balance the decline in the population due to differentiation downstream. A different effect occurs in the downstream populations for these higher γ_{LT} rates, where initially receiving additional cells

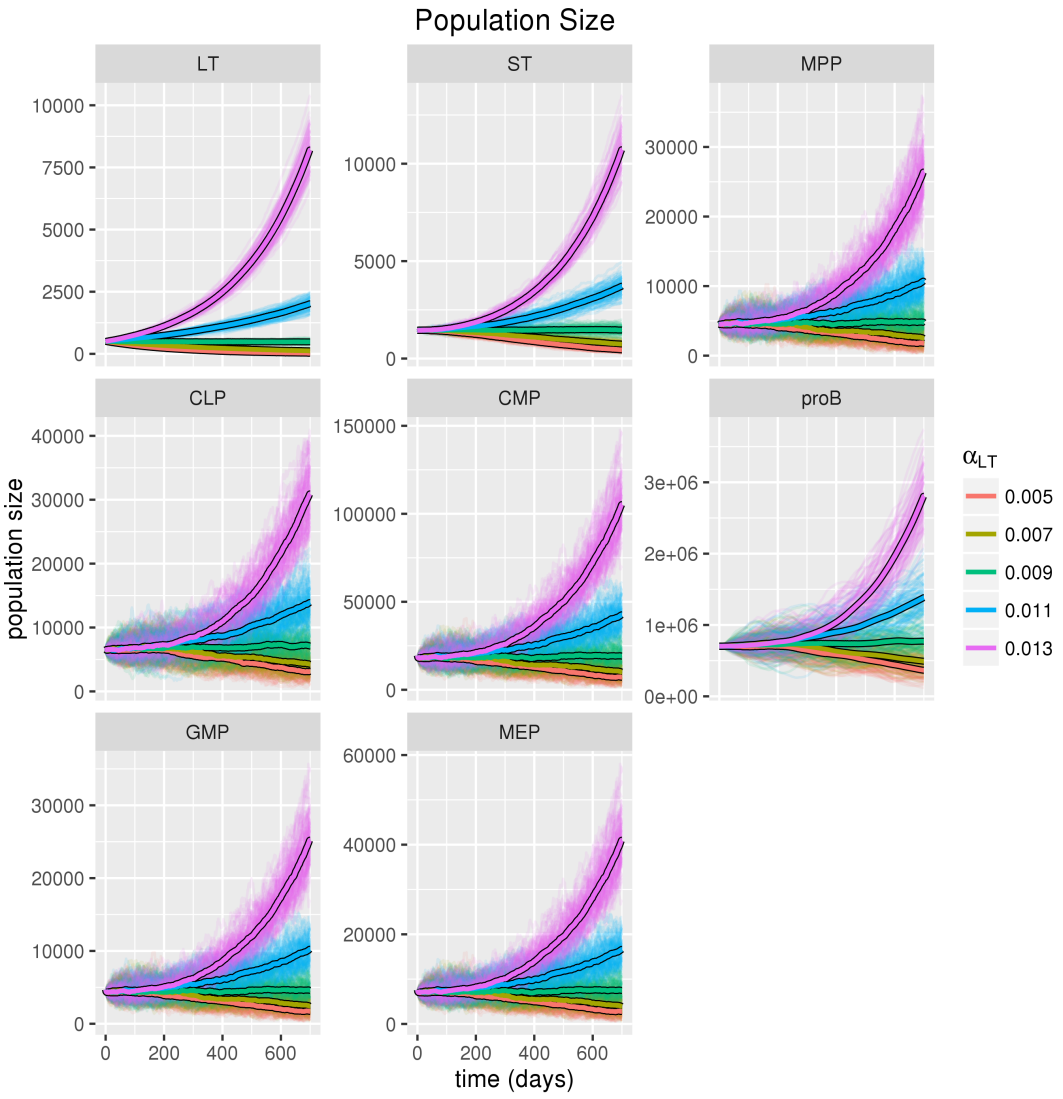

Figure B.5: Trajectories of population size over time are shown for varying α_{LT} rates: red ($\alpha_{LT} = 0.005$), orange ($\alpha_{LT} = 0.007$), green ($\alpha_{LT} = 0.009$), blue ($\alpha_{LT} = 0.011$), purple ($\alpha_{LT} = 0.013$). 100 simulation trajectories for each α_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

from the LT-HSC population due to the high differentiation rate causes an increase in population sizes relative to the stable state. As the LT-HSC population declines in size, however, all downstream populations also being to decline, as they do not receive sufficient input from the LT-HSC population to offset their own differentiation downstream. For γ_{LT} rates lower than the stable γ_{LT} rate of 0.009, we notice the opposite effect. These trajectories are shown in Figure B.6.

DIFFpop also tracks the fraction of cells in a population that contain a barcode or label. Let us investigate the dynamics of label uptake for various α_{LT} and γ_{LT} rates if we initially label only the LT-HSC population. In general, we see a quicker uptake in label with increasing α_{LT} rate throughout the downstream populations. The reasoning is similar to that in the population size results, whereby as the LT-HSC population grows due to an increased mitotic rate relative to the differentiation rate, a net increase in the number of labelled cells moves through the differentiation hierarchy. Similar to the results shown for population sizes, we see that higher γ_{LT} rates result in an initially higher uptake of label throughout the hierarchy, however; this rate is not sustainable due to the overall decline of the LT-HSC population and label uptake declines relative to the lower γ_{LT} rates. The label fraction plots over various α_{LT} and γ_{LT} rates are show in Figures B.7 and B.8 respectively.

We next investigated changes in diversity of the cell populations over time. To this end, we used Shannon's Equitability for various α_{LT} and γ_{LT} rates. Shannon's Equitability is based on Shannon's Diversity Index, which is defined as $SDI = \sum_j p_j \log p_j$,where p_j is the proportion of cells belonging to clone j, where here, a clone is defined by a certain barcode. Then, Shan-

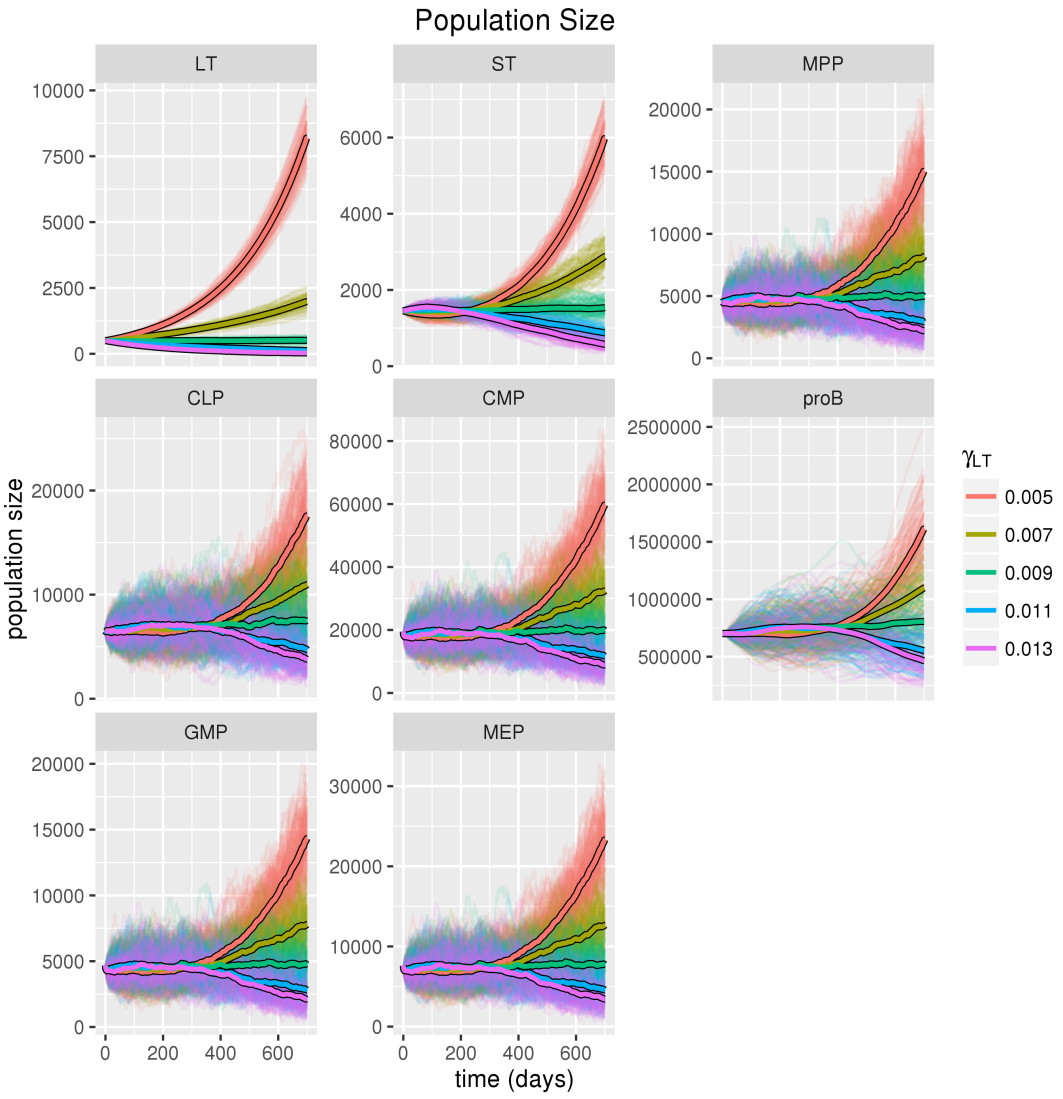

Figure B.6: Trajectories of population size over time are shown for varying γ_{LT} rates: red ($\gamma_{LT} = 0.005$), orange ($\gamma_{LT} = 0.007$), green ($\gamma_{LT} = 0.009$), blue ($\gamma_{LT} = 0.011$), purple ($\gamma_{LT} = 0.013$). 100 simulation trajectories for each γ_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

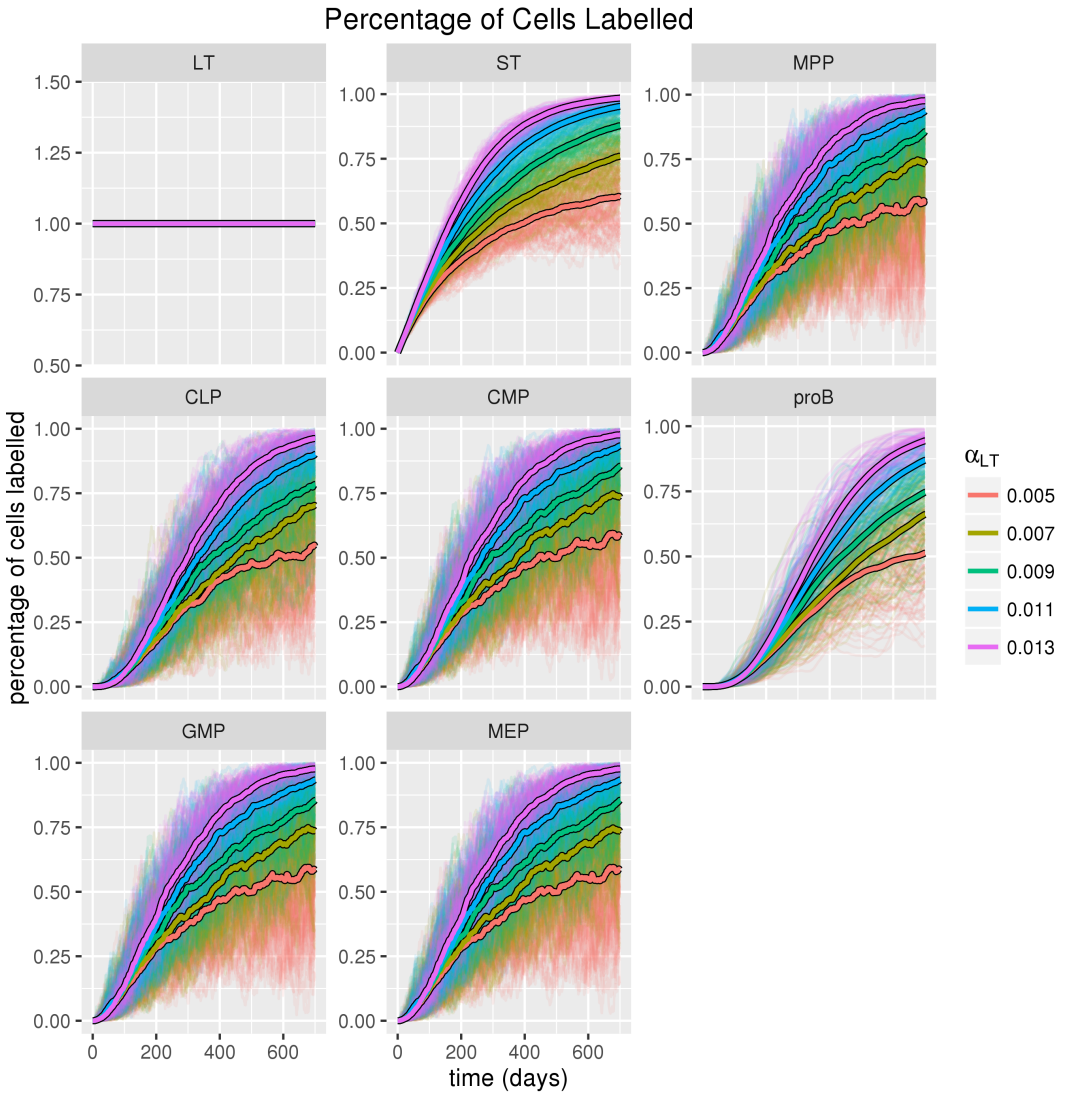

Figure B.7: Trajectories of the fraction of cells in each population that express a label over time are shown for varying α_{LT} rates: red ($\alpha_{LT} = 0.005$), orange ($\alpha_{LT} = 0.007$), green ($\alpha_{LT} = 0.009$), blue ($\alpha_{LT} = 0.011$), purple ($\alpha_{LT} = 0.013$). 100 simulation trajectories for each α_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

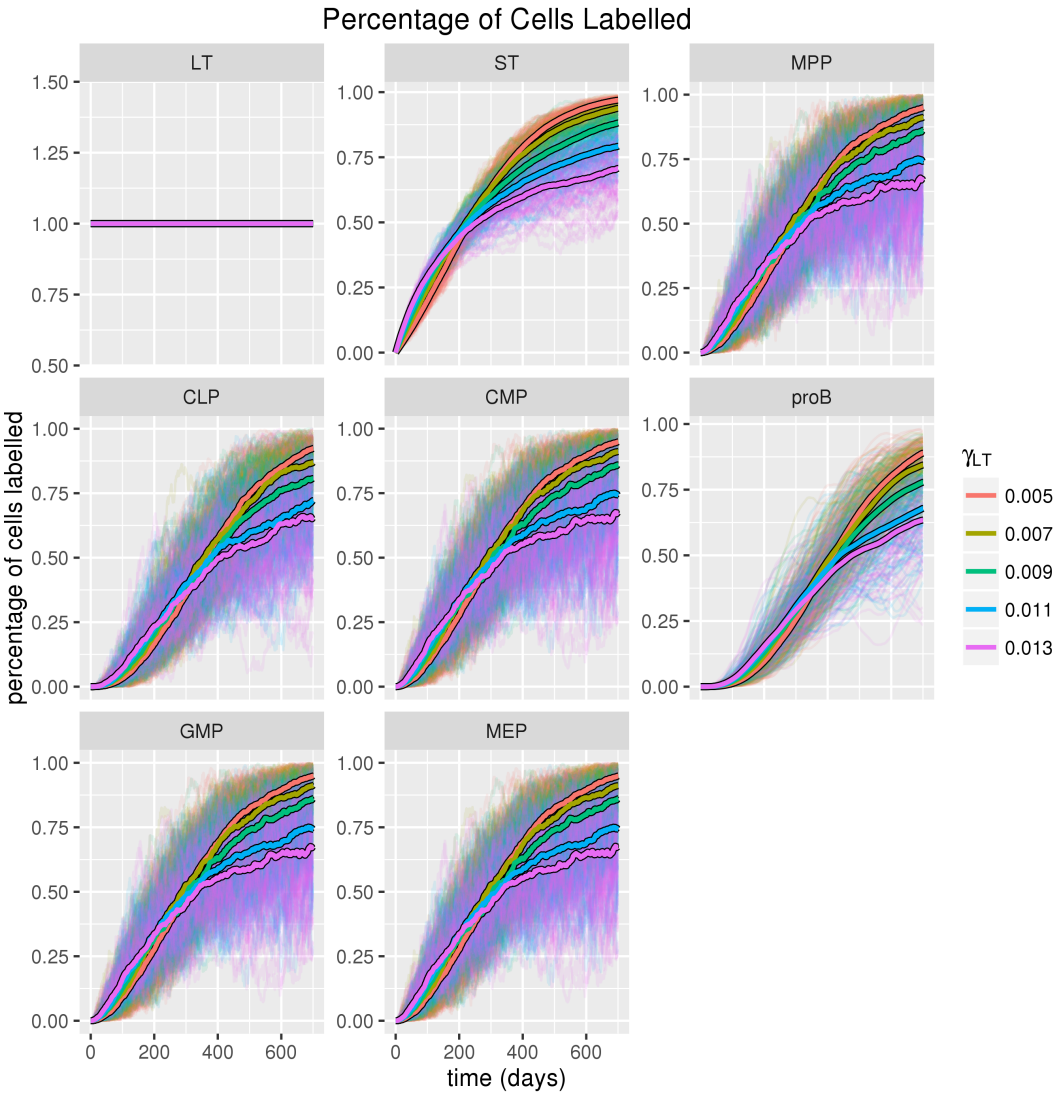

Figure B.8: Trajectories of the fraction of cells in each population that express a label over time are shown for varying γ_{LT} rates: red ($\gamma_{LT} = 0.005$), orange ($\gamma_{LT} = 0.007$), green ($\gamma_{LT} = 0.009$), blue ($\gamma_{LT} = 0.011$), purple ($\gamma_{LT} = 0.013$). 100 simulation trajectories for each γ_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

non's Equitability is Shannon's Diversity Index divided by its maximum, scaling the range of the equitability to $[0, 1]$. Here, we have once again barcoded all LT-HSC cells uniquely to start and not barcoded any downstream populations. We see simliar patterns here as we do in label uptake, with the addition of the effect of declining diversity in the $LT - HSC$ population. Note that this decline is not due to selection, as the introduction of a barcode has no fitness effects on the cells. The Shannon's Equitability over various α_{LT} and γ_{LT} rates are show in Figures B.9 and B.10 respectively.

DIFFpop is also capable of simulating how changes in fitness introduced by mutation can influence the dynamics of the system. Towards understanding clonal dynamics within a population, the user can specify how often to output a full census of the hierarchy. Here, we show the clonal dynamics from a single simulation where each population has a mutation probability of 1×10^{-7} per mitotic event and fitness changes are drawn from a double exponential distribution with equal slope parameter 1. In Figure B.11, the size of the colored bars correspond to the number of cells from each clone. Note in the following simulation, we have also increased our population size to the size of the hematopoietic system in adult mouse by setting $nLT = 17000$. We can introduce these changes by adding the following lines to our tree specification before calling the simulate function with *census* argument equal to 1.

```
# Add mutation events to self-renewing populations
addEdge(tree1, "LT", "LT", "mu", 1e-7)
addEdge(tree1, "ST", "ST", "mu", 1e-7)
addEdge(tree1, "MPP", "MPP", "mu", 1e-7)
addEdge(tree1, "CLP", "CLP", "mu", 1e-7)
```

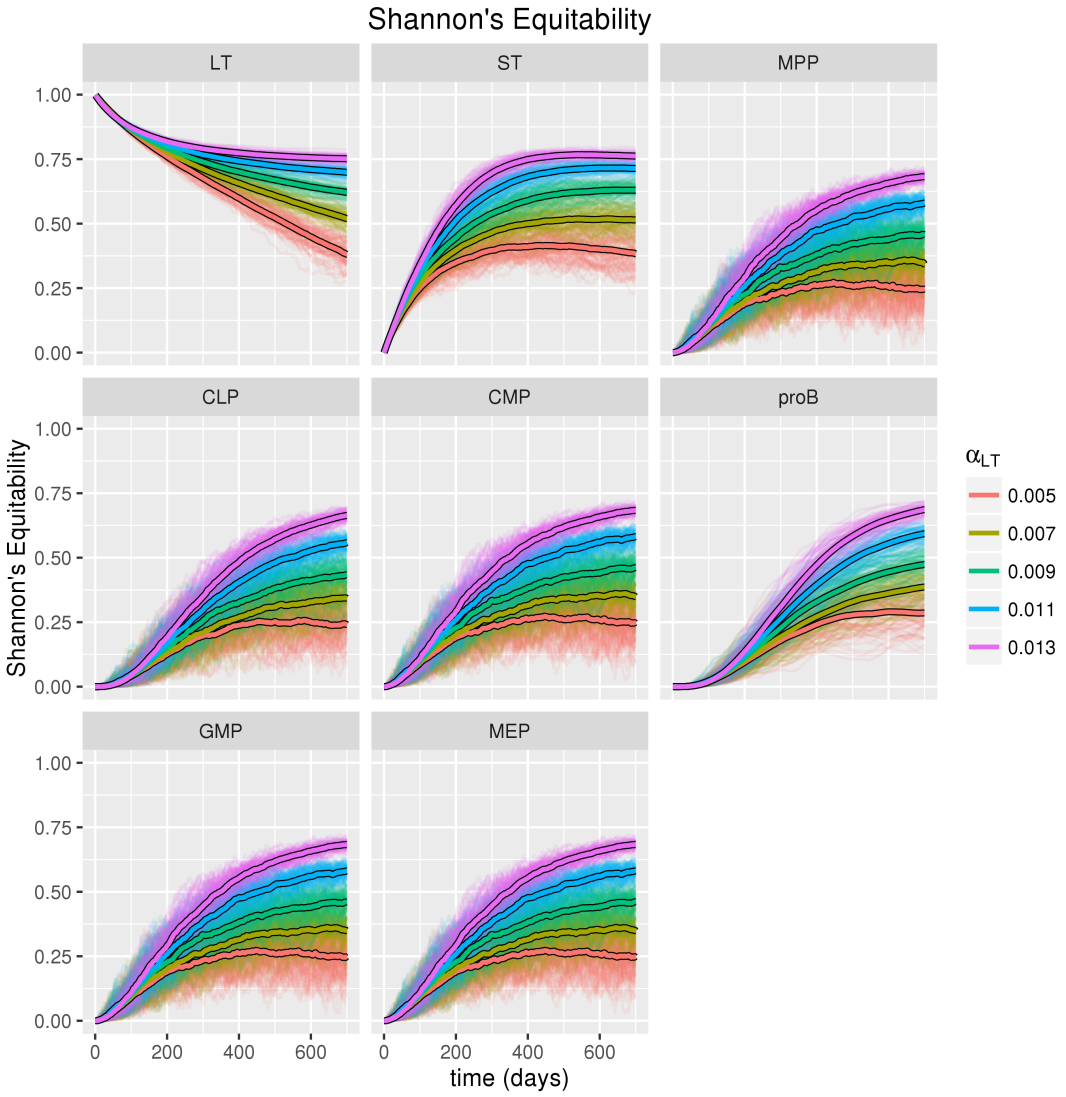

Figure B.9: Trajectories of Shannon's Equitability in each population that express a label over time are shown for varying α_{LT} rates: red ($\alpha_{LT} = 0.005$), orange ($\alpha_{LT} = 0.007$), green ($\alpha_{LT} = 0.009$), blue ($\alpha_{LT} = 0.011$), purple ($\alpha_{LT} = 0.013$). 100 simulation trajectories for each α_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

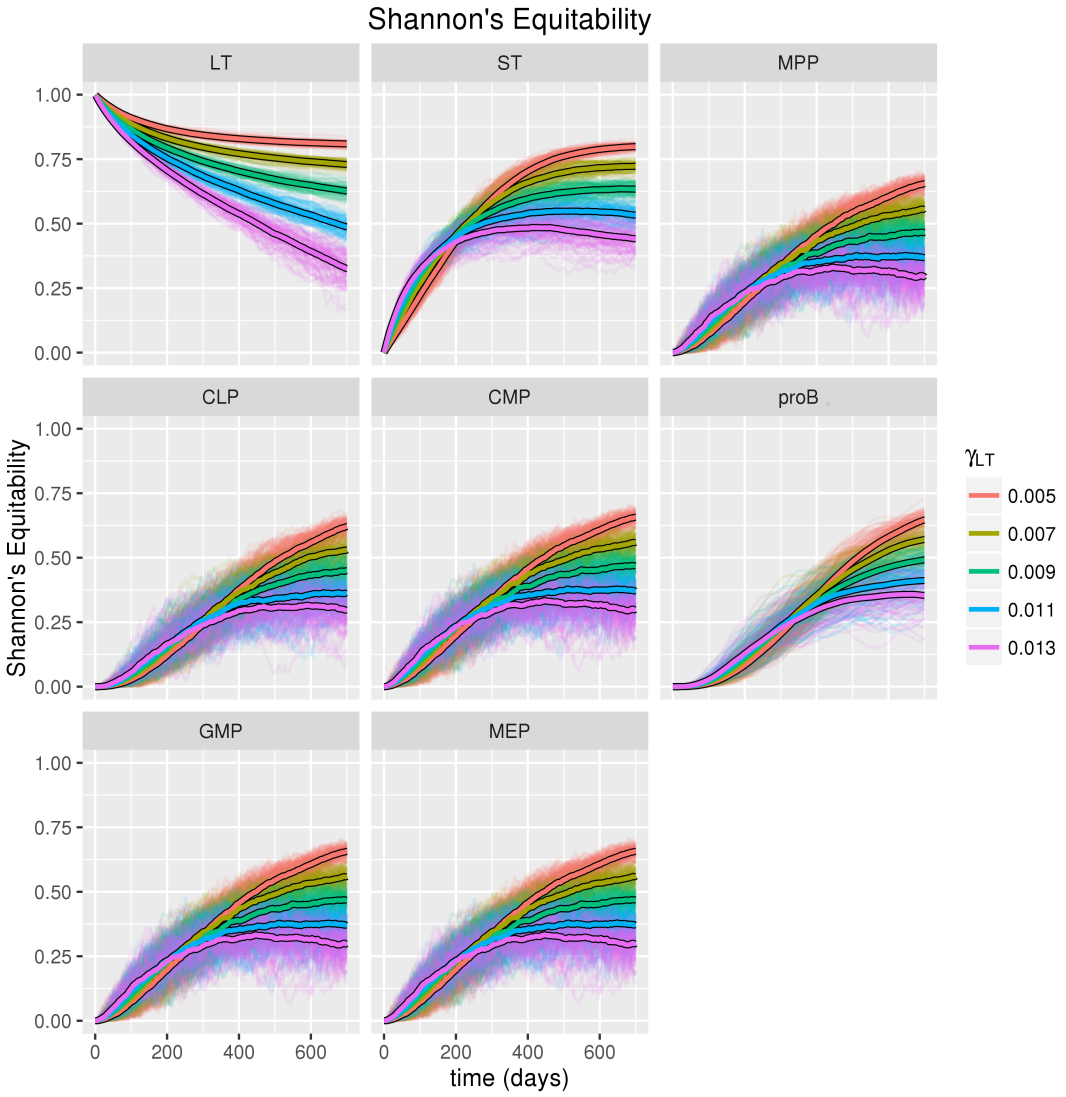

Figure B.10: Trajectories of Shannon's Equitability in each population that express a label over time are shown for varying γ_{LT} rates: red ($\gamma_{LT} = 0.005$), orange ($\gamma_{LT} = 0.007$), green ($\gamma_{LT} = 0.009$), blue ($\gamma_{LT} = 0.011$), purple ($\gamma_{LT} = 0.013$). 100 simulation trajectories for each γ_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

```
addEdge(tree1, "CMP", "CMP", "mu", 1e-7)

# Add a fitness distribution for those mutations
setFitnessDistribution(tree = tree1,
        distribution = "doubleexp",
        alpha_fitness = 1,
        beta_fitness = 1,
        pass_prob = 0,
        upper_fitness = NA,
        lower_fitness = 0)
```

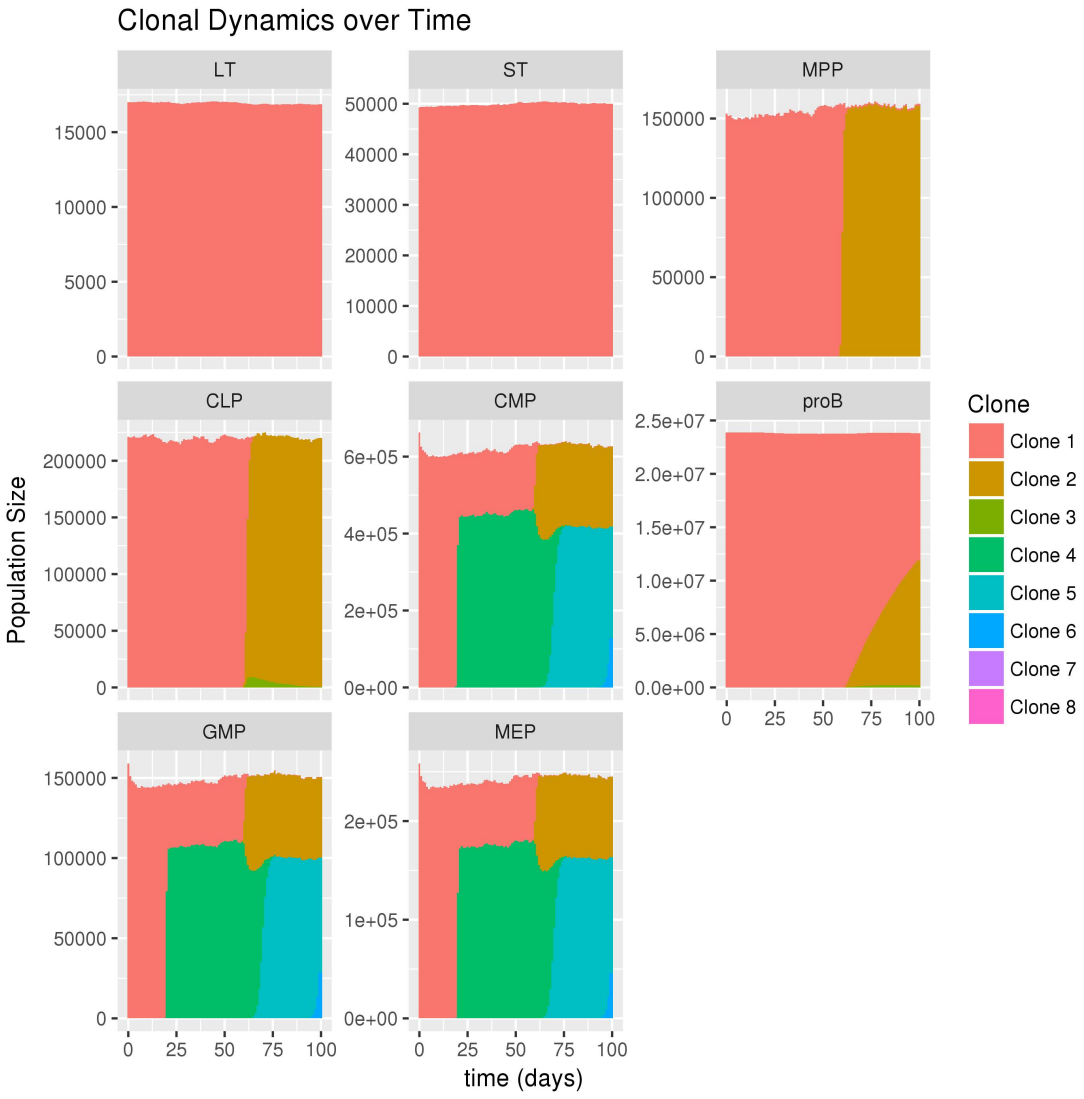

Figure B.11: The above plot displays the size of each clone as a different color over time. New clones are initiated by mutation during a mitosis event, with a change in fitness drawn from a double exponential distribution with equal positive and negative slope parameter of 1. Simulations were performed for 100 days at the system size in adult mouse with 17,000 LT-HSCs.

B.3 Vignette 2: Multi-type Moran Process

Background

We can also model the system assuming equilibrium has been reached and no population within the hierarchy is experiencing any significant change in size. Towards that end, let us consider adapting our model to use FixedPop and DiffTriangle structures, which guarantee that at every time point, the size of each population remains stable. This model is shown in Figure B.12.

Using DIFFpop in R

In R using the appropriate DIFFpop functions, we first specify the populations of the tree. To do this, we determine which of the three basic DIFFpop class is appropriate, give the population a name, initial population size, and initial population barcoding efficiency (what proportion of cells are given a unique barcode simulation initialization). Here, we use FixedPops and DiffTriangles. Again, we only uniquely barcode the LT-HSC population.

```
library(diffpop)

nLT = 500

# Create a DiffTree object
tree2 = DiffTree()

# Add FixedPops for each population to tree2
```

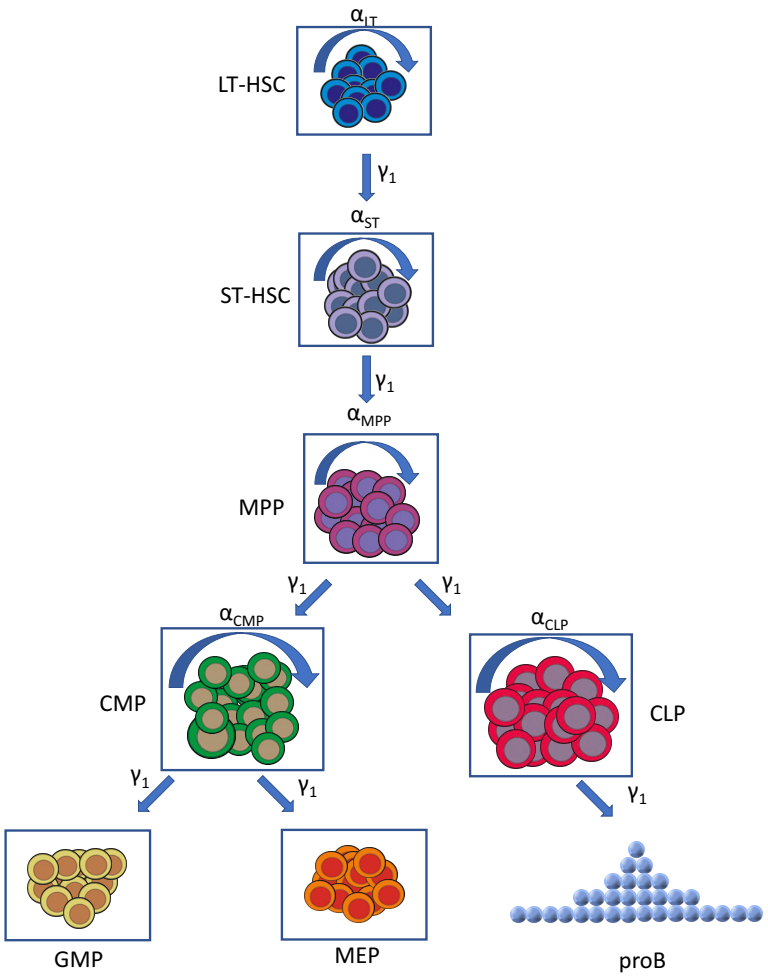

Figure B.12: Hematopoietic System model using FixedPops (boxes) and DiffTriangle (pro-B population). Abbreviations: long-term hematopoietic stem cell (LT-HSC), short-term hematopoietic stem cell (ST-HSC), multi-potent progenitor (MPP), common myeloid progenitor (CMP), common lymphoid progenitor (CLP), granulocyte-macrophage progenitor (GMP), megakaryocyte-erythroid progenitor (MEP). Events between populations consist of mitosis (α) and mitosis-independent differentiation (γ_1).

```
FixedPop(tree2, "LT", nLT, 1.0)
FixedPop(tree2, "ST", 2.9*nLT, 0.0)
FixedPop(tree2, "MPP", 9*nLT, 0.0)

FixedPop(tree2, "CLP", 13*nLT, 0.0)
FixedPop(tree2, "CMP", 39*nLT, 0.0)

FixedPop(tree2, "GMP", as.integer(0.24*39*nLT), 0.0)
FixedPop(tree2, "MEP", as.integer(0.39*39*nLT), 0.0)

# Add a DiffTriangle type for the proB cells
DiffTriangle(tree2, "proB", height = 6, first_level = 2*13*nLT)
```

We can then use the addEdge function to specify links between the populations. We will be using the point estimates from (Busch, 2015) as our event rates. As before, a parameterization for net proliferation is used, which is the self-renewal/mitosis rate minus the cell death rate. Here, we set this value as our mitotic rate; however, we could add any positive value to both the alpha and delta (death) event rate to remain within the correct parameterization.

```
# Add self-renewal events
addEdge(tree2, "LT", "LT", "alpha", 0.009)
addEdge(tree2, "ST", "ST", "alpha", 0.042)
addEdge(tree2, "MPP", "MPP", "alpha", 4)
addEdge(tree2, "CLP", "CLP", "alpha", 3.00)
addEdge(tree2, "CMP", "CMP", "alpha", 4)

# Add differentiation events
addEdge(tree2, "LT", "ST", "gamma1", 0.009)
addEdge(tree2, "ST", "MPP", "gamma1", 0.045)
addEdge(tree2, "MPP", "CLP", "gamma1", 0.022)
addEdge(tree2, "MPP", "CMP", "gamma1", 3.992)
addEdge(tree2, "CLP", "proB", "gamma1", 2.000)
```

```
addEdge(tree2, "CMP", "GMP", "gamma1", 2)
addEdge(tree2, "CMP", "MEP", "gamma1", 3)
```

To initiate a simulation of the specified tree, the last steps are to first specify which population is the root of the tree (the population that is furthest upstream), write our tree input files to a specified location, and then start the simulation using the simulateTree function, this time with the fixed parameter set to TRUE.

```
# Set the root population and simulate
setRoot(tree2, "LT")
simulateTree(tree = tree2,
        fixed = TRUE,
        time = 700,
        indir = "example/",
        outdir = "example/")
```

Results

Let us begin by looking at the population sizes over time for various α_{LT} and γ_{LT} rates. Because we are using structures that maintain a constant population size, we see no fluctuation in the population sizes over time. These population size over time plots for various α_{LT} and γ_{LT} rates are shown in Figures B.13 and B.14 respectively.

In Figure B.15, looking at the fraction of cells that have a barcode for various α_{LT} and γ_{LT} rates, we observe how our simulation procedure for fixed populations differs from our growing populations. Across various α_{LT} rates, we see the same trajectory for the fraction of cells

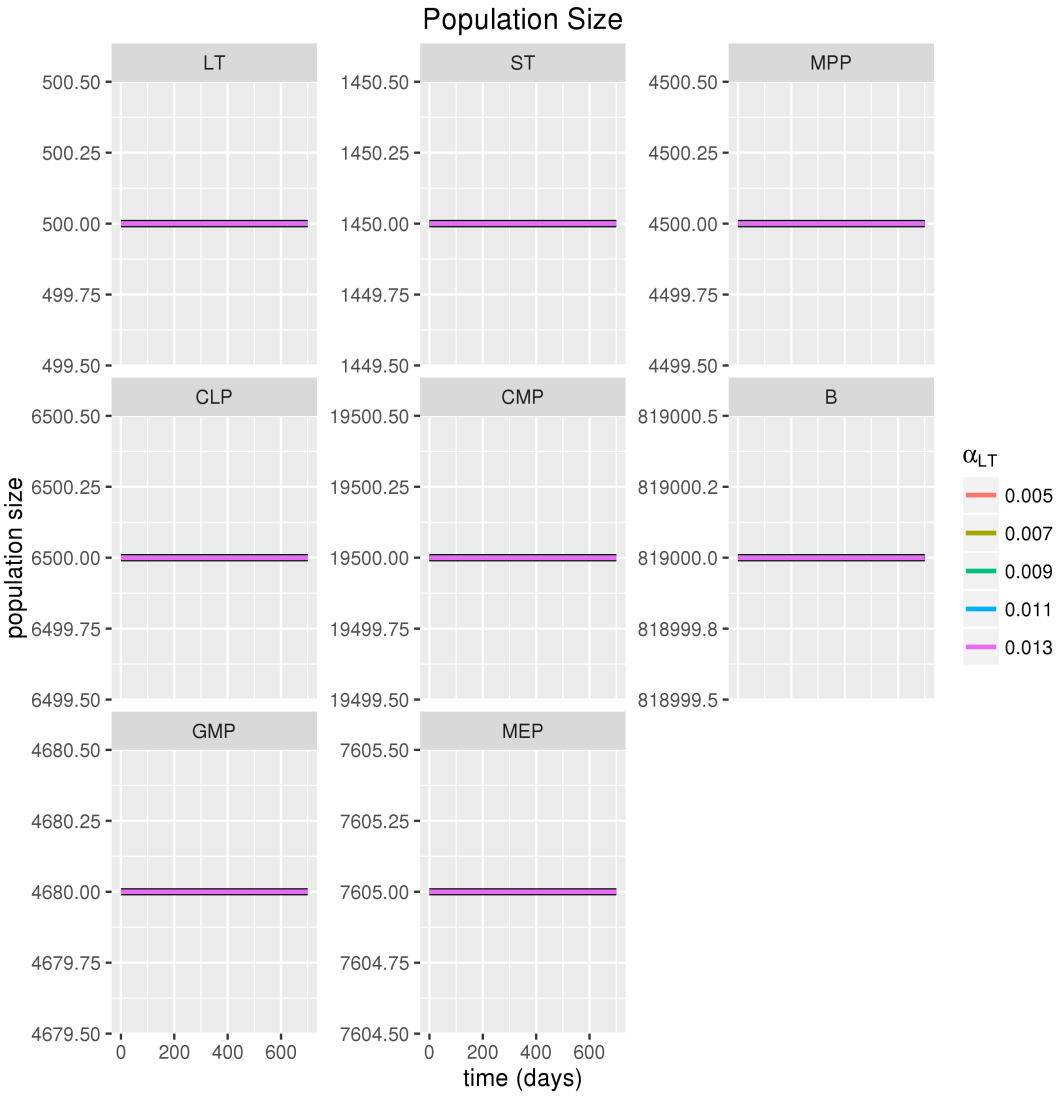

Figure B.13: Trajectories of population size over time are shown for varying α_{LT} rates: red ($\alpha_{LT} = 0.005$), orange ($\alpha_{LT} = 0.007$), green ($\alpha_{LT} = 0.009$), blue ($\alpha_{LT} = 0.011$), purple ($\alpha_{LT} = 0.013$). 100 simulation trajectories for each α_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above. Note: all trajectories plotted are flat and hence only the last trajectory plotted (purple) is visible.

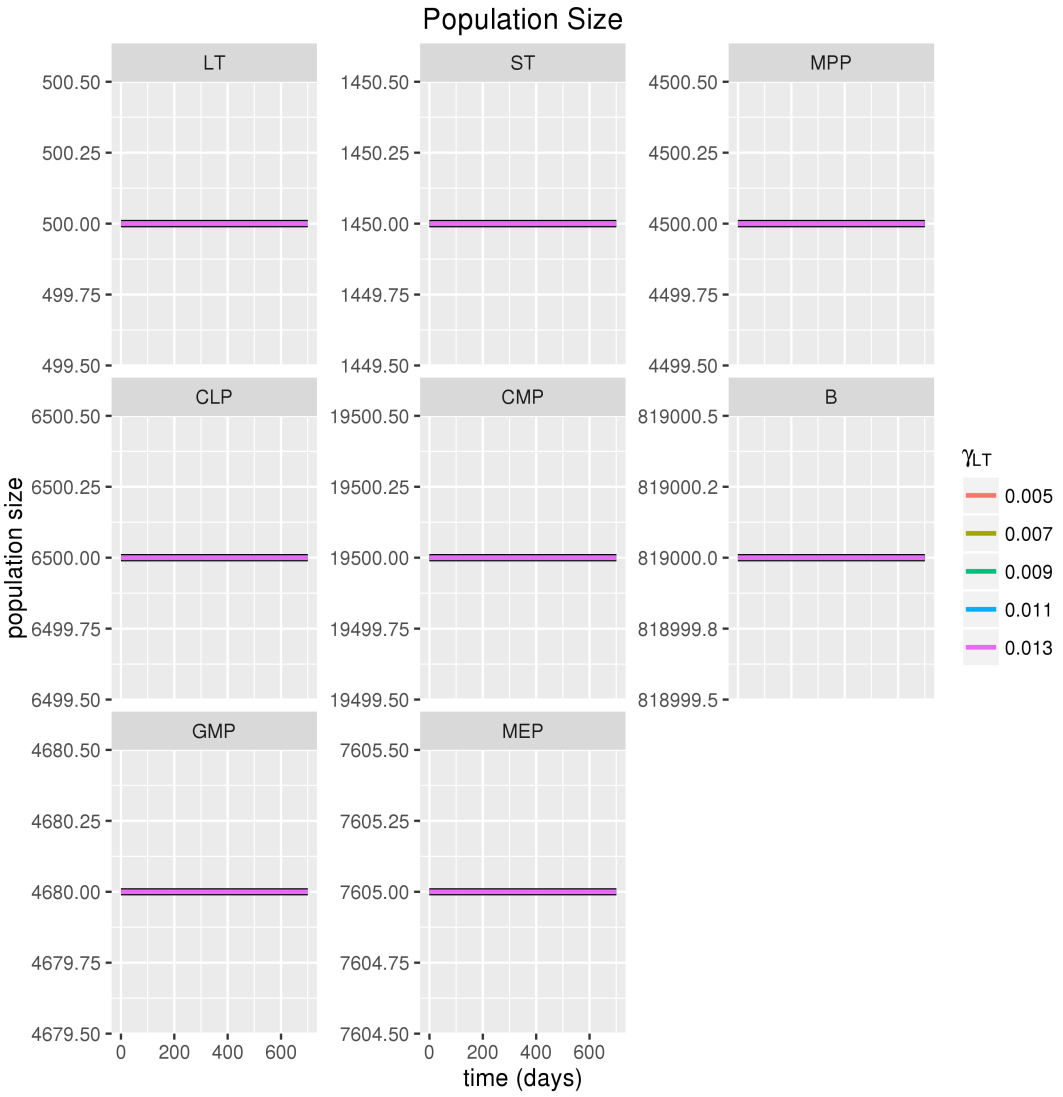

Figure B.14: Trajectories of population size over time are shown for varying γ_{LT} rates: red ($\gamma_{LT} = 0.005$), orange ($\gamma_{LT} = 0.007$), green ($\gamma_{LT} = 0.009$), blue ($\gamma_{LT} = 0.011$), purple ($\gamma_{LT} = 0.013$). 100 simulation trajectories for each γ_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above. Note: all trajectories plotted are flat and hence only the last trajectory plotted (purple) is visible.

that have a barcode. This is because when we change our α_{LT} rate, the fixed simulation automatically adjusts the net proliferation, adjusting either α_{LT} or δ_{LT}, to maintain a constant population size. Because we are not adjusting the differentiation rate downstream γ_{LT}, we experience the same level of differentiation of barcoded cells for all levels of α_{LT}.

We also varied the mitosis-independent differentiation rate from LT-HSC to ST-HSC, γ_{LT}. Across various γ_{LT} rates in Figure B.16, we observe trajectories that match intuition. The higher the differentiation rate to downstream populations, the more barcoded cells appear in the downstream populations. In this case, because the LT-HSC population size is fixed, we do not observe a decline in the fraction of barcoded cells for higher γ_{LT} rates like we did in the branching process model.

As with the branching process model, we can introduce mutations into our fixed population model and track the clonal dynamics over time. Here in Figure B.17, we show the clonal dynamics from a single simulation where each population has a mutation probability of 1×10^{-7} per mitotic event and fitness changes are drawn from a double exponential distribution with equal slope parameter 1, where the size of the colored bars represent the number of cells from each clone. Notice the that total height of each bar remains constant over time, as expected. Once again, we can make these changes to *tree*2 before simulating:

```
# Add mutation events for types capable of self-renewal
addEdge(tree2, "LT", "LT", "mu", 1e-7)
addEdge(tree2, "ST", "ST", "mu", 1e-7)
addEdge(tree2, "MPP", "MPP", "mu", 1e-7)
addEdge(tree2, "CLP", "CLP", "mu", 1e-7)
```

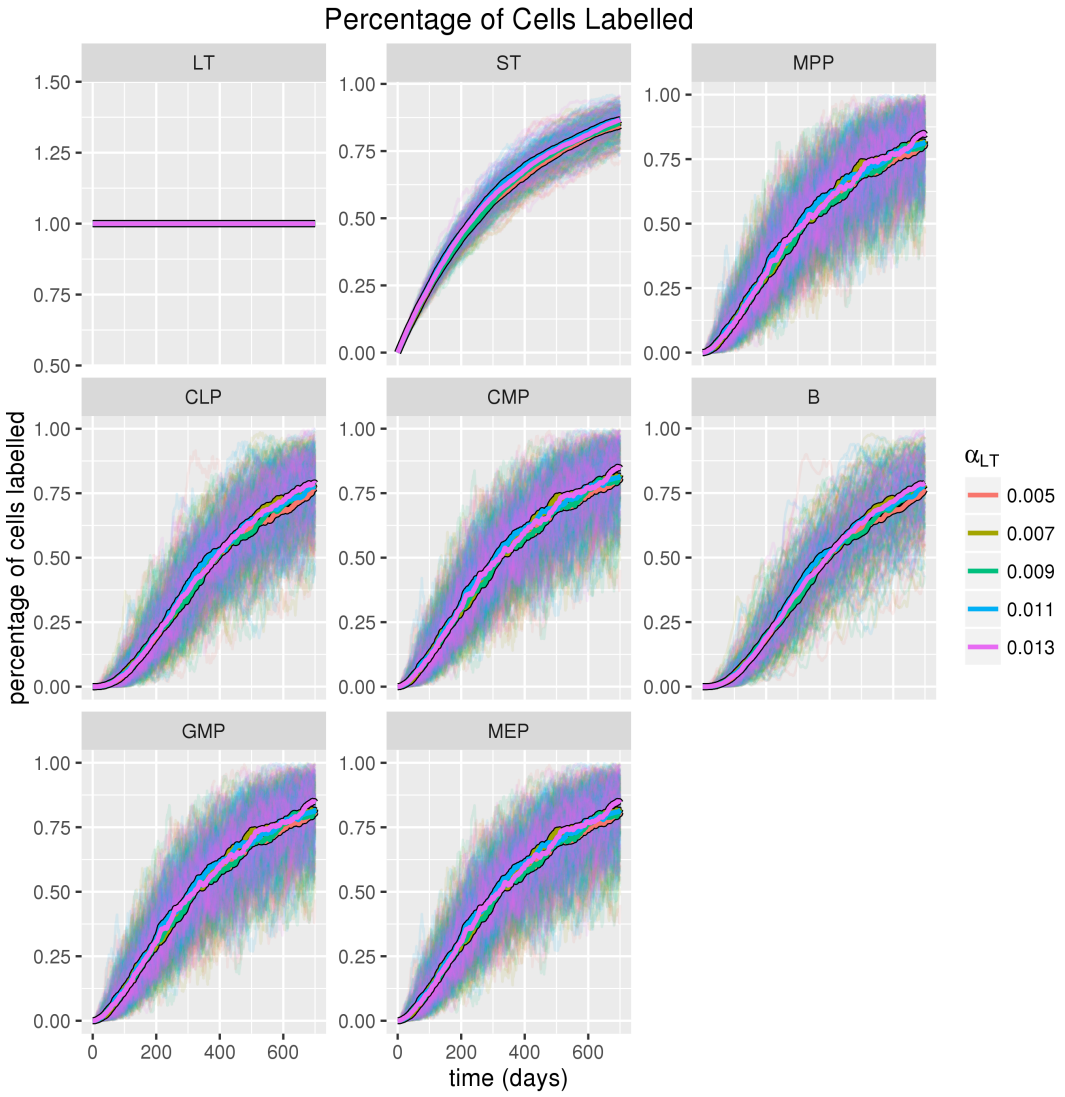

Figure B.15: Trajectories of the fraction of cells in each population that express a label over time are shown for varying α_{LT} rates: red ($\alpha_{LT} = 0.005$), orange ($\alpha_{LT} = 0.007$), green ($\alpha_{LT} = 0.009$), blue ($\alpha_{LT} = 0.011$), purple ($\alpha_{LT} = 0.013$). 100 simulation trajectories for each α_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

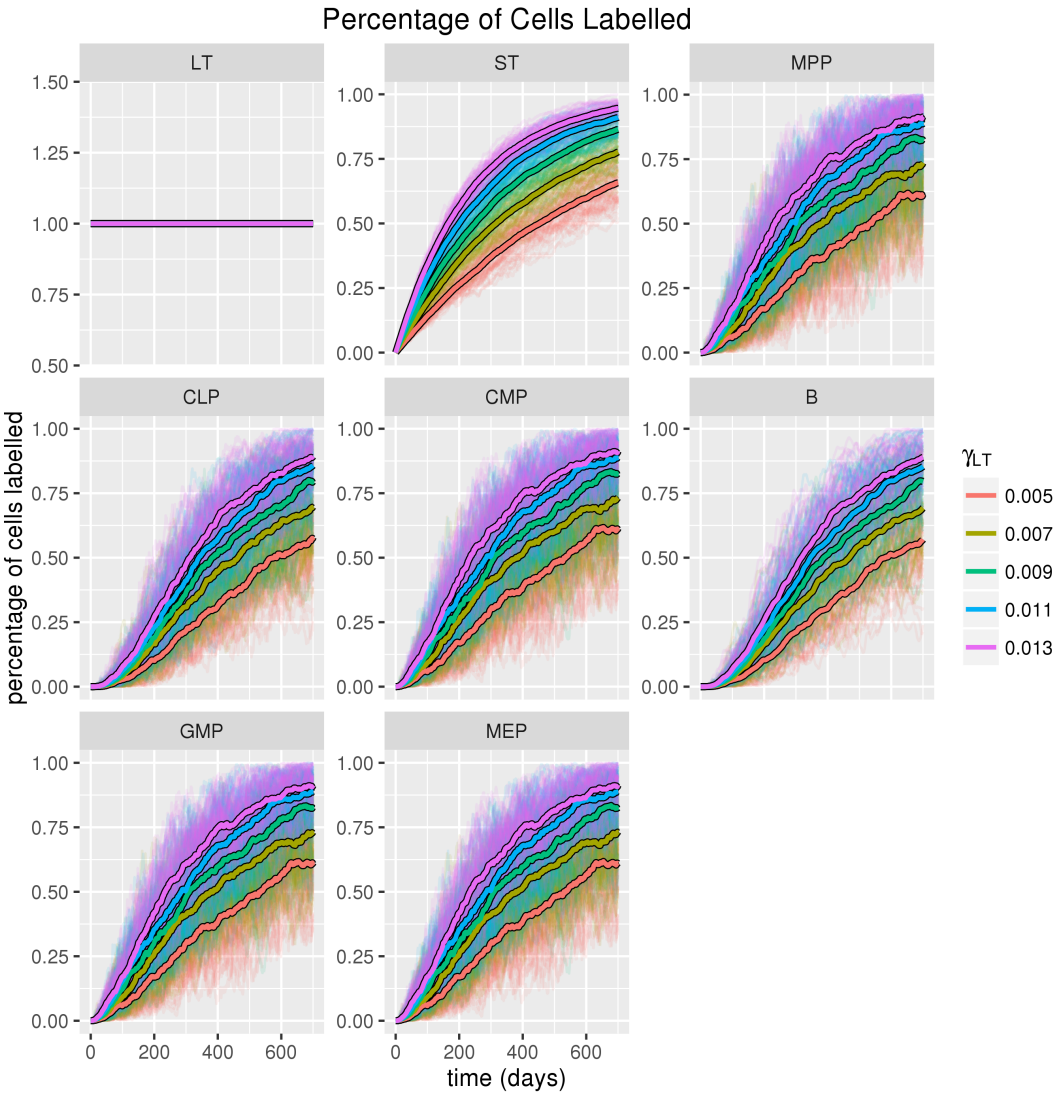

Figure B.16: Trajectories of the fraction of cells in each population that express a label over time are shown for varying γ_{LT} rates: red ($\gamma_{LT} = 0.005$), orange ($\gamma_{LT} = 0.007$), green ($\gamma_{LT} = 0.009$), blue ($\gamma_{LT} = 0.011$), purple ($\gamma_{LT} = 0.013$). 100 simulation trajectories for each γ_{LT} rate are plotted as well as a bold mean trajectory. Simulations were run for 700 days, around the average lifespan for a mouse using all other parameters as shown in code excerpts above.

```
addEdge(tree2, "CMP", "CMP", "mu", 1e-7)

# Add a fitness distribution for those mutations
setFitnessDistribution(tree = tree2,
        distribution = "doubleexp",
        alpha_fitness = 1,
        beta_fitness = 1,
        pass_prob = 0,
        upper_fitness = NA,
        lower_fitness = 0)
```

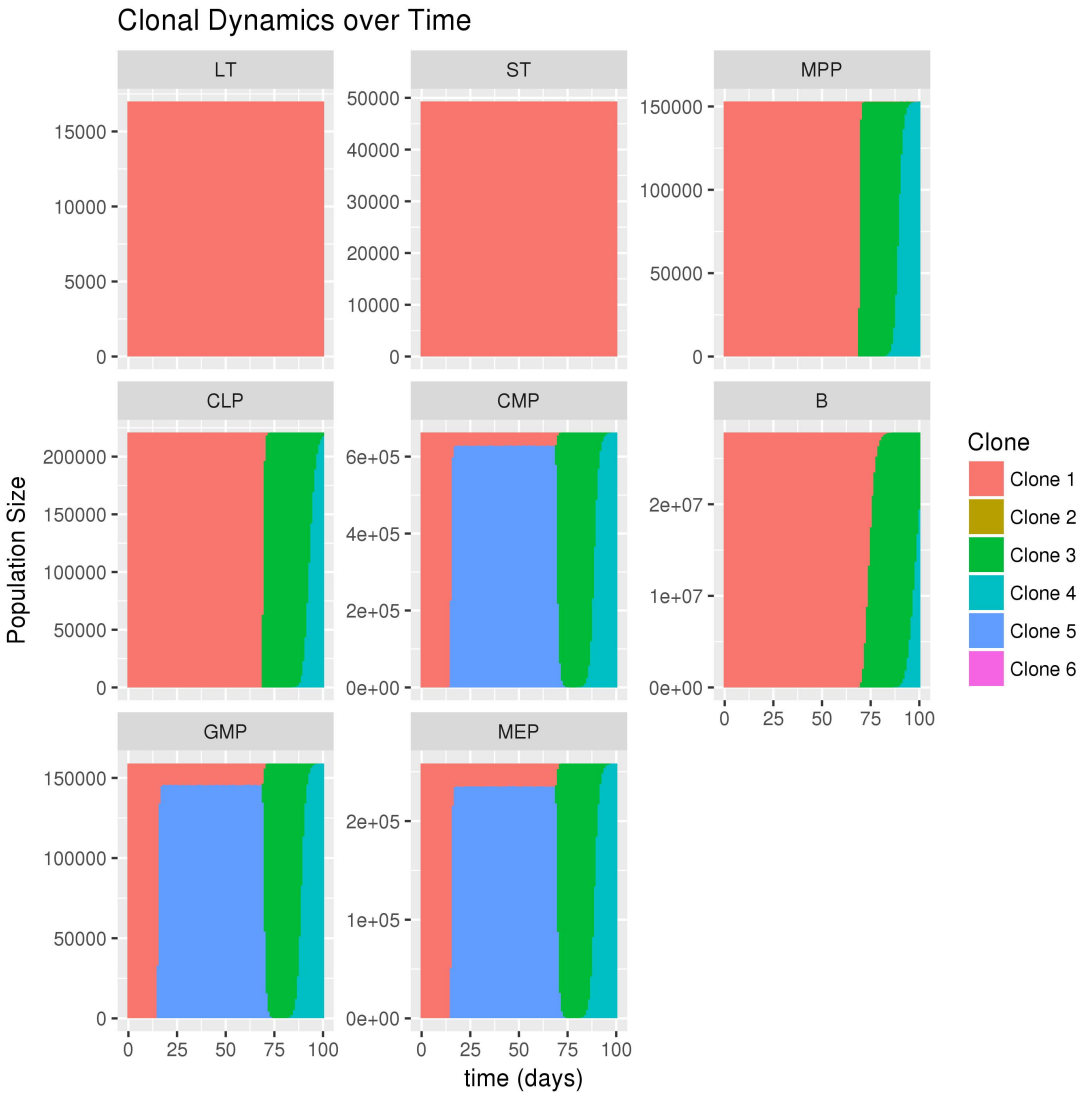

Figure B.17: The above plot displays the size of each clone as a different color over time. New clones are initiated by mutation during a mitosis event, with a change in fitness drawn from a double exponential distribution with equal positive and negative slope parameter equal to 1. Simulations were performed for 100 days at the true system size starting with 17,000 LT-HSCs.

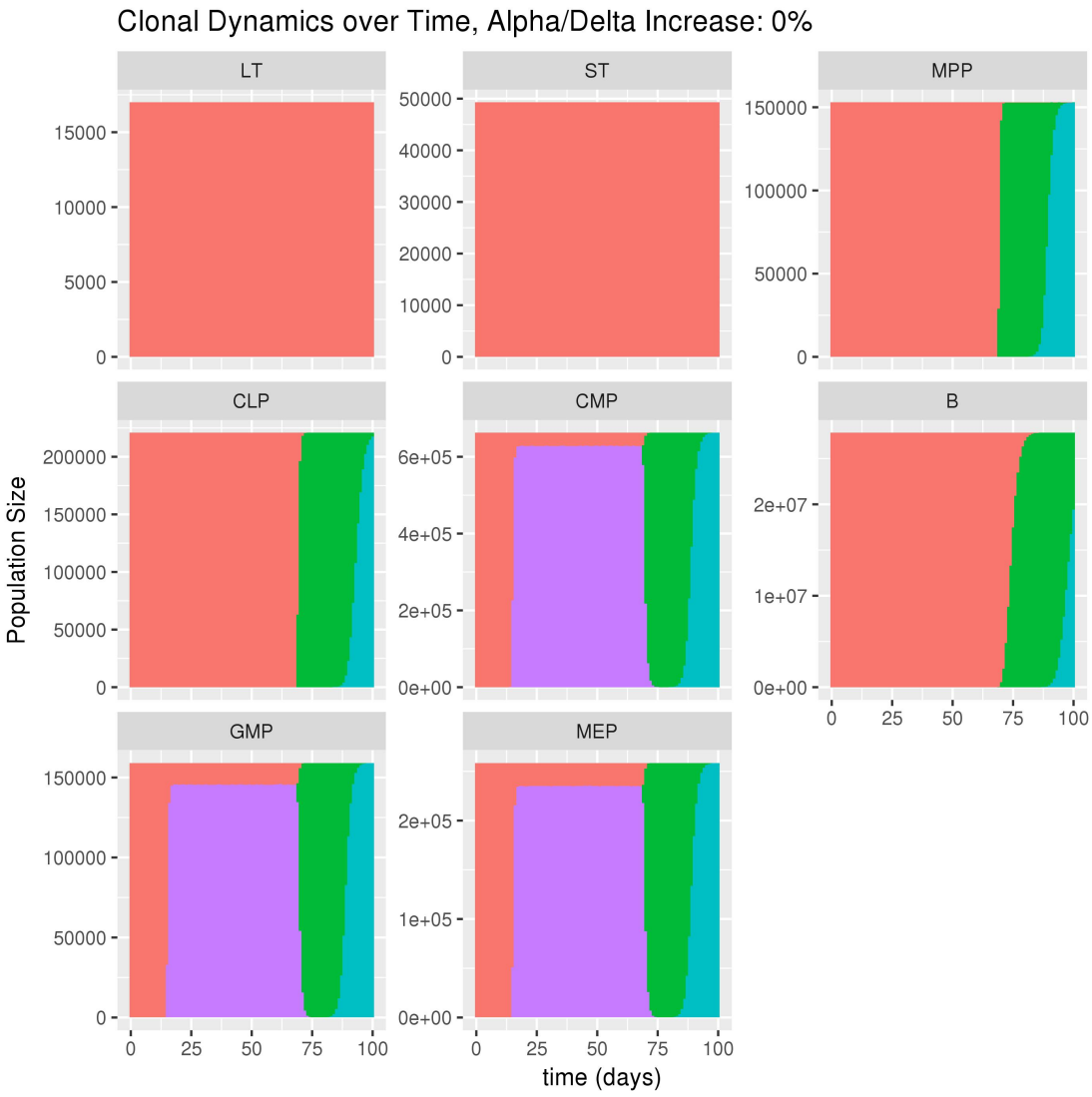

Figure B.18: No increase in either α or δ for any population. Clones sizes are plotted using different colors for each clone over time. At this increase level, few clones arise in all populations. Simulations were run at the true system size beginning with 17,000 LT-HSCs for 100 days.

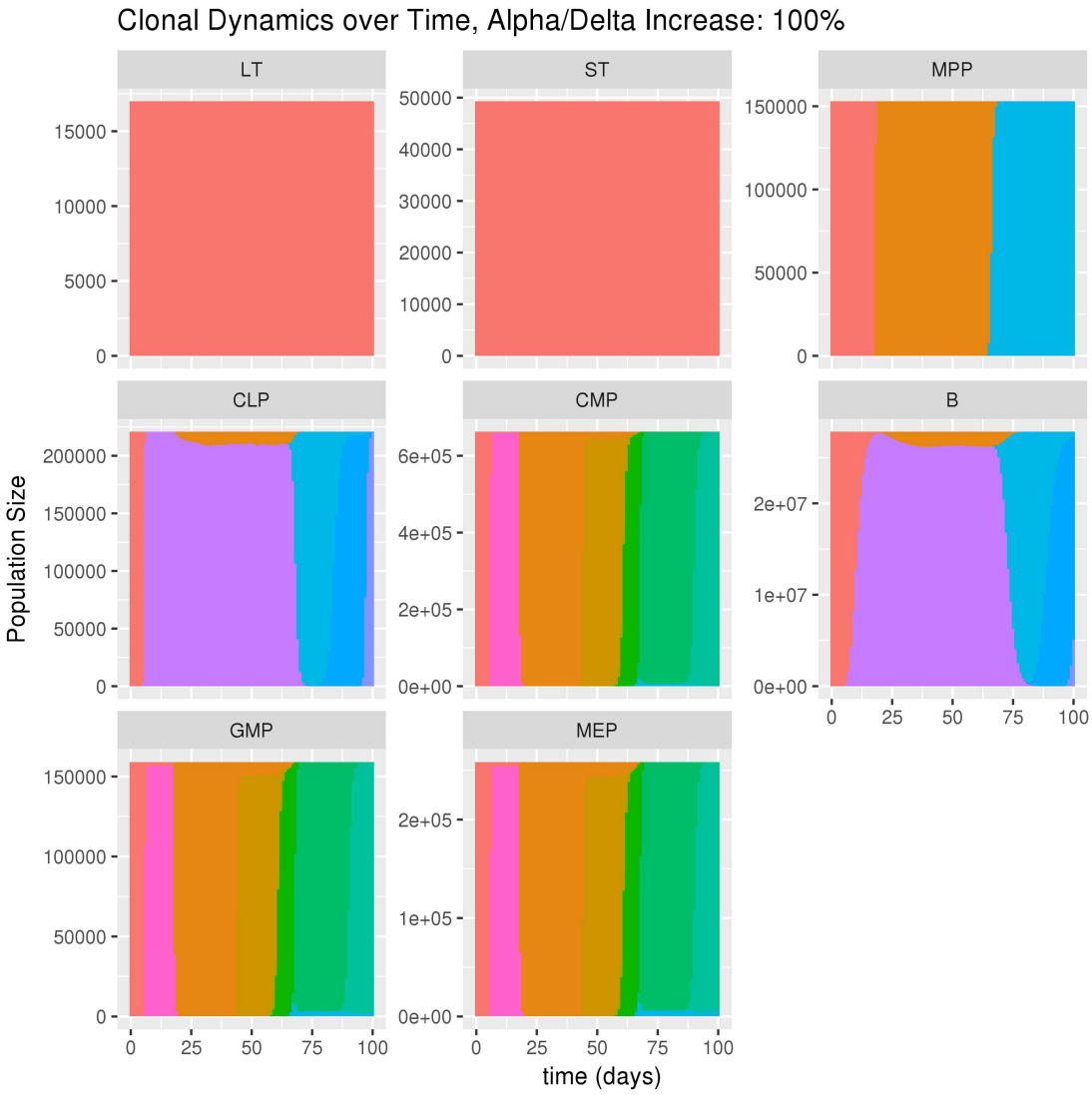

Figure B.19: 100% increase in both α and δ for all populations. Clones sizes are plotted using different colors for each clone over time. At this increase level, many clones arise and fix in the quickly proliferating populations (CMP, GMP, MEP). Simulations were run at the true system size beginning with 17,000 LT-HSCs for 100 days.

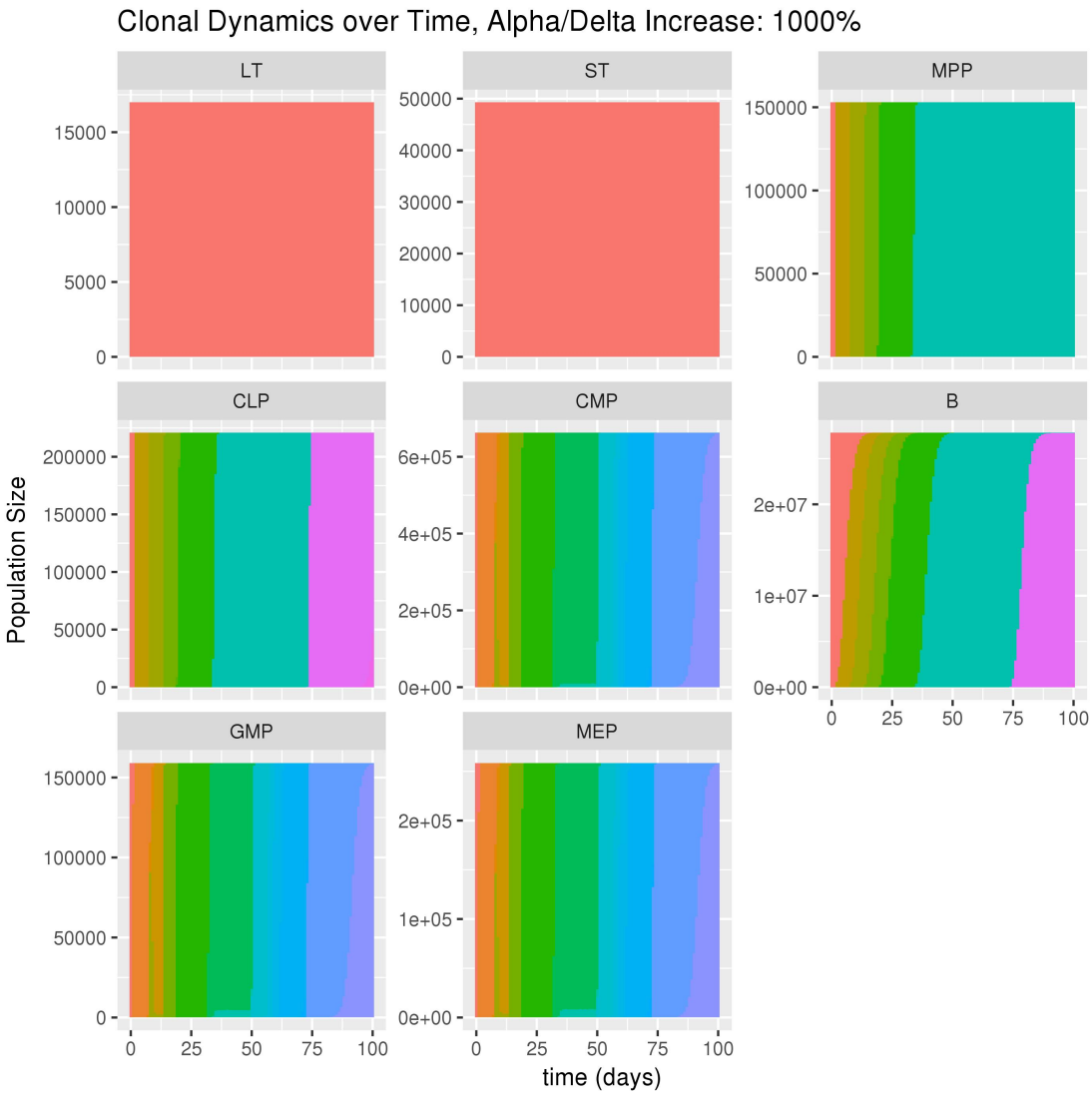

Figure B.20: 1000% increase in both α and δ for all populations. Clones sizes are plotted using different colors for each clone over time. At this increase level, many clones arise and fix in the quickly in all downstream populations. Simulations were run at the true system size beginning with 17,000 LT-HSCs for 100 days.

As we stated earlier, Busch et al. parameterize a joint net proliferation rate ($\alpha - \delta$) for each population, meaning that if we increase both α and δ by the same amount for a population, we remain within this parameterization[16]. Doing this changes both the amount of mutations that occur and the amount of time it takes for a mutation to fix in the population. Because there are more mitotic divisions in a time unit, the number of mutations increase with increasing α. Also, because there are more mitosis events, there is more opportunity for selection events to favor dominating, high-fitness clones, resulting in less time for a clone to fix in the population. In Figures B.18, B.19, and B.20, we show the clonal dynamics bar plots for three scenarios all of which fit the net proliferation parameter for each population, only we have increased α and δ for each population by 100% (Figure B.19) and 1000% (Figure B.20).

C

ESTIpop Supplemental Materials

C.1 Methods

Previous results (Yakovlev and Yanev, 2009)

The following results are from Yakovlev and Yanev[79] which we adapt to our current situation and repeat for completeness. Consider for $d \in \mathbb{N}^+$, the d-type branching process $\{\boldsymbol{Z}^{(i)}(t), t \geq 0, i = 1\ldots, d\}$, where at time t, $\boldsymbol{Z}^{(i)}(t)$ is a vector of size d with each component $Z_j^{(i)}(t)$ count-

ing the number of type j individuals alive at time t assuming the process is initiated with 1 type i individual. The probability generating function of this process given a single ancestor is

$$F^{(i)}(t;\boldsymbol{s}) \equiv \mathbb{E}\left[\boldsymbol{s}^{\boldsymbol{Z}^{(i)}(t)}|Z_i^{(i)}(0)=1\right].$$

Using the branching property and independence of individuals, we see that the generating function for the d-type branching process $\{\boldsymbol{Z}^{(i)}(t;N_i), t \geq 0\}$, where at time t, $\boldsymbol{Z}^{(i)}(t;N_i)$ is a vector of size d with each component $Z_j^{(i)}(t)$ counting the number of type j individuals alive at time t assuming the process is initiated with N_i type i individuals is

$$F_{N_i}^{(i)}(t;\boldsymbol{s}) \equiv \mathbb{E}\left[\boldsymbol{s}^{\boldsymbol{Z}(t)}|Z_i^{(i)}(0)=N_i\right] = \left[F^{(i)}(t;\boldsymbol{s})\right]^{N_i}$$

Basic properties of probability generating functions allow us to define the moments in terms of the derivatives of the p.g.f., so we define the first two moments for each $i = 1,\ldots,d$ as such:

$$\begin{aligned}
m_k^{(i)}(t) &\equiv \mathbb{E}\left[Z_k^{(i)}(t)|Z_i^{(i)}(0)=1\right] = \frac{\partial}{\partial s_k}F^{(i)}(t;\boldsymbol{s})|_{\boldsymbol{s}=\boldsymbol{1}}, \quad k=1,2,\ldots,d \\
b_{jk}^{(i)}(t) &= \frac{\partial^2}{\partial s_j \partial s_k}F^{(i)}(t;s)|_{\boldsymbol{s}=\boldsymbol{1}}, \quad j,k \in \{1,2,\ldots,d\} \\
\sigma_k^{(i)}(t)^2 &= \text{Var}\left[Z_k^{(i)}(t)\right] = b_{kk}^{(i)}(t) + m_k^{(i)}(t) - m_k^{(i)}(t)^2 \\
C_{jk}^{(i)}(t) &\equiv \text{Cov}\left[Z_j^{(i)}(t), Z_k^{(i)}(t)\right] = b_{jk}^{(i)}(t) - m_j^{(i)}(t)m_k^{(i)}(t), \quad j,k \in \{1,2,\ldots,d\}, j \neq k
\end{aligned}$$

Also note that the covariance between processes initiated by two different ancestors are independent, so

$$\text{Cov}\left[Z_j^{(i)}(t), Z_k^{(l)}(t)\right] = 0$$

since two processes beginning with different ancestors are independent. We assume that the covariance matrix, $\mathbf{C}^{(i)}(t) \equiv [C_{jk}^{(i)}(t)]_{d\times d}$ is finite and all diagonal elements are strictly positive, or $\sigma_k^i(t)^2 > 0$.

If we define $Z_j^{(i)}(t; N_i)$ as the number of type j individuals at time t evolving from a process beginning with N_i type i individuals, then each individual gives rise to its own independent process, so

$$Z_j^{(i)}(t; N_i) = \sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t)$$

where $Z_{j(k)}^{(i)}(t)$ is the number of type j individuals in the k^{th} i.i.d. ancestor of the process.

Define the random variable

$$V_j^{(i)}(t; N_i) \equiv \frac{\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t) - N_i m_j^{(i)}(t)}{\sigma_j^{(i)}(t)\sqrt{N_i}}$$

as the process that counts the number of individuals centered and scaled by its mean and variance respectively.

Note that by construction, $\mathbb{E}\left[V_j^{(i)}(t;N_i)\right] = 0$ since

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t)\right] &= N_i \times \mathbb{E}\left[Z_j^{(i)}(t)\right] \\
&= N_i \times \mathbb{E}\left[Z_j^{(i)}(t) | Z_i^{(i)}(0) = 1\right] \\
&= N_i \times m_j^{(i)}(t)
\end{aligned}$$

and $\text{Var}\left[V_j^{(i)}(t;N_i)\right] = 1$:

$$\begin{aligned}
\text{Var}\left[V_j^{(i)}(t;N_i)\right] &= \text{Var}\left[\frac{\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t) - N_i m_j^{(i)}(t)}{\sigma_j^{(i)}(t)\sqrt{N_i}}\right] \\
&= \frac{1}{\sigma_j^{(i)}(t)^2 N_i} \text{Var}\left[\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t) - N_i m_j^{(i)}(t)\right] \\
&= \frac{1}{\sigma_j^{(i)}(t)^2 N_i} \text{Var}\left[\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t)\right] \\
&= \frac{1}{\sigma_j^{(i)}(t)^2 N_i} \sum_{k=1}^{N_i} \text{Var}\left[Z_j^{(i)}(t)\right] \\
&= \frac{1}{\sigma_j^{2(i)}(t) N_i} \sum_{k=1}^{N_i} \sigma_j^{(i)}(t)^2 \\
&= \frac{1}{\sigma_j^{(i)}(t)^2 N_i} N_i \sigma_j^{(i)}(t)^2 \\
&= 1
\end{aligned}$$

By the Central Limit Theorem, $V_j^{(i)}(t;N_i) \xrightarrow{d} N(0,1)$ as $N_i \to \infty$.

Thus, by the Central Limit Theorem for i.i.d. vectors[79],

$$\left(V_1^{(i)}(t;N_i), V_2^{(i)}(t;N_i), \ldots, V_d^{(i)}(t;N_i)\right) \to \left(Y_1^{(i)}(t), Y_2^{(i)}(t), \ldots, Y_d^{(i)}(t)\right) \equiv \mathbf{Y}^{(i)}(t)$$

which has a multivariate normal distribution with $\mathbb{E}\left[\mathbf{Y}^{(i)}(t)\right] = 0$, $\text{Var}\left[\mathbf{Y}^{(i)}(t)\right] = 1$, and

$$\text{Cov}\left(\mathbf{Y}^{(i)}(t)\right) = C_{jk}^{(i)}(t).$$

Ancestors from different types

We extend the previous work by considering a process with ancestors coming from different types. Suppose we begin a process with $N = \sum_{i=1}^{d} N_i$ ancestors split up into types such that there are N_1 type 1 ancestors, N_2 type 2 ancestors, ..., and N_d type d ancestors. Let us define $q_i = \frac{N_i}{N} > 0$ as the type proportions such that as $N \to \infty$, the type proportions remain constant, and define the vector $\boldsymbol{N} = (q_1 N, \ldots, q_d N)$. We will now define the total number of type j individuals coming from all ancestor types by the sum of the trees for all unique ancestor types,

$$Z_j(t;\boldsymbol{N}) = \sum_{i=1}^{d} Z_j^{(i)}(t;N_i) = \sum_{i=1}^{d} \sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t)$$

Since $Z_j(t;\boldsymbol{N})$ still represents a sum of independent processes, the Central Limit Theorem still holds as $N \to \infty$. The moments of $Z_j(t;\boldsymbol{N})$ can be written as sums of the moments from

the process initiated by a single type,

$$\mathbb{E}\left[\sum_{i=1}^{d}\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t)\right] = \sum_{i=1}^{d} N_i m_j^{(i)}(t)$$
$$\text{Var}\left[\sum_{i=1}^{d}\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t)\right] = \sum_{i=1}^{d} N_i \sigma_j^{2(i)}(t)$$
$$\text{Cov}\left[\sum_{i=1}^{d}\sum_{k=1}^{N_i} Z_{j(k)}^{(i)}(t), \sum_{i=1}^{d}\sum_{k=1}^{N_i} Z_{l(k)}^{(i)}(t)\right] = \sum_{i=1}^{d} N_i \text{Cov}\left[Z_j^{(i)}(t), Z_l^{(i)}(t)\right] = \sum_{i=1}^{d} N_i C_{jl}^{(i)}(t)$$

Define

$$V_j(t;\boldsymbol{N}) \equiv \frac{\sum_{i=1}^{d}\left[Z_j^{(i)}(t;N_i) - N_i m_j^{(i)}(t)\right]}{\sqrt{\sum_{i=1}^{d} N_i \sigma_j^{2(i)}(t)}} = \frac{Z_j(t;N) - \sum_{i=1}^{d} N_i m_j^{(i)}(t)}{\sqrt{\sum_{i=1}^{d} N_i \sigma_j^{2(i)}(t)}}$$

As $N \to \infty$, each of the $N_i = q_i N \to \infty$ at the same rate and by the Central Limit Theorem,

$$V_j(t;\boldsymbol{N}) \xrightarrow{d} Y_j(t)$$

where $Y_j(t) \sim \text{Normal}(0,1)$. With the Central Limit Theorem for i.i.d. vectors,

$$(V_1(t;N_i), V_2(t;N_i), \ldots, V_d(t;N_i)) \xrightarrow{d} (Y_1(t), Y_2(t), \ldots, Y_d(t)) \equiv \mathbf{Y}(t)$$

which is normally distributed, $\mathbb{E}[\mathbf{Y}(t)] = 0$,$\text{Var}[\mathbf{Y}(t)] = 1$, with

$$\text{Cov}\left(Y_j(t), Y_k(t)\right) = \sum_{i=1}^{d} N_i C_{jk}^{(i)}(t)$$

Thus, even for a process beginning with N ancestors from different types according to $\mathbf{q} = (q_1, q_2, \ldots, q_d)$, the Central Limit Theorem holds, and a process beginning with a large enough initial population can be approximated with a normal distribution.

Assume $\mathbf{z}_1(t; \boldsymbol{N}), \mathbf{z}_2(t; \boldsymbol{N}), \ldots, \mathbf{z}_m(t; \boldsymbol{N})$ are realizations of the branching process above parameterized by θ with large enough m. Then the log-likelihood function can be approximated by a normal log-likelihood,

$$l(\theta; \mathbf{Z}(t), \boldsymbol{N}, t) \propto -\frac{m}{2} \log |(t; \theta)| - \frac{1}{2} \sum_{i=1}^{m} \left[\mathbf{Z}_{(i)}(t) - \mathbf{M}(t; \theta)\boldsymbol{N}\right]^{\intercal} (t; \theta)^{-1} \left[\mathbf{Z}_{(i)}(t) - \mathbf{M}(t; \theta)\boldsymbol{N}\right]$$

with $\mathbf{M}(t; \theta)$ representing the d-element vector that is the mean of the branching process, and $(t; \theta)$ representing the covariance matrix of the process.

A maximum likelihood estimator for θ is thus defined:

$$\hat{\theta} \equiv \arg\max_{\theta} l(\theta; \mathbf{Z}(t), \boldsymbol{N}, t)$$

Calculating the moments for a continuous-time Markov branching process

We focus on a particular example - when the lifetime of cells is exponentially distributed. Denote a d-type Markov branching process initiated by a single type i ancestor by $\mathbf{Z}^{(i)}(t)$. A type i ancestor has offspring probability generating function $f_i(\mathbf{s})$ where $\mathbf{s} = (s_1, \ldots, s_d)$ with $|s_k| \leq 1$. Assume a type i individual lives for an exponentially distributed amount of time parameterized by a_i and divides into $\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^d$ offspring with probability $p_{\mathbf{j}}^{(i)}$. Thus,

$$f_i(\mathbf{s}) = \sum_{\mathbf{j}} p_{\mathbf{j}}^{(i)} s_1^{j_1} s_2^{j_2} \ldots s_d^{j_d}$$

The probability generating function for the continuous time process initiated with a single type i ancestor at time t is defined as

$$F_i(\mathbf{s}, t) = \mathbb{E}\left[\mathbf{s}^{\mathbf{Z}^{(i)}(t)}\right] = \sum_{j_1, \ldots j_d} P(\mathbf{Z}^{(i)}(t) = \mathbf{j}) s_1^{j_1} s_2^{j_2} \ldots s_d^{j_d}$$

where $\mathbf{s}^{\mathbf{Z}^{(i)}(t)} = \prod_{k=1}^{d} s_k^{Z_k^{(i)}(t)}$.

Define the generating function $u^{(i)}(\mathbf{s}) \equiv a_i [f_i(\mathbf{s}) - s_i]$ which is used to move between probabilities and lifetimes to rates with respect to the offspring distribution function. We solve

for the first two moments using the Kolmogorov Backward Equation, written

$$\begin{aligned}\frac{\partial}{\partial t}F_i(\mathbf{s};t) &= u^{(i)}[\mathbf{F}(\mathbf{s};t)] \\ &= a_i\left[f_i\left[F_1(\mathbf{s};t),\ldots,F_d(\mathbf{s};t)\right]-F_i(\mathbf{s};t)\right].\end{aligned}$$

We define the first moments of the process by the $d \times d$ matrix $\mathbf{M}(t)$, where $[\mathbf{M}(t)]_{i,j} = m_{ij}(t) = \mathbb{E}\left[Z_j^{(i)}(t)\right]$. The second moments are defined by the vectors $\mathbf{d}_{jk}(t)$ where $d_{jk}^{(i)}(t) = \mathbb{E}\left[Z_j^{(i)}(t)Z_k^{(i)}(t)\right]$.

Derivation of $\mathbf{M}(t)$

The mean matrix $\mathbf{M}(t)$ can be found by using the Kolmogorov Backwards Equation and the derivative of the probability generating function for the process. Note

$$\mathbb{E}\left[Z_j^{(i)}(t)\right] = \frac{\partial}{\partial s_j}F_i(\mathbf{s};t)|_{\mathbf{s}=1}$$

Using the Kolmogorov Backward Equation,

$$
\begin{aligned}
\frac{\partial}{\partial t}\mathbb{E}\left[Z_j^{(i)}(t)\right] &= \frac{\partial}{\partial t}\frac{\partial}{\partial s_j}F_i\left(\mathbf{s};t\right)|_{\mathbf{s}=1} \\
&= \frac{\partial}{\partial s_j}\frac{\partial}{\partial t}F_i\left(\mathbf{s};t\right)|_{\mathbf{s}=1} \\
&= \frac{\partial}{\partial s_j}a_i\left[f_i\left[F_1(\mathbf{s};t),\ldots,F_d(\mathbf{s};t)\right]-F_i(\mathbf{s};t)\right]|_{\mathbf{s}=1} \\
&= \frac{\partial}{\partial s_j}a_i\left[\sum_{\mathbf{k}}p_{i\mathbf{k}}(t)F_1^{k_1}\ldots F_d^{k_d}-F_i\right]\Bigg|_{\mathbf{s}=1} \\
&= a_i\left[\sum_{\mathbf{k}}p_{i\mathbf{k}}(t)\left(k_1\partial_j F_1 F_1^{k_1-1}\ldots F_d^{k_d}+\cdots+k_d\partial_j F_d F_1^{k_1}\ldots F_d^{k_d-1}\right)-\partial_j F_i\right]\Bigg|_{\mathbf{s}=1} \\
&= a_i\left[\sum_{\mathbf{k}}p_{i\mathbf{k}}(t)\left(k_1\partial_j F_1+\cdots+k_d\partial_j F_d\right)-\partial_j F_i\right] \\
&= a_i\left[\sum_{\mathbf{k}}p_{i\mathbf{k}}(t)\left(k_1 m_{1j}(t)+\cdots+k_d m_{dj}(t)\right)-m_{ij}(t)\right] \\
&= a_i\left[b_{i1}m_{1j}(t)+\cdots+b_{id}m_{dj}(t)\right)]
\end{aligned}
$$

where $b_{ij} = \frac{\partial}{\partial s_j}f_i(\mathbf{s}) - \delta_{ij}$, where $\delta_{ij} = 1$ when $i = j$ and 0 otherwise. For ease of understanding, we have defined $\partial_j F \equiv \frac{\partial}{\partial s_j}F$ and have excluded the parameters for $F(\mathbf{s},t)$ when not necessary. Doing this for all i,j results in the matrix ODE,

$$\frac{d}{dt}\mathbf{M}(t) = \mathbf{A}\mathbf{M}(t)$$

where

$$[\mathbf{A}]_{ij} = a_i b_{ij}.$$

The solution to the ODE yields

$$\mathbf{M}(t) = \exp\{\mathbf{A}t\}.$$

Derivation of the second moments, $\mathbf{d}_{jk}(t)$

The following results are a corrected version of results presented by Athreya and Ney on page 203 of their book "Branching processes"[11]. Given a type i ancestor, the ODE for the second moments with respect to type j and k can be solved by using the Kolmogorov Backward Equation. The previous derivation leads to a matrix ODE that is solved for an individual ancestor type, while we recognize that the solution is a system of ODE's for the second moment of types j and k that are solved for all ancestors. The resulting equations are

$$\frac{d}{dt} d_{jk}^{(i)}(t) = a_i \left[\sum_{l=1}^{d} \left(b_{il} d_{jk}^{(l)}(t) + \sum_{n=1}^{d} c_{ln}^{(i)} m_{lk}(t) m_{nj}(t) \right) \right]$$

where

$$b_{ij} = \left. \frac{\partial}{\partial s_j} f_i(\mathrm{s}) \right|_{\mathrm{s}=1} - \delta_{ij} \text{ and } c_{jk}^{(i)} = \left. \frac{\partial^2}{\partial s_j \partial s_k} f_i(\mathrm{s}) \right|_{\mathrm{s}=1}$$

For the vector $\mathbf{d}_{jk} = \left(d_{jk}^{(1)}(t), \ldots, d_{jk}^{(d)}(t)\right)^{\mathsf{T}}$ indexed by its ancestor type, we simplify the ODE to

$$\frac{d}{dt}\mathbf{d}_{jk}(t) = \mathbf{A}\mathbf{d}_{jk}(t) + \beta_{jk}(t), \quad d_{jk}^{(i)}(0) = 1 \text{ if } i = j = k$$

where $\beta_{jk}(t) = \left(\beta_{jk}^{(1)}(t), \ldots, \beta_{jk}^{(d)}(t)\right)^{\mathsf{T}}$ and

$$\beta_{jk}^{(i)}(t) = a_i \sum_{l=1}^{d} \sum_{n=1}^{d} c_{ln}^{(i)} m_{lk}(t) m_{nj}(t).$$

This ODE has a general solution

$$\mathbf{d}_{jk}(t) = \exp\{\mathbf{A}t\}\mathbf{d}_{jk}(0) + \int_0^t \exp\{\mathbf{A}(t-s)\}\beta_{jk}(s)ds$$

or, more simply,

$$\mathbf{d}_{jk}(t) = \mathbf{M}(t)\mathbf{d}_{jk}(0) + \int_0^t \mathbf{M}(t-s)\beta_{jk}(s)ds.$$

These vectors of second moments allow us to define the variance matrix for the process,

$$\left[{}^{(i)}(t)\right]_{jk} = d_{jk}^{(i)}(t) - m_{ij}(t)m_{ik}(t).$$

When dealing with rates such as the birth and death rates in place of probabilities and life-

time rates, we can rewrite expression above to account for this. In this case, define $\gamma_{i\mathbf{j}} = a_i p_{i\mathbf{j}}$ so that $a_i = \sum_{\mathbf{j}} \gamma_{i\mathbf{j}}$. Then we can rewrite the generator $u^{(i)}(\mathbf{s})$ as

$$u^{(i)}(\mathbf{s}) = \sum_{j} \gamma_{i\mathbf{j}} \boldsymbol{s}^{\boldsymbol{j}} - s_i \sum_{\boldsymbol{j}} \gamma_{i\mathbf{j}}.$$

The following expressions for $\mathbf{A}$, $c_{jk}^{(i)}$, and $\beta_{jk}^{(i)}$ can be written to account for this change as

$$\begin{aligned}
[\boldsymbol{A}]_{i,j} &= \frac{\partial}{\partial s_j} u^{(i)}(\mathbf{s})|_{\boldsymbol{s}=\mathbf{1}} \\
c_{jk}^{(i)} &= \frac{\partial^2}{\partial s_j \partial s_k} u^{(i)}(\boldsymbol{s})|_{bms=\mathbf{1}} \\
\beta_{jk}^{(i)}(t) &= \sum_{l=1}^{d} \sum_{n=1}^{d} c_{ln}^{(i)} m_{lk}(t) m_{nj}(t).
\end{aligned}$$

Using these results to find the first two moments of a process, we can write the likelihood above by plugging in the expressions for the mean and variance and estimate the parameters of a process with the maximum likelihood estimator for the parameters $\gamma_{ij}, i = 1, \ldots, d$. The resulting maximum likelihood estimators are obtained via standard optimization procedures in ESTIpop.

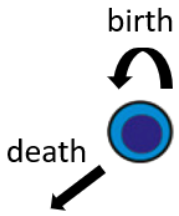

Figure C.1: One-type birth death model in which members of the population live for an exponentially-distributed time with parameter $1/(birth + death)$. At the end of a lifetime, an individual will give birth to two new individuals with probability $birth/(birth + death)$ for a net population increase of 1 and will die and be removed from the population with probability $death/(birth + death)$ for a net population decrease of 1.

C.2 Vignette 1: One-Type Birth-Death Process

C.2.1 Background

To begin our example, let us consider the one-type birth-death process shown in Figure C.1. In this process, there is only one population type. Individuals from this population undergo birth events at a rate of *birth* events per individual per unit time and death events at a rate of *death* events per individual per unit time.

C.2.2 Simulation using ESTIpop

We begin our study of the one-type process by first generating ground truth data via simulation, using the methods that are available in the ESTIpop package. In the following sections, we make repeated calls to the following base code to simulate the data under various conditions.

```
library(estipop)

# Specify how many units of time to simulate
time = 1

# Simulation will initiate with a single type with size 500
initial = c(500)

# Specify two fixed transitions, birth and death
transitionList = TransitionList(FixedTransition(population = 0,
                 rate = 1.0,
                 fixed = c(2)),
         FixedTransition(population = 0,
                 rate = 0.7,
                 fixed = c(0)))

# No other stops beyond time
stopList = StopList()

# Simulation 1000 trials
ntrials = 1000

full_res = matrix(ncol = 2)

# Run simulations and store results into res
for(i in 1:ntrials){
  res = branch(time, initial, transitionList, stopList)
  full_res = rbind(full_res, as.matrix(res))
}

full_res = na.omit(full_res)
data = as.matrix(full_res[full_res[,1] == 1,2])
```

Above, we specify that we simulate the model for one unit of time with an initial population size of 500. Our birth rate is set at 1.0 and death rate at 0.7 for a net growth rate (birth - death) of 0.3. Using a simple loop, we simulate 1,000 such data points.

Estimation using ESTIpop

We estimate the birth and death rates using ESTIpop and the following script.

```
# Set up our estimation parameters
N = c(500)

time = 1

transitionList = TransitionList(FixedTransition(population = 0,
                                        fixed = c(2)),
                                FixedTransition(population = 0,
                                        fixed = c(0)))

initial = c(1, 0.5)

# Estimate using the estimateBP function
estimates = estimateBP(time = time,
        N = N,
        transitionList = transitionList,
        data = data,
        initial = initial)
```

Above, we must first specify all of our estimation parameters: an initial population size (N) of 500, time of observations 1 unit, our model, using a TransitionList object, as well as an initial optimization point.

Estimation Evaluation in Various Scenarios

In the following subsections, we introduce fluctuations in various simulation parameters in order to evaluate our estimation method. In each condition, we simulate 1,000 data sets of vari-

ous sizes ranging from 10 to 500. For each data set, we estimate the birth and death parameters using ESTIpop. We summarize the set of 1,000 estimates by plotting the mean estimate for each rate along with the 25th and 75th percentiles of the estimates.

Magnitude of Birth and Death Rates (Constant Net Growth)

In this scenario, we are interested in the effect of changing the birth and death rates such that their difference, the net growth rate, remains constant. In other methods, such as using linear regression with a log transformation, we are only able to characterize the net growth rate, meaning that the differences in these scenarios would be unidentifiable. ESTIpop is able to correctly detangle birth and death rates from the net growth rate by taking into account the second moments of the underlying branching process. In Figure C.2, we plot the mean estimates (black line) and 25th and 75th percentiles against sample size for the various conditions.

As expected from the theory, greater birth and death rates result in larger variances in our estimates for the same sample size.

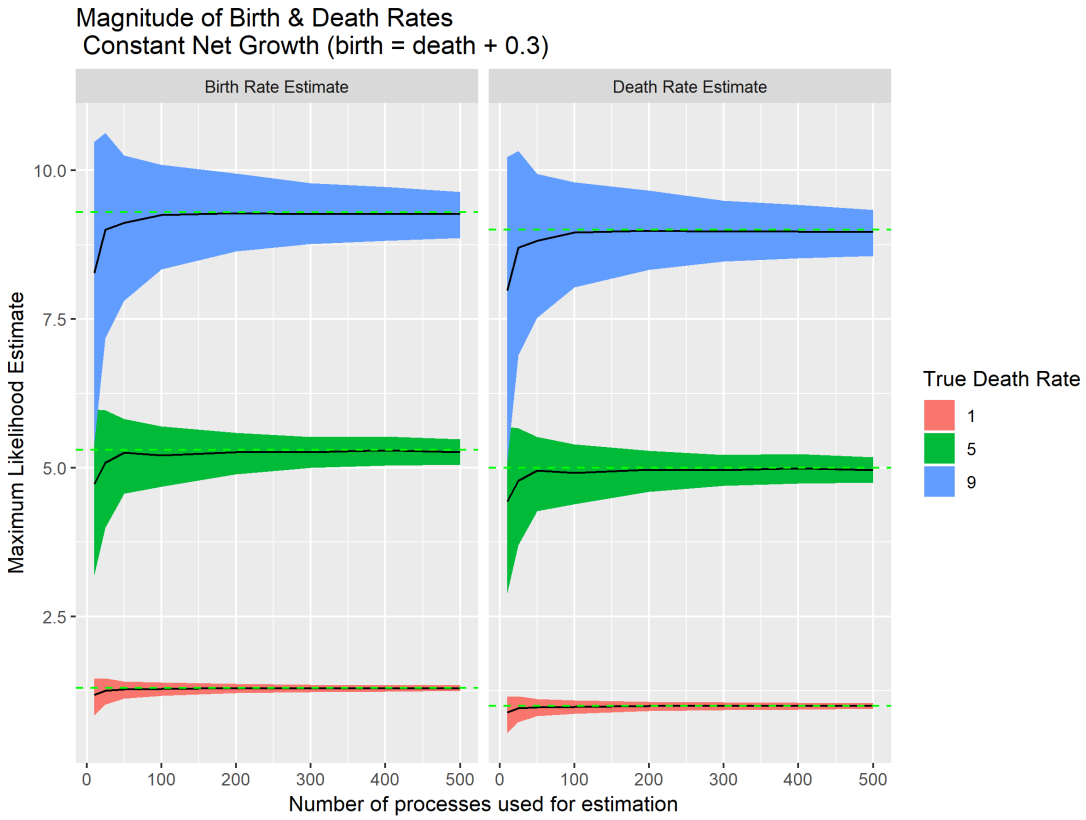

Figure C.2: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the birth (left) and death (right) rates against sample size for death rates 1.0, 5.0, and 9.0 (birth = death + 0.3).

Magnitude of Net Growth Rate

In this scenario, we investigate the effect of changing the net growth rate by varying the death rate while keeping the birth rate constant. In other methods, we are only able to characterize the net growth rate, meaning that the differences in these scenarios would be unidentifiable. ESTIpop is able to correctly detangle birth and death rates from the net growth rate by taking into account the second moments of the underlying branching process. In Figure C.3, we plot the mean estimates (black line) and 25th and 75th percentiles against sample size for the various conditions.

We observe that larger net growth rates resulted in larger variances in our estimates of both the birth and death rates.

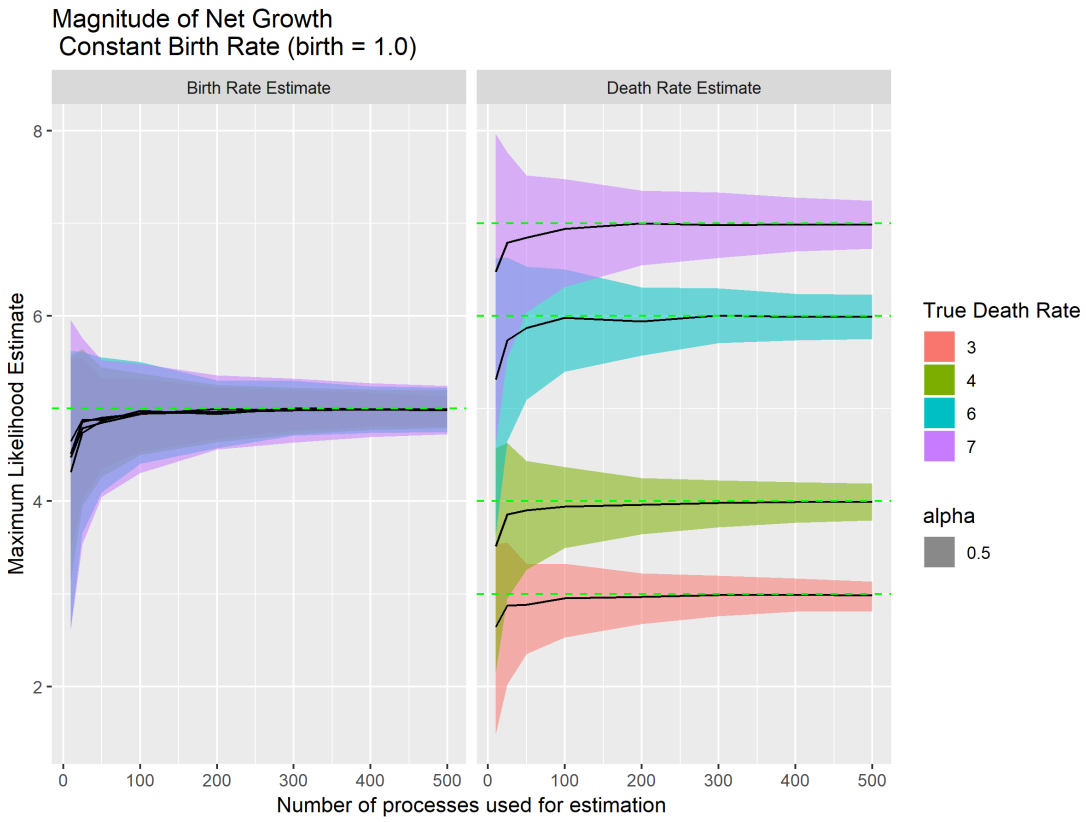

Figure C.3: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the birth (left) and death (right) rates against sample size for death rates 3.0, 4.0, 6.0, and 7.0, while holding the birth rate constant at 5.0.

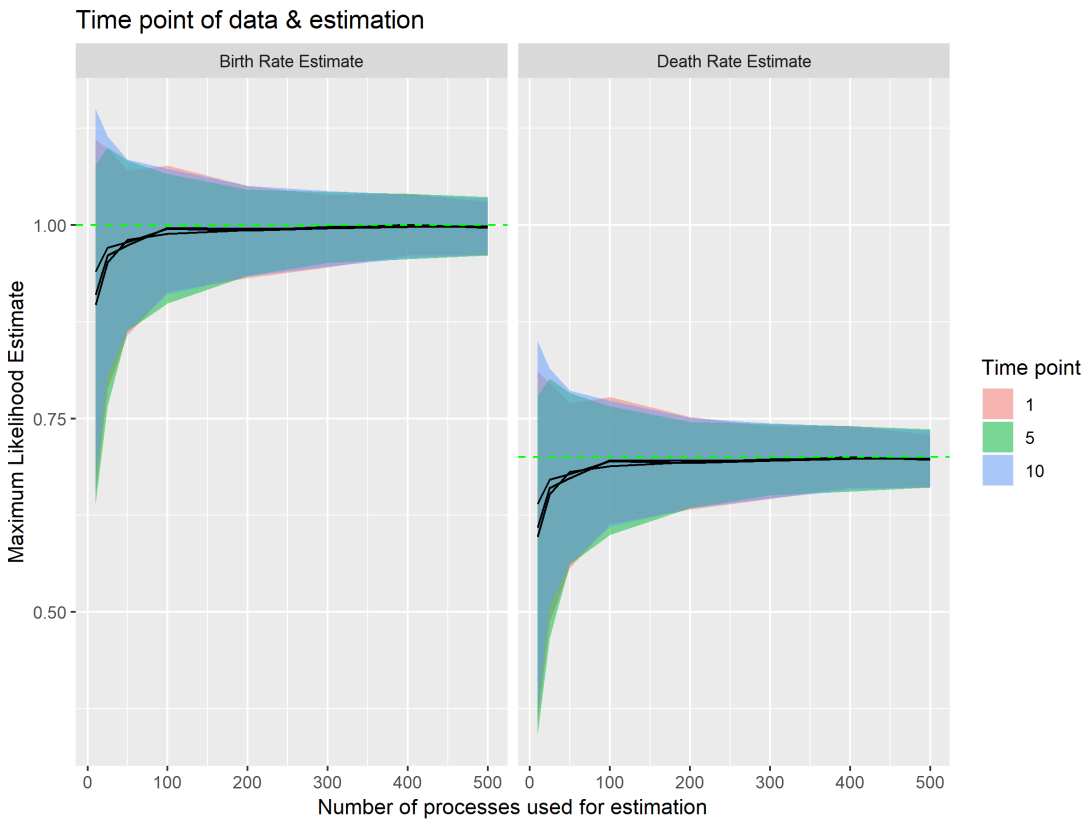

Figure C.4: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the birth (left) and death (right) rates against sample size for using time points 1, 5, or 10, holding the birth and death rates constant at 1.0 and 0.7 respectively.

Time Points

In this scenario, we investigate the effect of using different time points to estimate the birth and death rates. In these simulations, all observations for each estimation come from the same selected time point. In Figure C.4, we plot the mean estimates (black line) and 25th and 75th percentiles against sample size for the various conditions.

We observe that using observations from timepoints 1, 5, or 10 made little to no difference

in the variance of the estimates.

Initial Population Size

In this scenario, we investigate the effect of changing the initial population size. In biological experiments, this could possibly coincide with the number of seeding cells in cell viability assay or the number of initial cells in a differentiation hierarchy. In Figure C.5, we plot the mean estimates (black line) and 25th and 75th percentiles against sample size for the various conditions.

We observe that the initial size of the population made little difference in the variance of the estimates, suggesting that possibly the magnitude of the birth and death rates and the sample size has greater influence over the variance of our estimator than the initial population size.

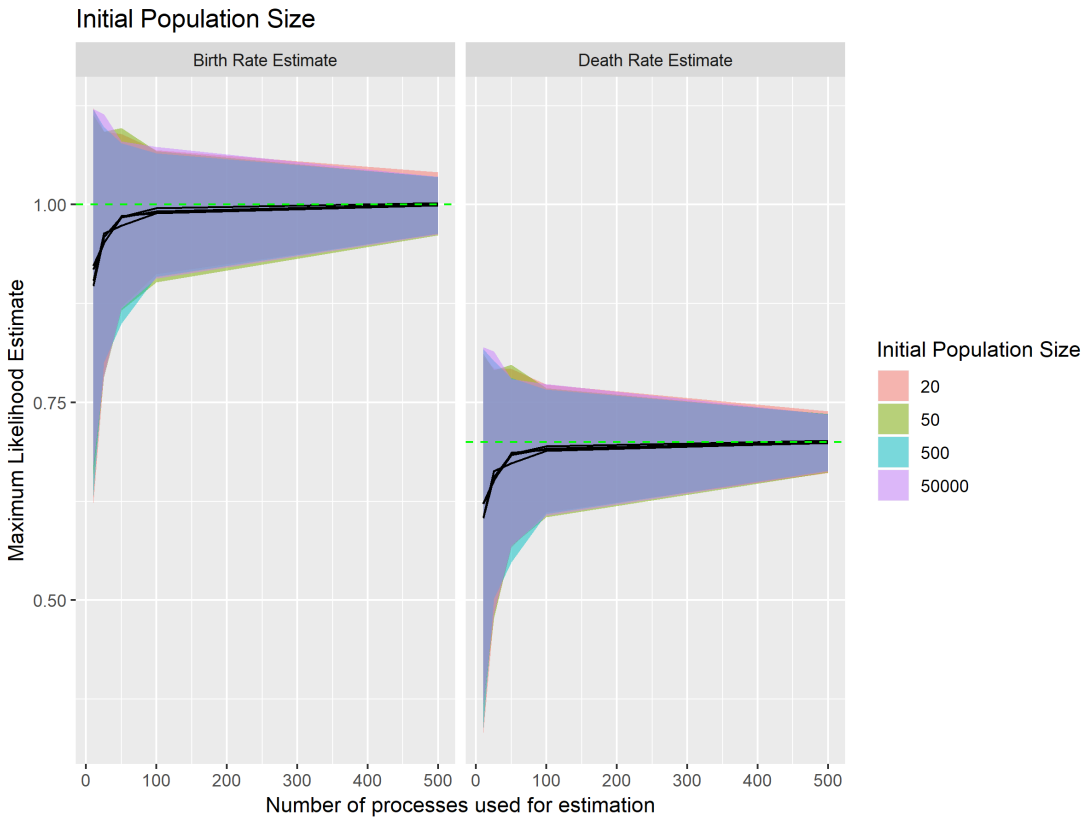

Figure C.5: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the birth (left) and death (right) rates against sample size for using initial population sizes of 20, 50, 500, and 50,000,, holding the birth and death rates constant at 1.0 and 0.7 respectively.

Measurement Error

In the following scenarios, we investigate the effect of possible measurement error on our estimation procedures. Measurement error in experimental procedures can occur due to a wide variety of reasons, including machine precision limits, technical variation, operator error, or systematic biases. We investigate both random and systematic measurement error.

Random Measurement Error In the random measurement error scenario, for each original data point denoted *value*, we draw from a uniform distribution with minimum $value - value \times error$ and maximum $value + value \times error$, meaning that on average, the error in the system is 0, and measurement error could both skew the data positively or negatively.

We observe that with increasing error rate, the estimates of the birth and death rate increased in magnitude (Figure C.6). This is explained by noting that an increase in error rate has the effect of increasing the variance of our experimental data, which results in a higher estimation of the birth and death rate even if the estimate of net growth ($birth - death$) is maintained. We also note that higher error rates increased the variance of our estimates.

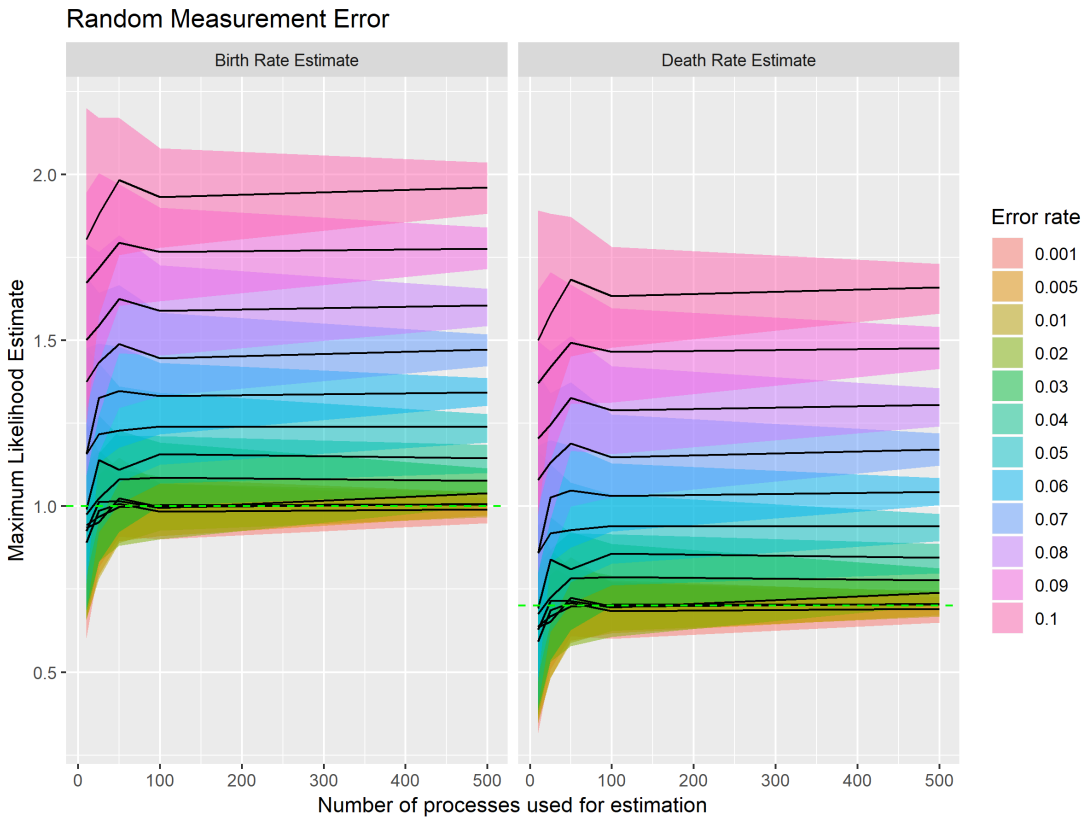

Figure C.6: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the birth (left) and death (right) rates against sample size for using various error rates, holding the birth and death rates constant at 1.0 and 0.7 respectively.

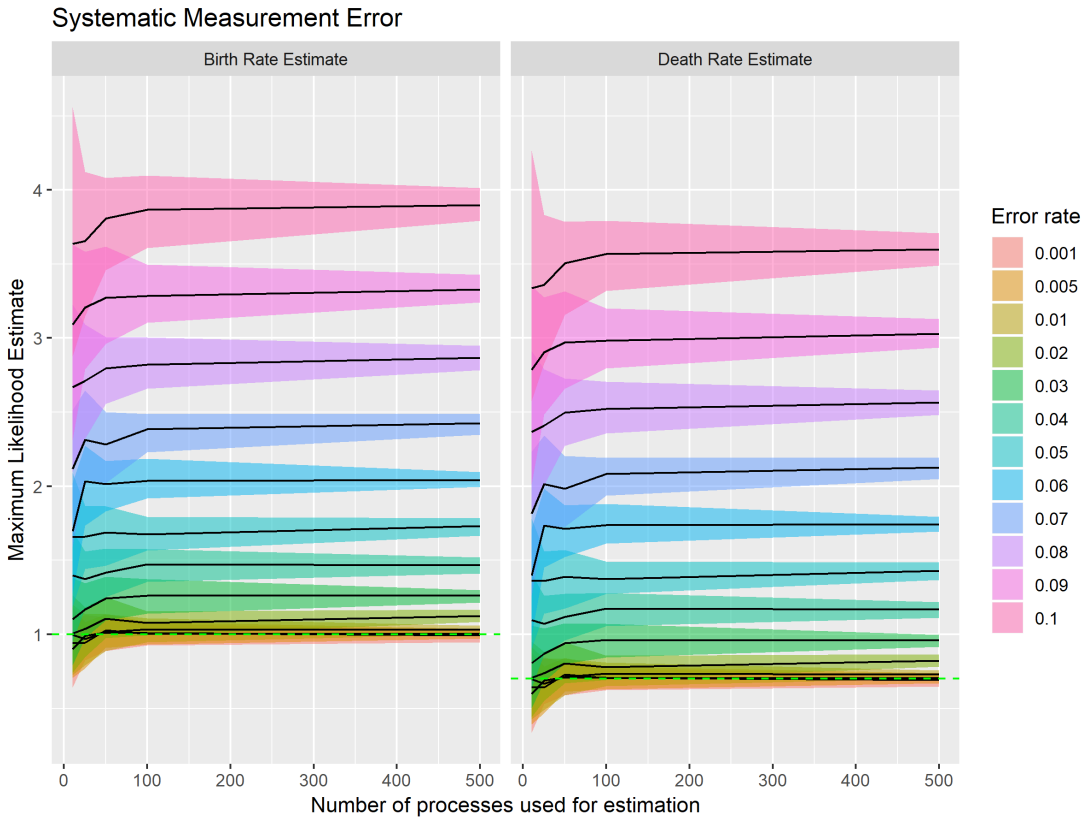

Figure C.7: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the birth (left) and death (right) rates against sample size for using various error rates, holding the birth and death rates constant at 1.0 and 0.7 respectively.

SYSTEMATIC MEASUREMENT ERROR In the systematic measurement error scenario, for each original data point denoted *value*, we flip a coin and assigned a new value, either $value - value \times error$ or $value + value \times error$.

Similar to the scenario in random measurement error, in the systematic measurement error scenario we observe that with increasing error rate, the estimates of the birth and death rate increased in magnitude (Figure C.7). This is explained by noting that an increase in error rate has the effect of increasing the variance of our experimental data, which results in a

higher estimation of the birth and death rate even if the estimate of net grown ($birth - death$) is maintained. This effect appeared even larger in systematic error, as the variance induced by systematic error is larger than the variance induced by random error, where by random chance our induced measurement error could perturb the actual values much less. We also note that higher error rates increased the variance of our estimates.

Comparison to Other Approaches

Log-transform Linear Regression A common approach to estimate the growth parameter for an exponentially-expanding population is to use a log-transform on the population counts and then use linear regression to estimate the growth parameter. We use the following code to estimate:

```
# Set up our time column
time_mod = c(rep(time, nrow(data)), rep(0, nrow(data)))

# Set up our log-transformed data
data_mod = c(data, rep(N, nrow(data)))

# Fit a regression to the transformed data
log.mod = lm(log(data_mod) ~ time_mod)
```

Non-linear Least Squares Another approach to estimate the growth paramter is to use non-linear least squares using the *nls* package in R. We use the following code for this fit and functional form $y = e^{a+bt}$, where y is our observed data, e^a is our initial population size, b represents our net growth rate, and t is the observation time:

```
# Fit non-linear least squares to data
nls.mod <- nls(data_mod ~ I(exp(1)^(a + b * time_mod)),
        start = list(a = 8, b = 1))
```

ESTIpop - Net Growth Model To compare against these methods, we adapt our estimation code to only estimate a birth rate, thereby estimating the net growth parameter under the assumption that births exceeded deaths, which because we simulate the ground truth data, we knew to be true. We use the following code:

```
# Set up our estimation parameters
N = c(500)

time = 1

transitionList = TransitionList(FixedTransition(population = 0,
                                                fixed = c(2)))

initial = c(0.5)

# Estimate using the estimateBP function
        estimates = estimateBP(time = time,
        N = N,
        transitionList = transitionList,
        data = data,
        initial = initial)
```

Further, to quantify our certainty in our estimate, we use the standard errors from both the log-transformed linear model and the non-linear fit to calculate 95% confidence intervals around our point estimates. For our ESTIpop results, we calculate a 95% confidence interval by bootstrapping from the simulated data set using the following code:

```
# Store our results to a vector of bootstraped estimates
bootstraps = c()
```

```
# For each bootstrapped sample
for(j in 1:100){
        # Sample from the data with replacement
        bs_data = as.matrix(sample(data, size = nrow(data),
                                   replace = T))

        # Estimate using this data set
        bs_estimates = estimateBP(time = time,
                N = N,
                transitionList = transitionList,
                data = bs_data,
                initial = initial)
        bootstraps = c(bootstraps, bs_estimates$par[1])
}
```

We summarize the results in the following tables:

Table C.1: Mean Point Estimate and Bootstrapped Confidence Interval for ESTIpop

Sample Size	Mean Point Estimate	Mean Lower Bound	Mean Upper Bound
12	0.7001426	0.6781868	0.7216032
50	0.6999343	0.6886587	0.7113181
100	0.6997904	0.6917874	0.7078073

Table C.2: Mean Point Estimate and Confidence Interval for log-transformed linear fit

Sample Size	Mean Point Estimate	Mean Lower Bound	Mean Upper Bound
12	0.6994784	0.6742910	0.7246659
50	0.6990848	0.6869721	0.7111974

Sample Size	Mean Point Estimate	Mean Lower Bound	Mean Upper Bound
100	0.6989030	0.6904195	0.7073865

Table C.3: Mean Point Estimate and Confidence Interval for non-linear fit

Sample Size	Mean Point Estimate	Mean Lower Bound	Mean Upper Bound
12	0.7003260	0.6607738	0.7408940
50	0.7000069	0.6808690	0.7193716
100	0.6998230	0.6863995	0.7133570

In general we see little difference between the three methods with perhaps only a slighly worse performance for the non-linear fit. The major difference between the three methods is that using ESTIpop, it is possible to distinguish birth rates from death rates instead of reducing the change down to net growth (Table 4).

Table C.4: Mean Point Estimate for Birth & Death Rates

Sample Size	Mean Birth Rate Estimate	Mean Death Rate Estimate
12	0.9289181	0.2287755
50	0.9837548	0.2838205
100	0.9886928	0.2889024

Exact and Approximate Simulation

In the previous sections, we simulate in ESTIpop using exact simulation via Gillespie's Stochastic Simulation Algorithm. Alternatively, ESTIpop also provides the option to use an approximate simulation based on the asymptotic distribution derived in the supplement. We now investigate the differences between the two. In the following script, we generate 10 samples of a simple birth-death process 100 times using both approximate and exact simulation.

```
# Set up parallelization
cores=detectCores()
print(paste("I'm working with", cores-1, "cores"))
cl <- makeCluster(cores[1]-1) #not to overload your computer
registerDoParallel(cl)

# Set up our model
tL = TransitionList(FixedTransition(0, 1.0, c(2)),
                    FixedTransition(0, 0.7, c(0)))

# Perform timed simulations
exact = foreach(i_=1:100, .combine = "rbind") %dopar%{
  t1 = system.time(replicate(10,
        simBirthDeath(birth = 1.0, death = 0.7, init = 500,
                time = 30, approx = F)))
  t1
}

approx = foreach(i_=1:100, .combine = "rbind") %dopar%{
 t2 = system.time(replicate(10,
        simBirthDeath(birth = 1.0, death = 0.7, init = 500,
                time = 30, approx = T)))
 t2
}
```

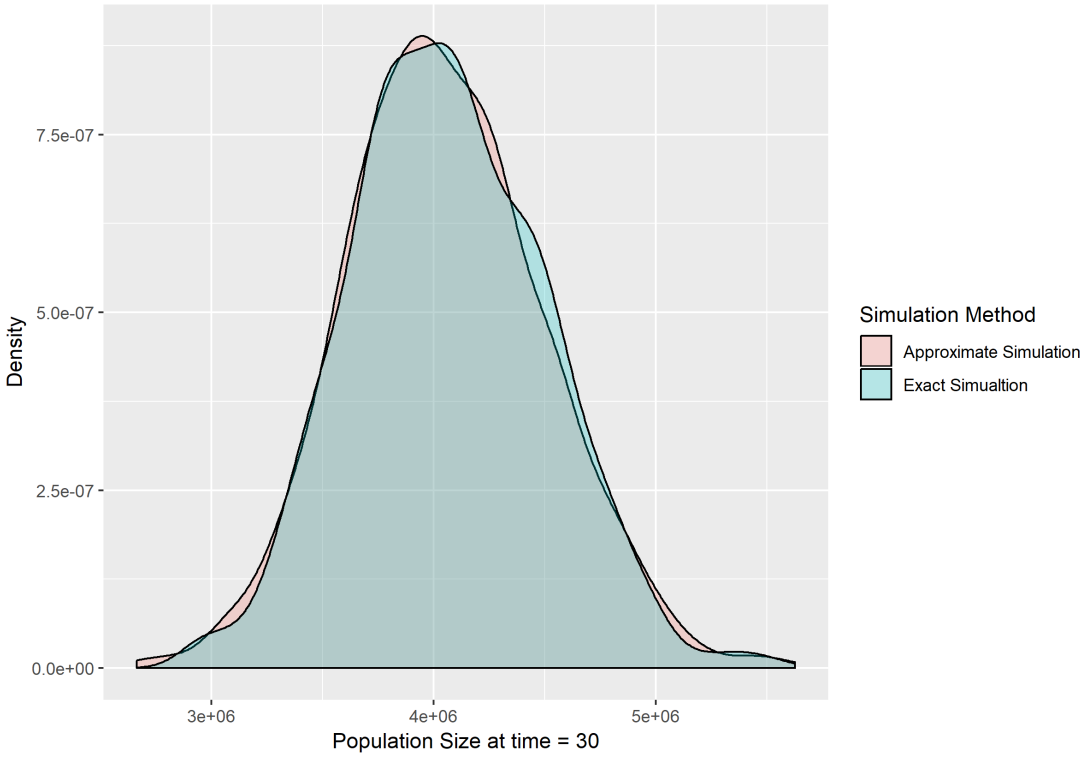

Figure C.8: Density plots of the resulting population size at time 30 for an initial population of 500 individuals with $birth = 1.0$ and $death = 0.7$ using both exact and approximate simulation.

In Figure C.8, we observe that the population distribution at time 30 is nearly identical for both exact and approximate simulation. In Table 5, we see that there is a significant time reduction in execution for the approximate simulation compared to the exact simulation. This time reduction can easily been seen in that approximate simulation requires only one draw from a random distriution, whereas, exact simulation requires multiple random draws for just a single transition, and the total simulation time could be made up of millions or billions of such transitions.

Table C.5: Average execution times, exact vs. approximate simulation

Method	User Time	System Time	Elapsed Time
Exact	182.17273	0.10448	315.24076
Approximate	0.45405	0.01790	0.84303

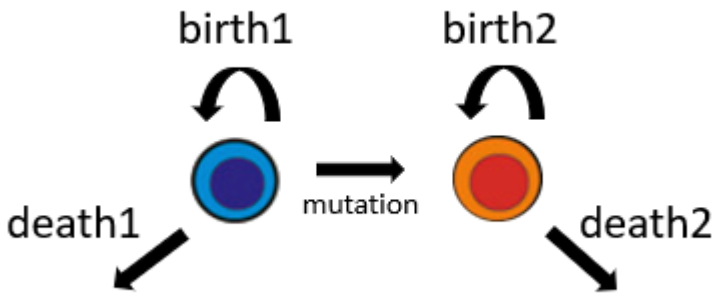

Figure C.9: Two-type birth-death-mutation model. Each type has its own birth and death rate parameters in addition to a mutation event from type 1 to type 2.

C.3 Vignette 2: Two-Type Birth-Death-Mutation Process

Background

To begin our example, let us consider the one-type birth-death process shown in Figure C.9. These models are used extensively in investigating the dynamics of tumor cells in response to treatment, where a population initially sensitive to therapy will gain resistance via mutation and expand at a different rate.

Simulation using ESTIpop

We begin our study of the two-type process by first generating ground truth data via simulation, using the methods that are available in the ESTIpop package. Here, we use the *simBirthDeathMutation* function available in ESTIpop.

```
library(estipop)

# Specify how many units of time to simulate
time = 1

# Simulation 1000 trials
ntrials = 1000

full_res = matrix(ncol = 3)

# Run simulations and store results into res
for(i in 1:ntrials){
  res = simBirthDeathMutation(birth1 = 1.0,
                              death1 = 0.65,
                              mutation = 0.1,
                              birth2 = 1.1,
                              death2 = 0.7,
                              init = c(100, 100),
                              time = 1)
  full_res = rbind(full_res, as.matrix(res))
}

full_res = na.omit(full_res)
data = as.matrix(full_res[full_res[,1] == 1,c(2,3)])
```

Above, we specify that we will simulate the model for one unit of time with initial population sizes of 100 for each type. Our birth rate and death rate for type 1 is set at 1.0 and 0.65 respectively. Our mutation rate is set at 0.1. Our birth and death rate for type 2 is set at 1.1, and 0.7 respectively, implying that mutation bestows a fitness increase. Using a simple loop, we simulate 1,000 such data points.

Estimation using ESTIpop

We estimate the birth and death rates using ESTIpop and the following script:

```
# Set up our estimation parameters
N = c(100, 100)

time = 1

transitionList = TransitionList(
        FixedTransition(population = 0, fixed = c(2, 0)),
        FixedTransition(population = 0, fixed = c(0, 0)),
        FixedTransition(population = 0, fixed = c(1, 1)),
        FixedTransition(population = 1, fixed = c(0, 2)),
        FixedTransition(population = 1, fixed = c(0, 0)))

initial = c(1, 0.65, 0.2, 1.2, 0.8)

# Estimate using the estimateBP function
estimates = estimateBP(time = time,
        N = N,
        transitionList = transitionList,
        data = data,
        initial = initial,
        known = c(FALSE, FALSE, FALSE, FALSE, FALSE))
```

Above, we must first specify all of our estimation paramters: initial population sizes (N) of 100 and 100 for both types, time point of observations at 1 unit, our model, using a TransitionList object, as well as an initial optimization point. In this example, we explore the effect of trying to estimate various rate parameters while holding the others as known using another parameter of the estimateBP function.

As an example, we hold the death rate for type 1 and the birth rate for type 2 fixed at their true values during estimation.

```r
# Initial estimate with TRUE death2 and birth1
initial = c(1, 0.65, 0.2, 1.1, 0.8)

# Estimate using the estimateBP function with known parameters
estimates = estimateBP(time = time,
        N = N,
        transitionList = transitionList,
        data = data,
        initial = initial,
        known = c(FALSE, TRUE, FALSE, TRUE, FALSE))

print(estimates)
```

```
## $par
## [1] 0.9938033850 0.0000000002 0.6149358515
##
## $value
## [1] 6758.128
##
## $counts
## function gradient
##       17       17
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
##
## $fullpars
## [1] 0.9938033850 0.6500000000 0.0000000002 1.1000000000 0.6149358515
##
## $fixed
## [1] FALSE  TRUE FALSE  TRUE FALSE
```

From the output above, we see that both the death rate for type 2 and the birth rate for type 1 were held at their initialized values. The following plots show the mean MLE and 95% inner

quantile range across 100 replicates for various sample sizes. In each plot, the title specifies the pattern of which rates were held constant with a 1 representing that a particular rate was held fixed at its true value and a 0 representing that a particular rate was included in estimation. For example, in Figure C.10, "00000" indicates that all rates were estimated.

We now investigate the effect of holding certain rates constant on estimating the type 1 birth rate, the type 1 death rate, and the mutation rate.

Estimating the Type 1 Birth Rate & Type 1 Death Rate

We begin by estimating the type 1 birth rate under the easiest scenario: when all other rate parameters are fixed at their true values. The results from those simulation runs are shown in Figure C.11. Comparing Figure C.10 to Figure C.11, we observe that there is much lower variance in our estimate when estimating just a single rate compared to trying to estimate all five rate parameters simultaneously.

Similarly, when trying to estimate the type 1 death rate in the easiest fashion, when all other rates are held fixed at their true values, we observe a dramatic decrease in the variance of our estimate compared to estimating all rates simultaneously (Figures C.10 & C.12).

In Figure C.13 the results from estimating both the type 1 birth and type 1 death rates are shown. In this plot, we observe that the variance in both estimates increases back to the same magnitude as trying to estimate all rates simultaneously.

This pattern is seen in all other simulation patterns as well. Whenever the type l birth and death rates are estimated simultaneously, the variance for each is much larger; however, when

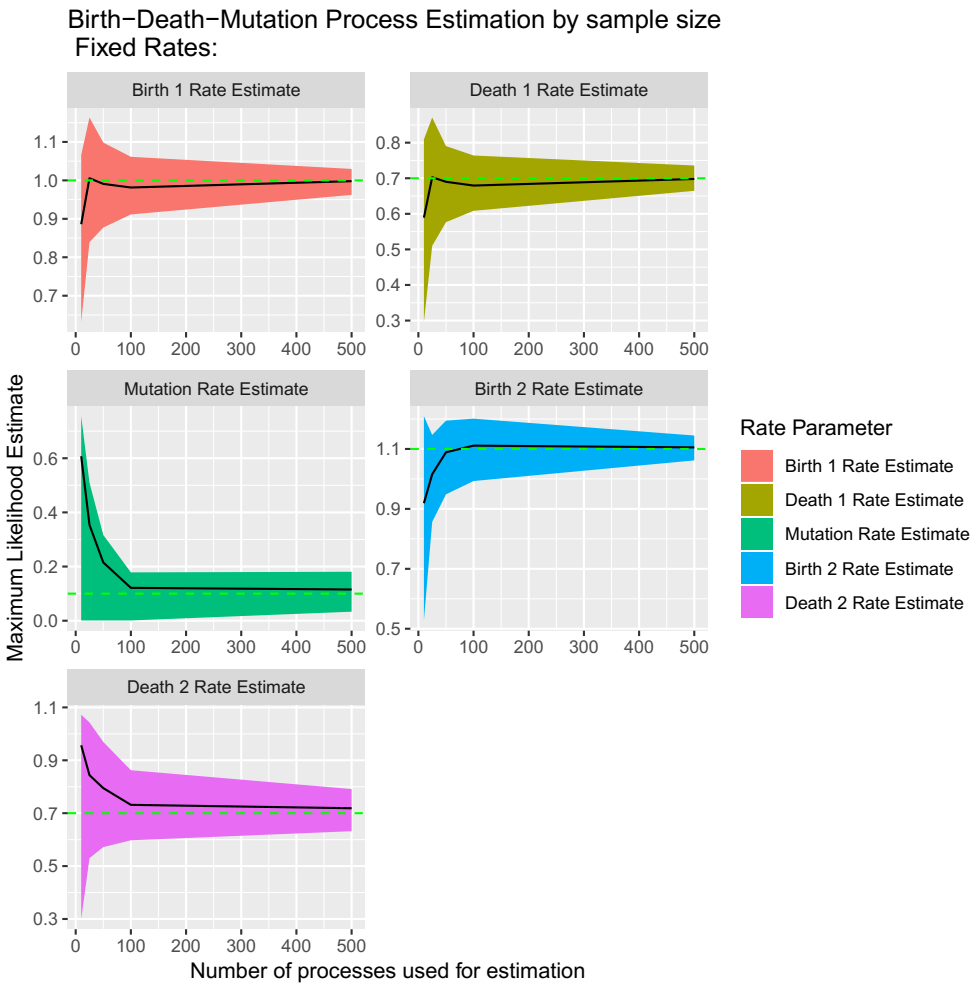

Figure C.10: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

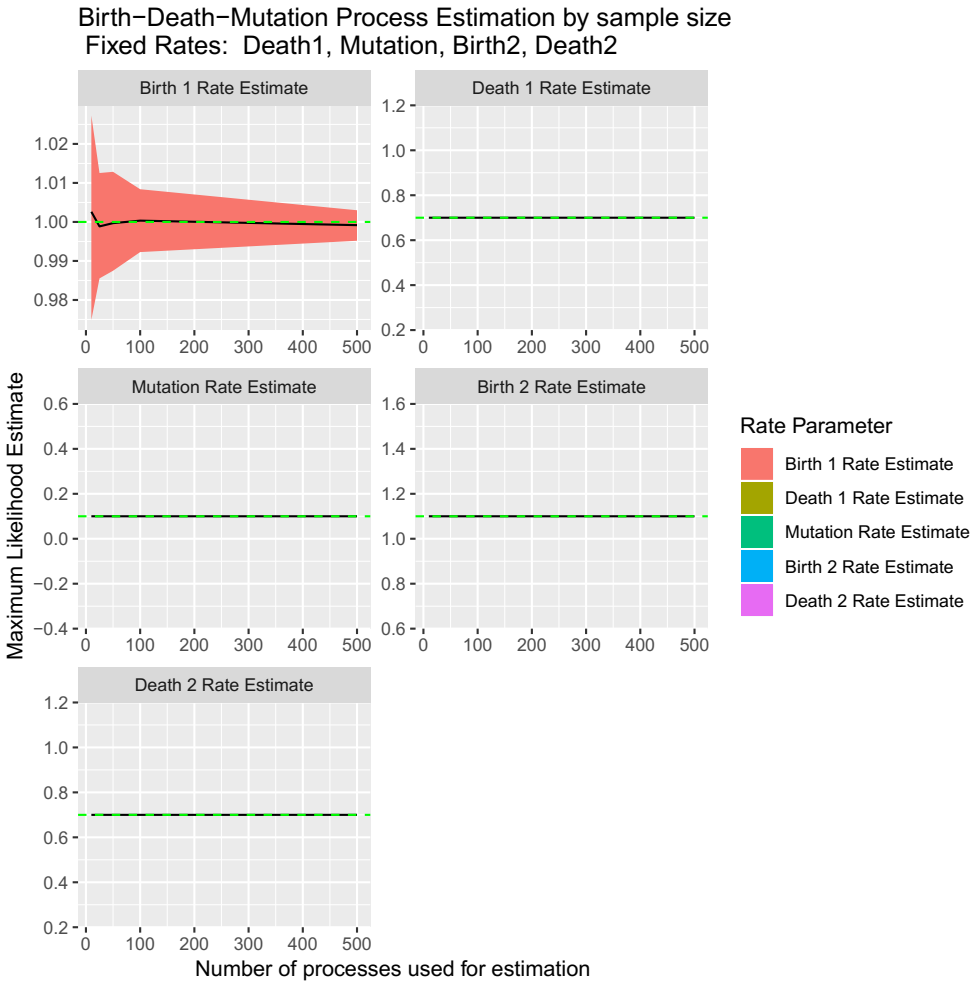

Figure C.11: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

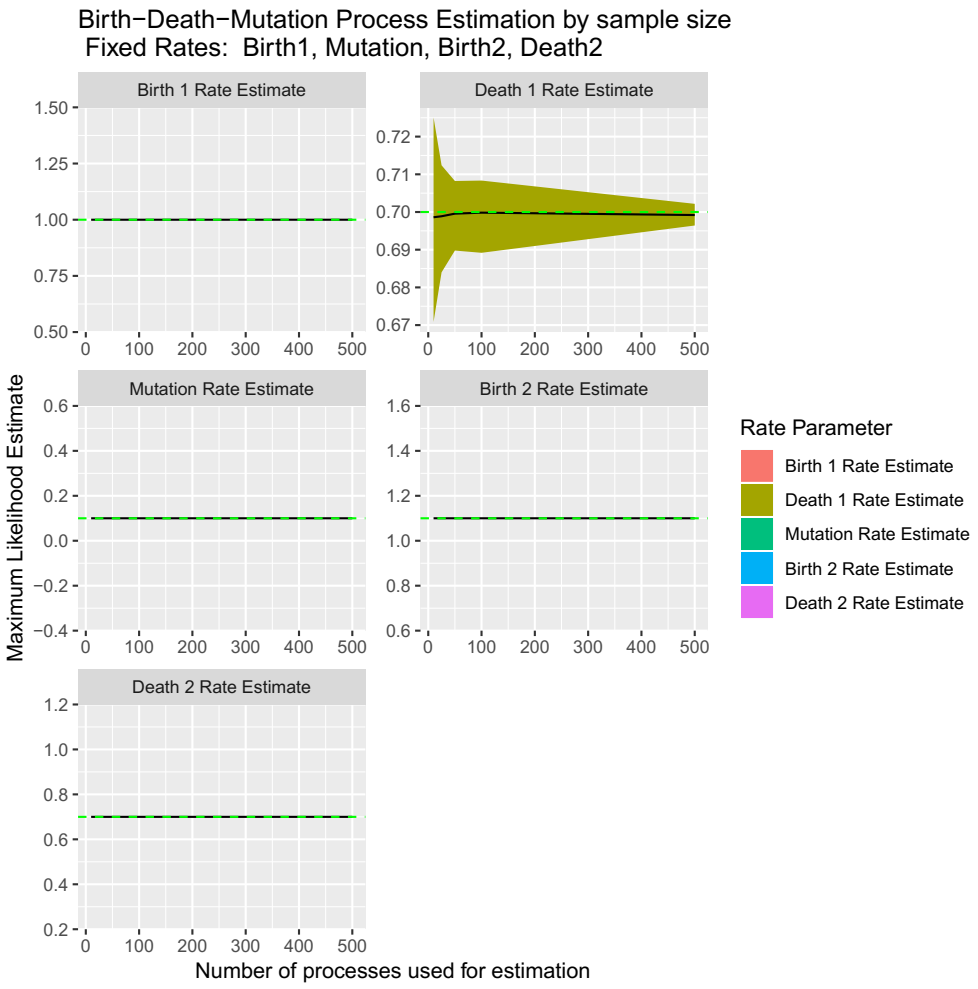

Figure C.12: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

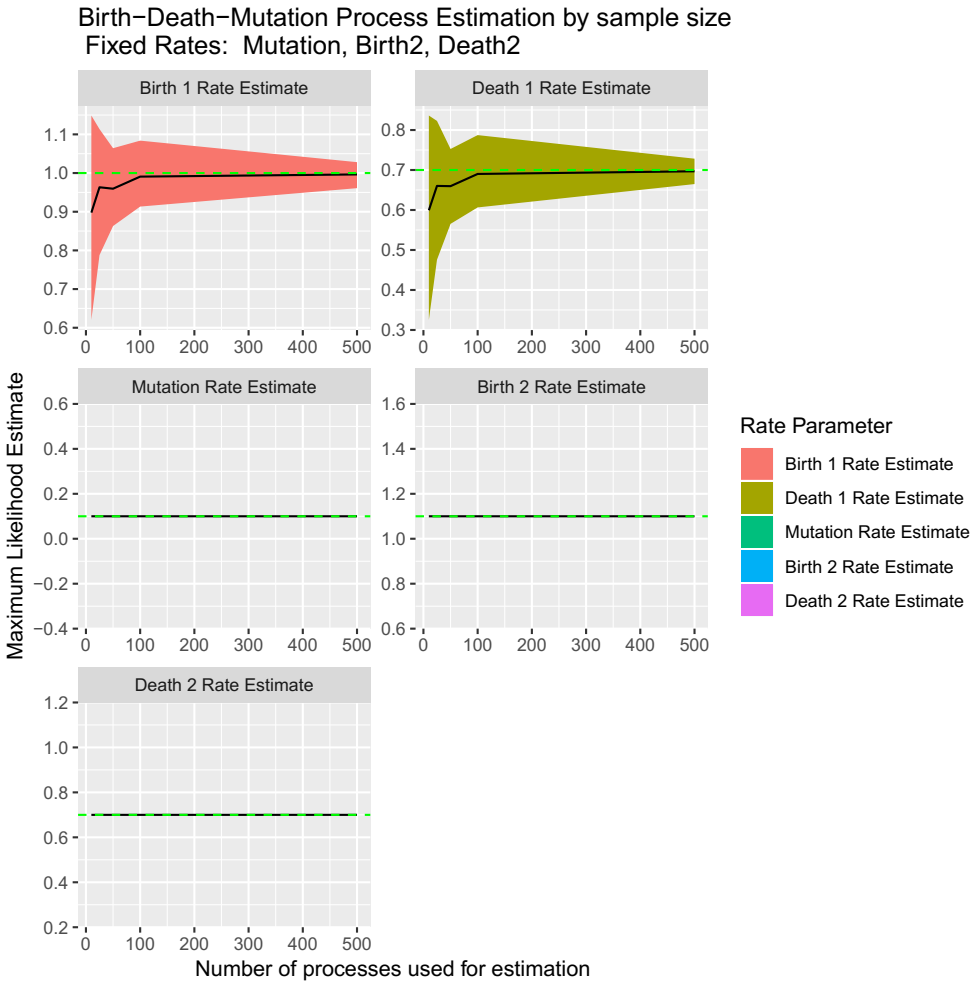

Figure C.13: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

one of the rates is fixed at its true value, estimation for free parameter is much more efficient. We know that it takes fewer samples to accurately characterize the net growth parameter for a birth-death process as shown in the previous Vignette 1. So, when one of birth or death is fixed and net growth reliably estimated, the other parameter can be effectively calculated as a sum or difference of the fixed rate and the net growth.

Estimating the Mutation Rate

Different patterns emerge when investigating the mutation rate from type 1 to type 2. The results from estimating only the mutation rate are shown in Figure C.14. In comparison to estimating all rates simultaneously (Figure C.10), the variance in our estimate is dramatically reduced.

Looking at the scenario in which the birth and death rates of type 1 as well as the mutation rate are estimated simultaneously, the variance of the mutation rate estimate is still relatively low (Figure C.15). In general, the pattern we see is that the variance of the mutation rate estimate is low whenever both the birth and death rate of the type 2 population is fixed. Whenever one or both of these rates are estimated simultaneously with the mutation rate, for small sample sizes we observe an initial increase in the mutation rate which is offset by either a decrease in the type 2 birth rate (Figure C.16) or an increase in the type 2 death rate (Figure C.17 & Figure C.18).

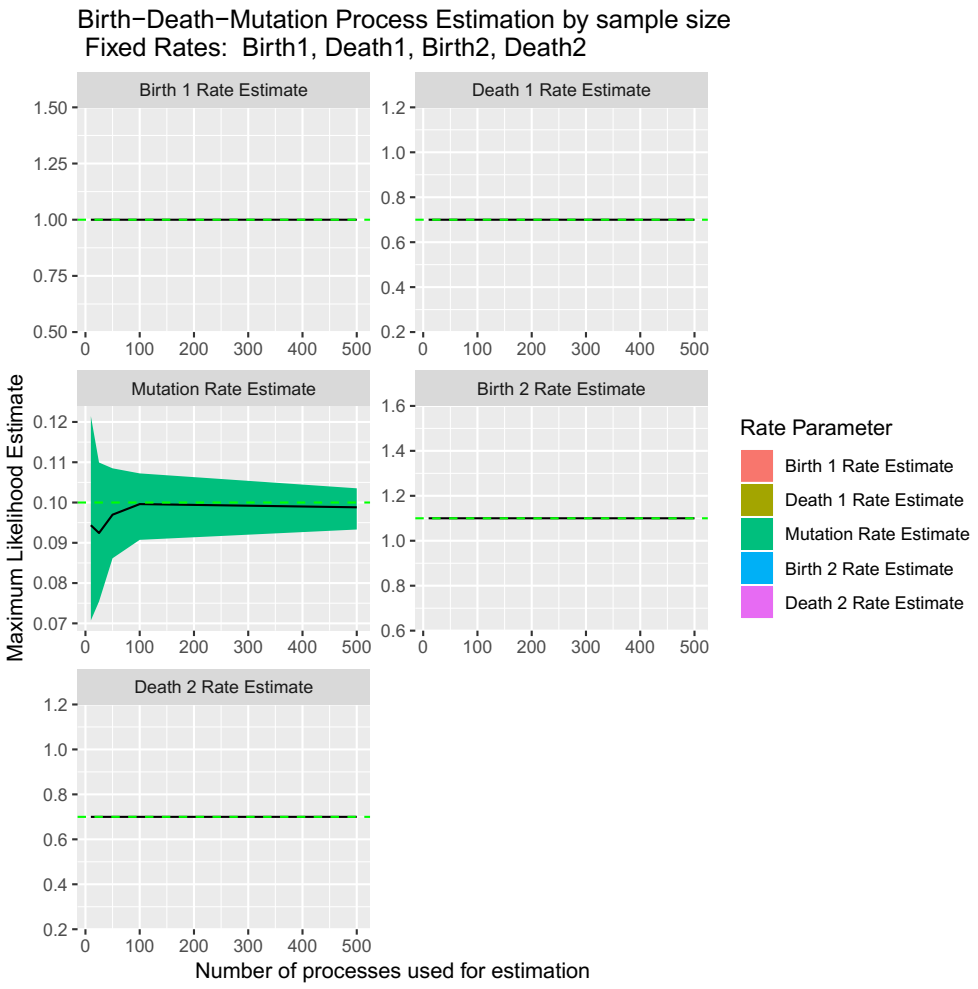

Figure C.14: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

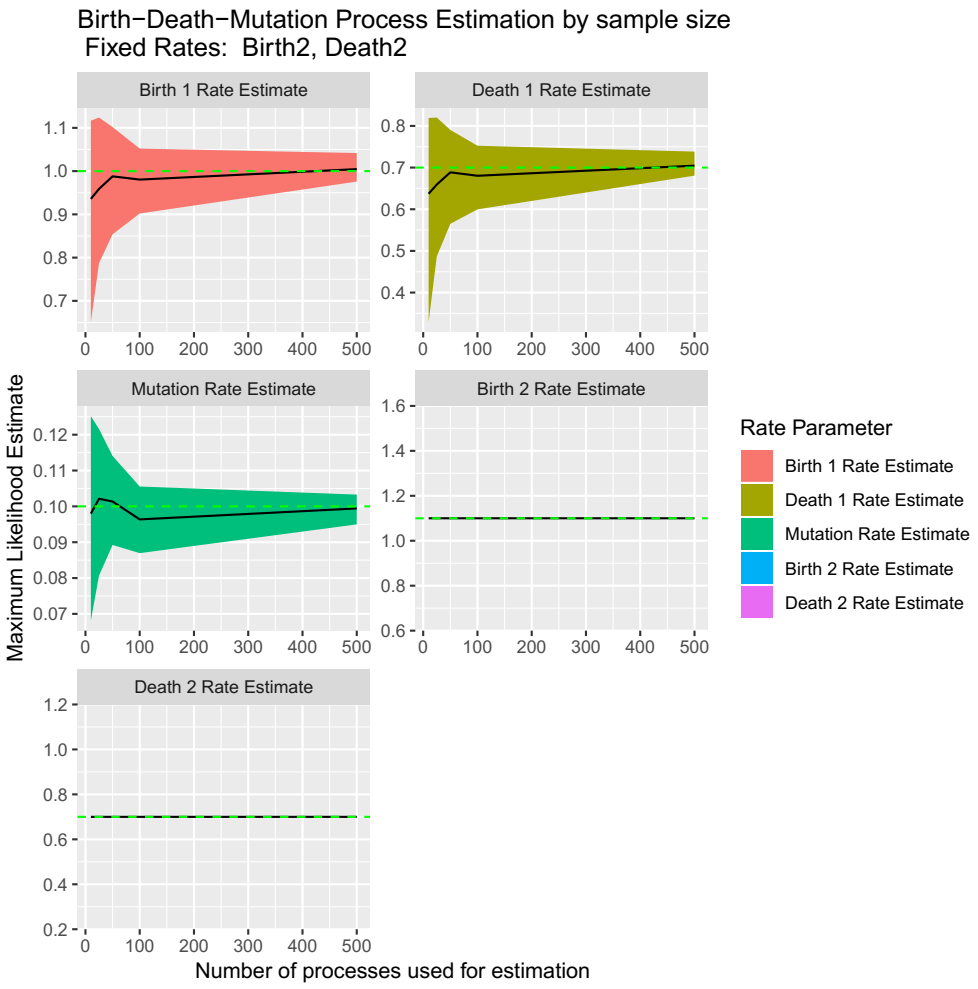

Figure C.15: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

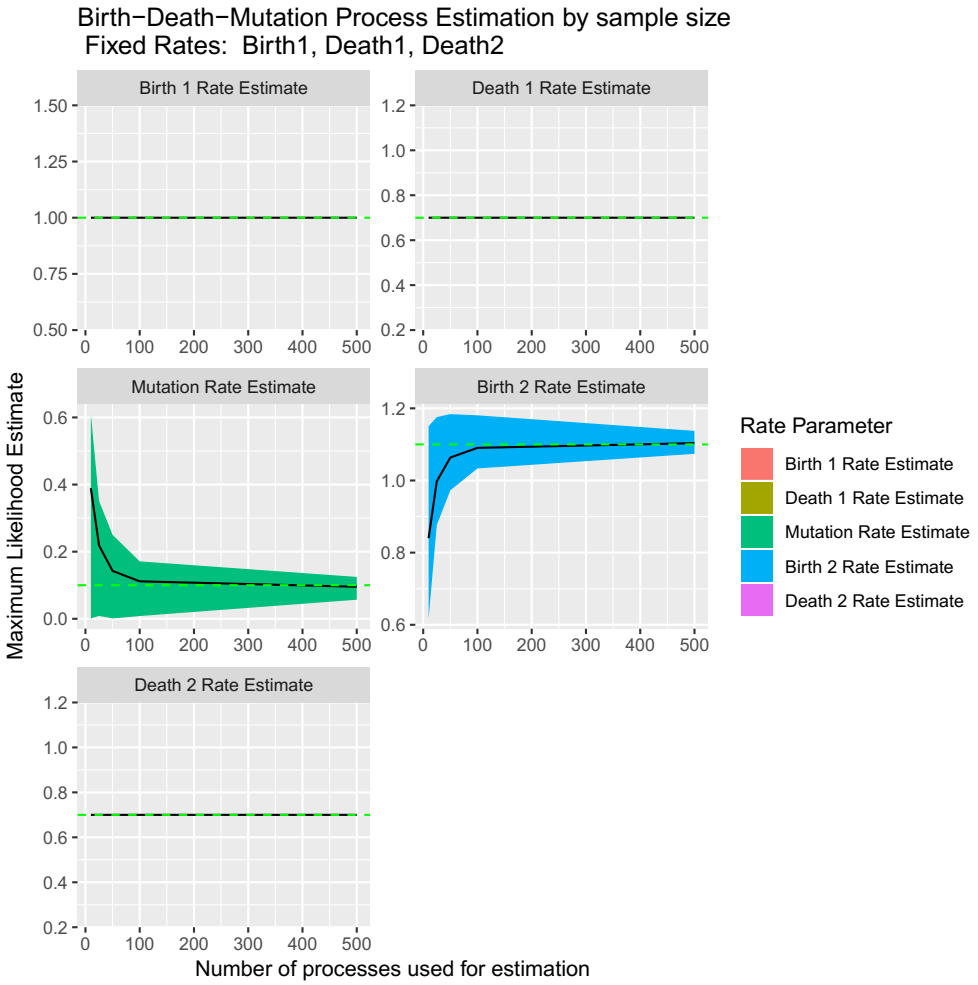

Figure C.16: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

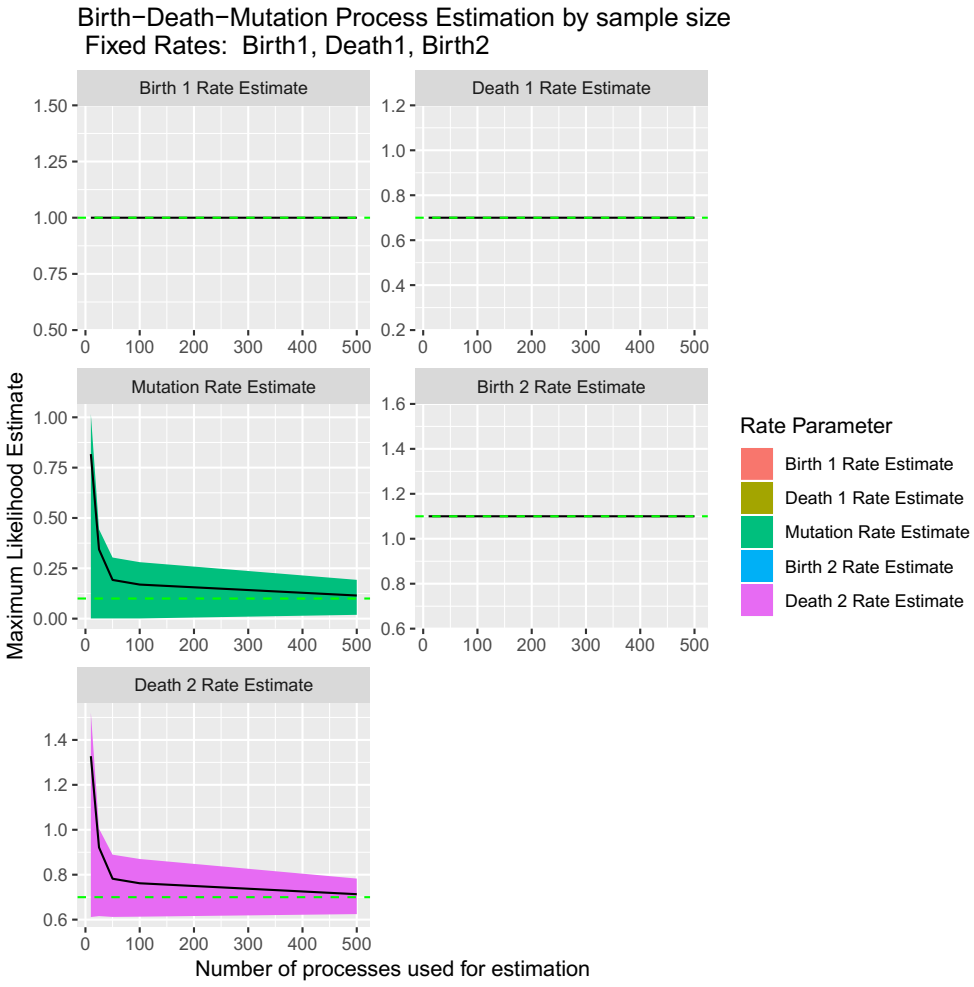

Figure C.17: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

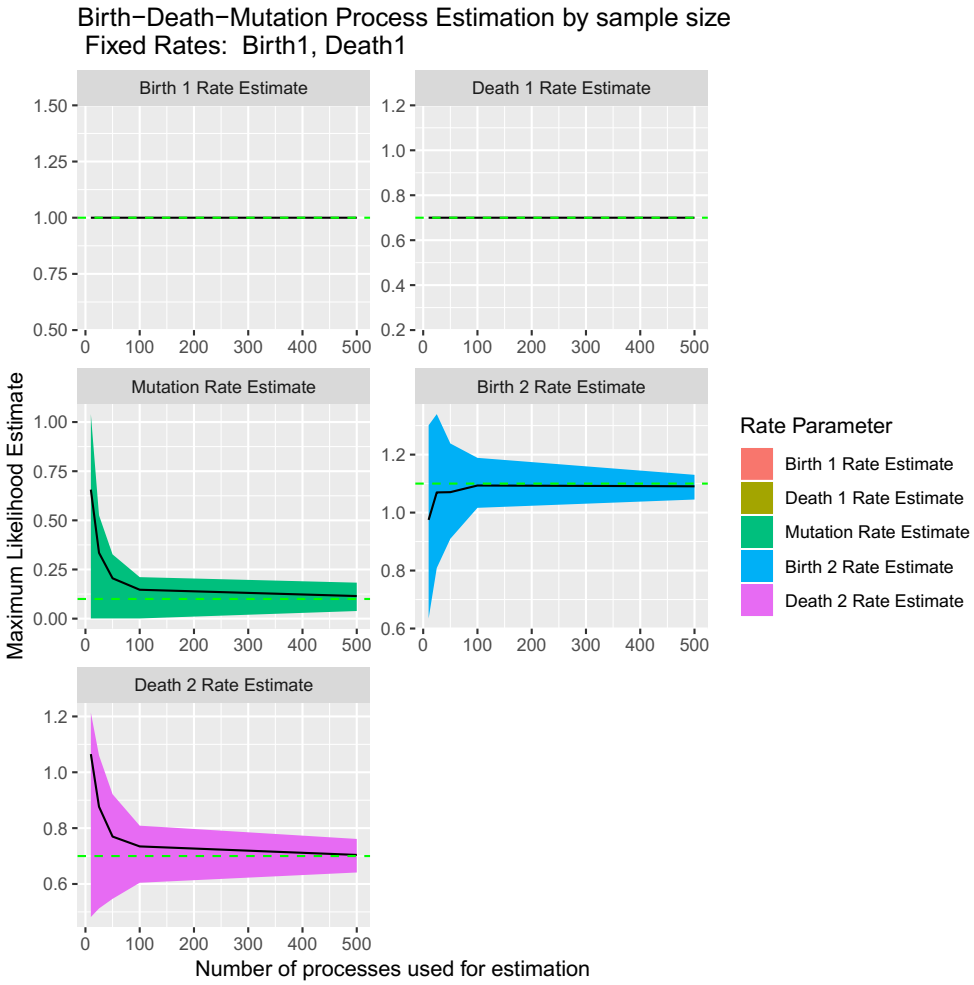

Figure C.18: Mean estimates (black line) and 25th and 75th percentiles (shaded area) for the rate parameters in the two-type birth-death-mutation model plotted for increasing sample sizes. True values are shown as dotted green horizontal lines.

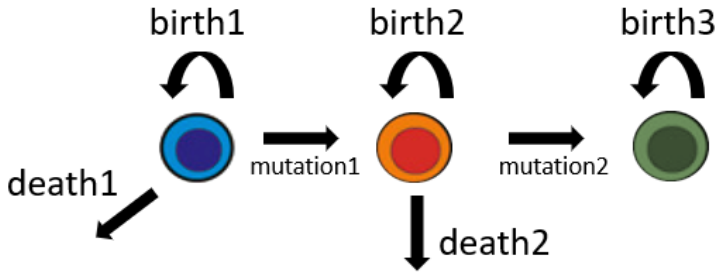

Figure C.19: An example of a Three-type branching process model.

C.4 Vignette 3: Three-Type Process

Background

As an additional example, we investigate a three-type model. Through the previous two vignettes, we show that as model complexity increases, estimation becomes a harder problem requiring more samples to accurately estimate rate parameters. In this vignette, we display the results of estmation in thie three-type model shown in Figure C.19 with 1,000 samples. Even though the estimates may or may not be very close to the true values, ESTIpop provides a method for estimation of this model where none has previously existed.

Simulation using ESTIpop

We begin by simulating 1,000 samples of the model shown above. To speed simulation, we take advantage of parallelization.

```
library(estipop)
library(doParallel)
library(foreach)

# Set up parallelization
cores=detectCores()
cl <- makeCluster(cores[1]-1)
registerDoParallel(cl)

# Set up time to simulate - 1 unit
time = 1

# Set our initial population sizes
initial = c(500,500,500)

# Set some variables for use in our TransitionList
b = 0.7
d = 0.4
tr = 0.2

# Specify the model using TransitionList
transitionList = TransitionList(
        FixedTransition(population = 0, rate = b,
                fixed = c(2, 0, 0)),
        FixedTransition(population = 0, rate = d,
                fixed = c(0, 0, 0)),
        FixedTransition(population = 0, rate = tr,
                fixed = c(1, 1, 0)),
        FixedTransition(population = 1, rate = b-.1,
                fixed = c(0, 2, 0)),
        FixedTransition(population = 1, rate = d,
                fixed = c(0, 0, 0)),
        FixedTransition(population = 1, rate = tr-.1,
                fixed = c(0, 1, 1)),
        FixedTransition(population = 2, rate = b+.1,
                fixed = c(0, 0, 2)))

# Store off our true rates
truth = c(b, d, tr, b-.1, d, tr-.1, b+.1)
```

```
# No StopList items
stopList = StopList()

# Simulate 1000 samples
ntrials = 1000

full_res = foreach(j_=1:ntrials, .combine = "rbind") %dopar%{
        res = branch(time, initial, transitionList, stopList)
        as.matrix(res)
}
full_res = na.omit(full_res)
data = as.matrix(full_res[full_res[,1] == 1,2:4])
```

Above, we specify that we will simulate the model for one unit of time with initial population sizes of 500 for each type. Using parallelization, we simulate 1,000 such data points.

Table C.6: Software classes available in DIFFPop

	Birth 1 Rate	Death 1 Rate	Mutation 1 Rate	Birth 2 Rate	Death 2 Rate	Mutation 2 Rate	Birth 3 Rate
truth	0.7000000	0.4000000	0.2000000	0.6000000	0.4000000	0.1000000	0.8000000
estimates	0.6921017	0.3935928	0.2099199	0.6089661	0.4146292	0.1499245	0.7601946

Estimation using ESTIpop

We estimate the rates using ESTIpop and the following script:

```
# Set up our estimation parameters
N = c(500, 500, 500)

time = 1

transitionList = TransitionList(
        FixedTransition(population = 0, fixed = c(2, 0, 0)),
        FixedTransition(population = 0, fixed = c(0, 0, 0)),
        FixedTransition(population = 0, fixed = c(1, 1, 0)),
        FixedTransition(population = 1, fixed = c(0, 2, 0)),
        FixedTransition(population = 1, fixed = c(0, 0, 0)),
        FixedTransition(population = 1, fixed = c(0, 1, 1)),
        FixedTransition(population = 2, fixed = c(0, 0, 2)))

initial = c(b, d, tr+.2, b-.1, d, tr-.1, b+.1)

# Estimate using the estimateBP function
estimates_obj = estimateBP(time = time,
        N = N,
        transitionList = transitionList,
        data = data,
        initial = initial)
estimates = estimates_obj$par
```

In Table C.6 we display the true rate parameters and the corresponding estimates. We ob-

serve that with 1,000 observations, our estimated rate parameters are actually not far from the true values. With fewer observations, we would be more likely to estimate rate parameters further from their known true value.

References

[1] Department of Homeland Security. Annual Flow Report; U.S. Lawful Permanent Residents 2013. http://www.dhs.gov/sites/default/files/publications/ois_lpr_fr_2013.pdf.

[2] International Myeloma Foundation: iStopMM: Black Swann Initiative. https://www.myeloma.org/istopmm.

[3] National Cancer Institute: Cancer incidence: Surveillance, Epidemiology, and End Results (SEER) registries research data. http://seer.cancer.gov/data.

[4] National Cancer Institute: Surveillance, Epidemiology, and End Results (SEER) program populations (1969-2014). http://www.seer.cancer.gov/popdata.

[5] The World Bank: World development indicators: Crude birth rate in the U.S. http://databank.worldbank.org/data/reports.aspx?source=2&type=metadata&series=SP.DYN.CBRT.IN.

[6] The World Bank. World Development Indocators: Crude Birth Rate in the U.S. http://databank.worldbank.org/data/reports.aspx?source=2&type=metadata&series=SP.DYN.CBRT.IN.

[7] US Census Bureau: American FactFinder. http://factfinder2.census.gov.

[8] World Health Organization: Global Health Observatory data repository 2013. http://apps.who.int/gho/data/?theme=main&vid=60630.

[9] Aas, T., Børresen, A.-L., Geisler, S., Smith-Sørensen, B., Johnsen, H., Varhaug, J. E., Akslen, L. A., and Lønning, P. E. Specific p53 mutations are associated with de novo resistance to doxorubicin in breast cancer patients. *Nature medicine 2*, 7 (1996), 811.

[10] Akunuru, S., and Geiger, H. Aging, clonality, and rejuvenation of hematopoietic stem cells. *Trends in molecular medicine 22*, 8 (2016), 701–712.

[11] Athreya, K. B., and Ney, P. *Branching processes [by] K. B. Athreya [and] P. E. Ney*. Springer-Verlag Berlin, New York, 1972.

[12] Bhang, H.-e. C., Ruddy, D. A., Radhakrishna, V. K., Caushi, J. X., Zhao, R., Hims, M. M., Singh, A. P., Kao, I., Rakiec, D., Shaw, P., et al. Studying clonal dynamics in response to cancer therapy using high-complexity barcoding. *Nature medicine 21*, 5 (2015), 440.

[13] Birmann, B. M., Giovannucci, E., Rosner, B., Anderson, K. C., and Colditz, G. A. Body mass index, physical activity, and risk of multiple myeloma. *Cancer Epidemiology and Prevention Biomarkers 16*, 7 (2007), 1474–1478.

[14] Birmann, B. M., Giovannucci, E. L., Rosner, B. A., and Colditz, G. A. Regular aspirin use and risk of multiple myeloma: a prospective analysis in the health professionals follow-up study and nurses' health study. *Cancer prevention research 7*, 1 (2014), 33–41.

[15] Blair, L. M., and Feldman, M. W. The role of climate and out-of-africa migration in the frequencies of risk alleles for 21 human diseases. *BMC genetics 16*, 1 (2015), 81.

[16] Busch, K., Klapproth, K., Barile, M., Flossdorf, M., Holland-Letz, T., Schlenner, S. M., Reth, M., Höfer, T., and Rodewald, H.-R. Fundamental properties of unperturbed haematopoiesis from stem cells in vivo. *Nature 518*, 7540 (2015), 542.

[17] Carson, K., Bates, M., and Tomasson, M. The skinny on obesity and plasma cell myeloma: a review of the literature. *Bone Marrow Transplantation 49*, 8 (2014), 1009.

[18] Chang, S.-H., Luo, S., O'Brian, K. K., Thomas, T. S., Colditz, G. A., Carlsson, N. P., and Carson, K. R. Association between metformin use and progression of monoclonal gammopathy of undetermined significance to multiple myeloma in us veterans with diabetes mellitus: a population-based retrospective cohort study. *The Lancet Haematology 2*, 1 (2015), e30–e36.

[19] Chmielecki, J., Foo, J., Oxnard, G. R., Hutchinson, K., Ohashi, K., Somwar, R., Wang, L., Amato, K. R., Arcila, M., Sos, M. L., et al. Optimization of dosing for egfr-mutant non–small cell lung cancer with evolutionary cancer modeling. *Science translational medicine 3*, 90 (2011), 90ra59–90ra59.

[20] Cole, C. E., Ballandby, R., Schroeder, J. E., Lee, J. A., Mathiason, M. A., Hoeg, R. T., Bottner, W. A., Farnen, J. P., Ettinger, R. S., Peters, A. M., et al. Assessment of psychological distress in patients suffering from hematological disorders., 2007.

[21] Dagogo-Jack, I., and Shaw, A. T. Tumour heterogeneity and resistance to cancer therapies. *Nature reviews Clinical oncology 15*, 2 (2018), 81.

[22] Duffy, S. W., Nagtegaal, I. D., Wallis, M., Cafferty, F. H., Houssami, N., Warwick, J., Allgood, P. C., Kearins, O., Tappenden, N., O'sullivan, E., et al. Correcting for lead time and length bias in estimating the effect of screen detection on cancer survival. *American journal of epidemiology 168*, 1 (2008), 98–104.

[23] Durrett, R. Branching process models of cancer. In *Branching process models of cancer*. Springer, 2015, pp. 1–63.

[24] Eddelbuettel, D., François, R., Allaire, J., Ushey, K., Kou, Q., Russel, N., Chambers, J., and Bates, D. Rcpp: Seamless r and c++ integration. *Journal of Statistical Software 40*, 8 (2011), 1–18.

[25] Frenken, K., and Boschma, R. A. A theoretical framework for evolutionary economic geography: industrial dynamics and urban growth as a branching process. *Journal of economic geography 7*, 5 (2007), 635–649.

[26] Ganuza, M., Hall, T., Finkelstein, D., Chabot, A., Kang, G., and McKinney-Freeman, S. Lifelong haematopoiesis is established by hundreds of precursors throughout mammalian ontogeny. *Nature cell biology 19*, 10 (2017), 1153.

[27] Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry 81*, 25 (1977), 2340–2361.

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

[29] Go, R. S., Gundrum, J. D., and Neuner, J. M. Determining the clinical significance of monoclonal gammopathy of undetermined significance: a seer–medicare population analysis. *Clinical Lymphoma Myeloma and Leukemia 15*, 3 (2015), 177–186.

[30] Gordis, L. Epidemiology. 4th, 2013.

[31] Haccou, P., Haccou, P., Jagers, P., Vatutin, V. A., and Vatutin, V. *Branching processes: variation, growth, and extinction of populations*. No. 5. Cambridge university press, 2005.

[32] Hayes, J. H., and Barry, M. J. Screening for prostate cancer with the prostate-specific antigen test: a review of current evidence. *Jama 311*, 11 (2014), 1143–1149.

[33] Hideshima, T., Mitsiades, C., Tonon, G., Richardson, P. G., Anderson, K. C., et al. Understanding multiple myeloma pathogenesis in the bone marrow to identify new therapeutic targets. *Nature Reviews Cancer 7*, 8 (2007), 585.

[34] Holohan, C., Van Schaeybroeck, S., Longley, D. B., and Johnston, P. G. Cancer drug resistance: an evolving paradigm. *Nature Reviews Cancer 13*, 10 (2013), 714.

[35] Howlader, N., Noone, A., Krapcho, M., Miller, D., Bishop, K., Altekruse, S., Kosary, C., Yu, M., Ruhl, J., Tatalovich, Z., et al. Seer cancer statistics review, 1975–2013. *Bethesda, MD: National Cancer Institute 19* (2016).

[36] Jaiswal, S., Fontanillas, P., Flannick, J., Manning, A., Grauman, P. V., Mar, B. G., Lindsley, R. C., Mermel, C. H., Burtt, N., Chavez, A., et al. Age-related clonal hematopoiesis associated with adverse outcomes. *New England Journal of Medicine 371*, 26 (2014), 2488–2498.

[37] Katzmann, J. A. Screening panels for monoclonal gammopathies: time to change. *The Clinical Biochemist Reviews 30*, 3 (2009), 105.

[38] Khimani, F., Curley, B., and Almubarak, M. Survey of patients referred to a university cancer center for benign hematology: quality measures and patient understanding. *Journal of oncology practice 11*, 1 (2014), 26–29.

[39] Korde, N., Kristinsson, S. Y., and Landgren, O. Monoclonal gammopathy of undetermined significance (mgus) and smoldering multiple myeloma (smm): novel biological insights and development of early treatment strategies. *Blood 117*, 21 (2011), 5573–5581.

[40] Kovatchev, B. P., Breton, M., Dalla Man, C., and Cobelli, C. In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes, 2009.

[41] Kuehl, W. M., and Bergsagel, P. L. Multiple myeloma: evolving genetic events and host interactions. *Nature Reviews Cancer 2*, 3 (2002), 175.

[42] Kumar, S. K., Rajkumar, S. V., Dispenzieri, A., Lacy, M. Q., Hayman, S. R., Buadi, F. K., Zeldenrust, S. R., Dingli, D., Russell, S. J., Lust, J. A., et al. Improved survival in multiple myeloma and the impact of novel therapies. *Blood 111*, 5 (2008), 2516–2520.

[43] Kyle, R., Durie, B., Rajkumar, S. V., Landgren, O., Bladé, J., Merlini, G., Kröger, N., Einsele, H., Vesole, D., Dimopoulos, M., et al. Monoclonal gammopathy of undetermined significance (mgus) and smoldering (asymptomatic) multiple myeloma: Imwg consensus perspectives risk factors for progression and guidelines for monitoring and management. *Leukemia 24*, 6 (2010), 1121.

[44] Kyle, R. A., Therneau, T. M., Rajkumar, S. V., Larson, D. R., Plevak, M. F., Offord, J. R., Dispenzieri, A., Katzmann, J. A., and Melton III, L. J. Prevalence of monoclonal gammopathy of undetermined significance. *New England Journal of Medicine 354*, 13 (2006), 1362–1369.

[45] Landgren, O., Graubard, B., Kumar, S., Kyle, R., Katzmann, J., Murata, K., Costello, R., Dispenzieri, A., Caporaso, N., Mailankody, S., et al. Prevalence of myeloma precursor state monoclonal gammopathy of undetermined significance in 12372 individuals 10–49 years old: a population-based study from the national health and nutrition examination survey. *Blood Cancer Journal 7*, 10 (2017), e618.

[46] Landgren, O., Graubard, B. I., Katzmann, J. A., Kyle, R. A., Ahmadizadeh, I., Clark, R., Kumar, S. K., Dispenzieri, A., Greenberg, A. J., Therneau, T. M., et al. Racial disparities in the prevalence of monoclonal gammopathies: a population-based study of 12 482 persons from the national health and nutritional examination survey. *Leukemia 28*, 7 (2014), 1537.

[47] Landgren, O., Gridley, G., Turesson, I., Caporaso, N. E., Goldin, L. R., Baris, D., Fears, T. R., Hoover, R. N., and Linet, M. S. Risk of monoclonal gammopathy of undetermined significance (mgus) and subsequent multiple myeloma among african american and white veterans in the united states. *Blood 107*, 3 (2006), 904–906.

[48] Landgren, O., Katzmann, J. A., Hsing, A. W., Pfeiffer, R. M., Kyle, R. A., Yeboah, E. D., Biritwum, R. B., Tettey, Y., Adjei, A. A., Larson, D. R., et al. Prevalence of

monoclonal gammopathy of undetermined significance among men in ghana. In *Mayo Clinic Proceedings* (2007), vol. 82, Elsevier, pp. 1468–1473.

[49] Landgren, O., Kyle, R. A., Pfeiffer, R. M., Katzmann, J. A., Caporaso, N. E., Hayes, R. B., Dispenzieri, A., Kumar, S., Clark, R. J., Baris, D., et al. Monoclonal gammopathy of undetermined significance (mgus) consistently precedes multiple myeloma: a prospective study. *Blood 113*, 22 (2009), 5412–5417.

[50] Landgren, O., and Weiss, B. Patterns of monoclonal gammopathy of undetermined significance and multiple myeloma in various ethnic/racial groups: support for genetic factors in pathogenesis. *Leukemia 23*, 10 (2009), 1691.

[51] Lewis, P. W., and Shedler, G. S. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly 26*, 3 (1979), 403–413.

[52] McDonald, T. O., and Kimmel, M. A multitype infinite-allele branching process with applications to cancer evolution. *Journal of Applied Probability 52*, 3 (2015), 864–876.

[53] McDonald, T. O., and Michor, F. Siapopr: a computational method to simulate evolutionary branching trees for analysis of tumor clonal evolution. *Bioinformatics 33*, 14 (2017), 2221–2223.

[54] McLellan, L., Pohlman, B., Rybicki, L., Foster, L., Tench, S., Cooper, M., McKenzie, M., Kilbane, M., Wright, C., Dimitrov, J., et al. Distress screening scores of malignant and benign hematology patients: Results of a pilot project., 2012.

[55] McShane, C. M., Murphy, B., Lim, K. H., and Anderson, L. A. Monoclonal gammopathy of undetermined significance as viewed by haematology healthcare professionals. *European journal of haematology 100*, 1 (2018), 20–26.

[56] Mergenthaler, U., Heymanns, J., Köppler, H., Thomalla, J., van Roye, C., Schenk, J., and Weide, R. Evaluation of psychosocial distress in patients treated in a community-based oncology group practice in germany. *Annals of Oncology 22*, 4 (2010), 931–938.

[57] Moran, P. A. P. *The statistical process of evolutionary theory*. Clarendon Press, 1962.

[58] Orkin, S. H., and Zon, L. I. Hematopoiesis: an evolving paradigm for stem cell biology. *Cell 132*, 4 (2008), 631–644.

[59] Otto, S. P., and Day, T. *A biologist's guide to mathematical modeling in ecology and evolution*. Princeton University Press, 2011.

[60] Pakes, A. G. An infinite alleles version of the markov branching process. *Journal of the Australian Mathematical Society 46*, 1 (1989), 146–169.

[61] Palumbo, A., and Anderson, K. Multiple myeloma. *New England Journal of Medicine 364*, 11 (2011), 1046–1060. PMID: 21410373.

[62] Quintás-Cardama, A., Kantarjian, H. M., and Cortes, J. E. Mechanisms of primary and secondary resistance to imatinib in chronic myeloid leukemia. *Cancer control 16*, 2 (2009), 122–131.

[63] Raab, M., Podar, K., Breitkreutz, I., et al. Multiple myeloma. *Lancet 374* (2009), 324–339.

[64] Rajkumar, S. V. Multiple myeloma: 2013 update on diagnosis, risk-stratification, and management. *American journal of hematology 88*, 3 (2013), 225–235.

[65] Rodriguez-Fraticelli, A. E., Wolock, S. L., Weinreb, C. S., Panero, R., Patel, S. H., Jankovic, M., Sun, J., Calogero, R. A., Klein, A. M., and Camargo, F. D. Clonal analysis of lineage fate in native haematopoiesis. *Nature 553*, 7687 (2018), 212.

[66] Rosiñol, L., Cibeira, M. T., Montoto, S., Rozman, M., Esteve, J., Filella, X., and Bladé, J. Monoclonal gammopathy of undetermined significance: predictors of malignant transformation and recognition of an evolving type characterized by a progressive increase in m protein size. In *Mayo Clinic Proceedings* (2007), vol. 82, Elsevier, pp. 428–434.

[67] Ryser, M. D., Worni, M., Turner, E. L., Marks, J. R., Durrett, R., and Hwang, E. S. Outcomes of active surveillance for ductal carcinoma in situ: a computational risk analysis. *Journal of the National Cancer Institute 108*, 5 (2015), djv372.

[68] Sasieni, P. D., and Adams, J. Standardized lifetime risk. *American journal of epidemiology 149*, 9 (1999), 869–875.

[69] Schinasi, L. H., Brown, E. E., Camp, N. J., Wang, S. S., Hofmann, J. N., Chiu, B. C., Miligi, L., Beane Freeman, L. E., de Sanjose, S., Bernstein, L., et al. Multiple

myeloma and family history of lymphohaematopoietic cancers: Results from the international multiple myeloma consortium. *British journal of haematology 175*, 1 (2016), 87–101.

[70] Sigurdardottir, E. E., Turesson, I., Lund, S. H., Lindqvist, E. K., Mailankody, S., Korde, N., Björkholm, M., Landgren, O., and Kristinsson, S. Y. The role of diagnosis and clinical follow-up of monoclonal gammopathy of undetermined significance on survival in multiple myeloma. *JAMA oncology 1*, 2 (2015), 168–174.

[71] Steensma, D. P., Bejar, R., Jaiswal, S., Lindsley, R. C., Sekeres, M. A., Hasserjian, R. P., and Ebert, B. L. Clonal hematopoiesis of indeterminate potential and its distinction from myelodysplastic syndromes. *Blood 126*, 1 (2015), 9–16.

[72] Sun, J., Ramos, A., Chapman, B., Johnnidis, J. B., Le, L., Ho, Y.-J., Klein, A., Hofmann, O., and Camargo, F. D. Clonal dynamics of native haematopoiesis. *Nature 514*, 7522 (2014), 322.

[73] Tabayoyong, W., and Abouassaly, R. Prostate cancer screening and the associated controversy. *Surgical Clinics 95*, 5 (2015), 1023–1039.

[74] Tavaré, S. The linear birth–death process: an inferential retrospective. *Advances in Applied Probability 50*, A (2018), 253–269.

[75] Therneau, T. M., Kyle, R. A., Melton III, L. J., Larson, D. R., Benson, J. T., Colby, C. L., Dispenzieri, A., Kumar, S., Katzmann, J. A., Cerhan, J. R., et al. Incidence of monoclonal gammopathy of undetermined significance and estimation of duration before first clinical recognition. In *Mayo Clinic Proceedings* (2012), vol. 87, Elsevier, pp. 1071–1079.

[76] Wallin, A., and Larsson, S. C. Body mass index and risk of multiple myeloma: a meta-analysis of prospective studies. *European Journal of Cancer 47*, 11 (2011), 1606–1615.

[77] Wattel, E., Preudhomme, C., Hecquet, B., Vanrumbeke, M., Quesnel, B., Dervite, I., Morel, P., and Fenaux, P. p53 mutations are associated with resistance to chemotherapy and short survival in hematologic malignancies. *Blood 84*, 9 (1994), 3148–3157.

[78] Waxman, A. J., Mink, P. J., Devesa, S. S., Anderson, W. F., Weiss, B. M., Kristinsson, S. Y., McGlynn, K. A., and Landgren, O. Racial disparities in incidence and outcome in multiple myeloma: a population-based study. *Blood 116*, 25 (2010), 5501–5506.

[79] Yakovlev, A. Y., Yanev, N. M., et al. Relative frequencies in multitype branching processes. *The annals of applied probability 19*, 1 (2009), 1–14.

[80] Zelen, M. Optimal scheduling of examinations for the early detection of disease. *Biometrika 80*, 2 (1993), 279–293.

[81] Zelen, M., and Feinleib, M. On the theory of screening for chronic diseases. *Biometrika 56*, 3 (1969), 601–614.

[82] Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS) 23*, 4 (1997), 550–560.

[83] Zingone, A., and Kuehl, W. M. Pathogenesis of monoclonal gammopathy of undetermined significance and progression to multiple myeloma. In *Seminars in hematology* (2011), vol. 48, Elsevier, pp. 4–12.