

# UC San Diego

## UC San Diego Electronic Theses and Dissertations

### Title

Mathematical Modeling of Viral Evolution and Epidemiology

### Permalink

<https://escholarship.org/uc/item/62s7q92d>

### Author

Moshiri, Alexander Niema

### Publication Date

2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

**Mathematical Modeling of Viral Evolution and Epidemiology**

A dissertation submitted in partial satisfaction of the  
requirements for the degree  
Doctor of Philosophy

in

Bioinformatics and Systems Biology

by

Alexander Niema Moshiri

Committee in charge:

Professor Siavash Mirarab, Chair  
Professor Pavel A. Pevzner, Co-Chair  
Professor Vineet Bafna  
Professor David M. Smith  
Professor Joel O. Wertheim

2019

Copyright  
Alexander Niema Moshiri, 2019  
All rights reserved.

The dissertation of Alexander Niema Moshiri is approved,  
and it is acceptable in quality and form for publication on  
microfilm and electronically:

---

---

---

---

Co-Chair

---

Chair

University of California San Diego

2019



## DEDICATION

I dedicate this dissertation to Ladan Moshiri, Ramin Moshiri, and Michelle Roxanna Moshiri-Warncke for years of love and support.

I dedicate this dissertation to Karen George for always being by my side.

I dedicate this dissertation to Ryan Micallef and Felix Garcia for teaching me to live my life a quarter mile at a time.

I dedicate this dissertation to all my friends in the Bioinformatics and Systems Biology program for some of the best memories of my life.

## EPIGRAPH

*Folk in these stories had lots of chances of turning back, only they didn't. Because they were holding onto something.*

*What are we holding onto, Sam?*

*That there's some good in this world, Mr. Frodo.  
And it's worth fighting for.*

—J.R.R. Tolkien

## TABLE OF CONTENTS

Signature Page . . . . .	iii
Dedication . . . . .	iv
Epigraph . . . . .	v
Table of Contents . . . . .	vi
List of Abbreviations . . . . .	ix
List of Figures . . . . .	xii
List of Tables . . . . .	xiv
Acknowledgements . . . . .	xv
Vita . . . . .	xvii
Abstract of the Dissertation . . . . .	xviii
Introduction . . . . .	1
Chapter 1      FAVITES: Simultaneous Simulation of Transmission Networks, Phylogenetic Trees, and Sequences . . . . .	7
1.1    Introduction . . . . .	8
1.2    Materials and Methods . . . . .	10
1.2.1    FAVITES Simulation Process . . . . .	10
1.2.2    Experimental Setup . . . . .	17
1.3    Results . . . . .	23
1.3.1    Comparison to Real Phylogenies . . . . .	23
1.3.2    Impact of Parameter Choices On the Epidemiology . . . . .	25
1.3.3    Evaluating Inference Methods . . . . .	28
1.4    Discussion . . . . .	30
1.5    Acknowledgements . . . . .	33
Chapter 2      A Two-State Model of Tree Evolution and its Applications to <i>Alu</i> Retrotransposition . . . . .	34
2.1    Introduction . . . . .	35
2.2    Materials and Methods . . . . .	38
2.2.1    Dual-Birth Model . . . . .	38
2.2.2    Theoretical Properties of the Dual-Birth Model . . . . .	40
2.2.3    Simulation Setup . . . . .	50

	2.2.4	Human <i>Alu</i> Dataset . . . . .	51
2.3		Results . . . . .	53
	2.3.1	Simulations: Dual-Birth Model . . . . .	53
	2.3.2	Simulations: Model Violations . . . . .	57
	2.3.3	Human <i>Alu</i> Analysis . . . . .	58
2.4		Discussion . . . . .	60
	2.4.1	Comparison to Other Models . . . . .	60
	2.4.2	Properties of the Dual-Birth . . . . .	63
	2.4.3	<i>Alu</i> Repeats . . . . .	65
2.5		Data Availability . . . . .	67
2.6		Acknowledgements . . . . .	67
Chapter 3		ProACT: Prioritization Using Ancestral Edge Lengths . . . . .	68
	3.1	Introduction . . . . .	69
	3.2	New Approaches . . . . .	71
		3.2.1 Motivating the Approach . . . . .	72
		3.2.2 Formal Description . . . . .	74
	3.3	Results . . . . .	74
		3.3.1 Simulation Results . . . . .	75
		3.3.2 Real San Diego Dataset . . . . .	80
	3.4	Discussion . . . . .	83
		3.4.1 Discussion of Results . . . . .	83
		3.4.2 Implications of Results . . . . .	85
	3.5	Materials and Methods . . . . .	87
		3.5.1 Simulated Datasets . . . . .	87
		3.5.2 San Diego Dataset . . . . .	89
		3.5.3 Evaluation Procedure . . . . .	89
	3.6	Acknowledgments . . . . .	91
Chapter 4		TreeSwift: A Massively Scalable Python Tree Package . . . . .	93
	4.1	Motivation and Significance . . . . .	94
	4.2	Software Description . . . . .	95
		4.2.1 Software Overview . . . . .	95
		4.2.2 Software Functionalities . . . . .	97
	4.3	Illustrative Example . . . . .	97
	4.4	Impact . . . . .	99
	4.5	Conclusions . . . . .	99
	4.6	Acknowledgements . . . . .	100
Chapter 5		Bioinformatics Education . . . . .	101
	5.1	Introduction . . . . .	102
		5.1.1 Bioinformatics Education: The New Frontier . . . . .	102
		5.1.2 The MOOC Revolution . . . . .	103

5.1.3	From MOOCs to MAITs . . . . .	103
5.1.4	“Bioinformatics” Means Nobody Gets Left Behind . . . . .	104
5.2	Methods . . . . .	104
5.2.1	Teaching Philosophy . . . . .	104
5.3	Results . . . . .	108
5.4	Discussion . . . . .	109
Appendix A	Supplemental Material for Chapter 1 . . . . .	112
Appendix B	Supplemental Material for Chapter 2 . . . . .	123
B.1	Theoretical Results . . . . .	124
B.1.1	Proofs . . . . .	124
B.1.2	Set of All Possible Orderings . . . . .	129
B.2	Supplementary Methods . . . . .	130
B.2.1	Simulation Setup . . . . .	130
B.2.2	Human <i>Alu</i> Analyses . . . . .	136
B.3	Supplementary Figures . . . . .	137
Appendix C	Supplemental Material for Chapter 3 . . . . .	147
Bibliography	. . . . .	155

## LIST OF ABBREVIATIONS

***pol*** Polymerase. 20, 22, 31, 71, 80, 89, 116, 153

**API** Application Program Interface. 8, 11

**ART** Antiretroviral Therapy. 2, 8, 19, 30, 69–72, 75–78, 83, 84, 88, 89, 153

**BA** Barabási–Albert. 12, 18, 21, 25, 27, 88, 114, 118, 119, 121

**BF** Blum and François (2006). 61

**bp** base pairs. 37, 57

**CHRP** California HIV/AIDS Research Program. 33

**CMA** Cumulative Moving Average. 75–77, 90, 150, 151

**DAG** Directed Acyclic Graph. 40

**DNA** Deoxyribonucleic Acid. 15

**EPU** Extended Polya Urn. 126, 127

**ER** Erdős–Rényi. 12, 21, 27, 28, 88, 114, 118, 119, 122

**F81** Felsenstein (1981). 15

**GTR** General Time-Reversible. 15, 16, 21, 22, 28, 50, 52, 55, 89, 90, 121, 130–132, 134, 142,  
153

**H<sup>+</sup>I** HIV-Positive Individual. 69–71, 84

**HIV** Human Immunodeficiency Virus. 2–4, 6, 8–10, 14, 17–20, 22, 28, 31, 69–72, 75, 80, 85, 87–89, 91, 116, 153

**HKY85** Hasegawa, Kishino, and Yano (1985). 54, 145

**HMM** Hidden Markov Model. 13, 21, 51, 52, 113

**ITS** Intelligent Tutor System. 106, 107, 109

**JC69** Jukes and Cantor (1969). 15, 24, 115, 122

**JSD** Jensen–Shannon Divergence. xiv, 24, 115, 122

**K80** Kimura (1980). 15, 54, 145

**KS** Kirkpatrick and Slatkin (1993). 61

**LANL** Los Alamos National Laboratory. 22, 30, 116

**LTT** Lineages Through Time. 97–99, 113, 114

**MAIT** Massive Adaptive Interactive Text. 5, 102–106, 108, 109, 111

**MCMC** Markov Chain Monte Carlo. 65

**ML** Maximum-Likelihood. 20

**MOOC** Massive Open Online Course. 5, 102–104, 108, 109

**MS** Matching Split. 51, 53, 135

**MSA** Multiple Sequence Alignment. 13, 20, 21, 52, 89

**NGS** Next Generation Sequencing. 102

**PDF** Probability Density Function. 91

**PIRC** Primary Infection Resource Consortium. 86, 89

**POSET** Partially Ordered Set. 41, 43

**PrEP** Pre-Exposure Prophylaxis. 30, 31, 70

**QALY** Quality-Adjusted Life-Year. 87

**RF** Robinson–Foulds. 28, 51, 53, 54, 119, 122, 135

**RNA** Ribonucleic Acid. 37, 85, 108

**SH<sup>+</sup>I** Sampled HIV-Positive Individual. 70–72, 74, 76, 77, 79, 80, 83–85, 89, 150, 151

**SINE** Short Interspersed Nuclear Element. 37, 63

**SSA** Simple Sampling Algorithm. 49

**tMRCA** Time of the Most Recent Common Ancestor. 20

**TN93** Tamura and Nei (1993). 15, 22, 80, 81, 90, 91

**UNAIDS** the Joint United Nations Programme on HIV/AIDS. 27

**WES** Whole Exome Sequencing. 108

**WGS** Whole Genome Sequencing. 108

**WS** Watts–Strogatz. 12, 21, 27, 114, 118, 119, 122



## LIST OF FIGURES

Figure 1.1:	FAVITES Workflow . . . . .	11
Figure 1.2:	Epidemiological Model . . . . .	19
Figure 1.3:	Real vs. Simulated Phylogenies . . . . .	26
Figure 1.4:	Sensitivity Analysis of Epidemiological Outcomes . . . . .	27
Figure 1.5:	Transmission Clustering Effectiveness . . . . .	29
Figure 2.1:	Dual-Birth Model . . . . .	40
Figure 2.2:	Theoretical Expectations . . . . .	46
Figure 2.3:	Tree Inference Error . . . . .	54
Figure 2.4:	Parameter Estimation Accuracy . . . . .	56
Figure 2.5:	Model Violations . . . . .	58
Figure 3.1:	ProACT Diagram . . . . .	73
Figure 3.2:	Adjusted ProACT Performance on Simulated Datasets . . . . .	77
Figure 3.3:	Raw ProACT Performance vs. Number of Individuals . . . . .	78
Figure 3.4:	Kendall’s Tau-b Test $p$ -Values (Step Functions) . . . . .	80
Figure 3.5:	Kendall’s Tau-b Test $p$ -Values (Sigmoid Functions) . . . . .	83
Figure 4.1:	Runtime Comparison . . . . .	96
Figure 4.2:	Lineages Through Time . . . . .	98
Figure 5.1:	Bloom’s Taxonomy . . . . .	105
Figure 5.2:	Example Code Challenge . . . . .	107
Figure 5.3:	Learner Demographics . . . . .	110
Figure A.1:	Seed Tree . . . . .	114
Figure A.2:	Average Branch Length vs. Expected Treatment Initiation Time . . . . .	114
Figure A.3:	Branch Length and Pairwise Distance Distributions . . . . .	115
Figure A.4:	Real vs. Simulated Viral Phylogenies . . . . .	116
Figure A.5:	Epidemiological Model States vs. Time . . . . .	117
Figure A.6:	Number of Infected Individuals vs. Expected Treatment Initiation Time . . . . .	118
Figure A.7:	Treated to Untreated Ratio vs. Expected Treatment Initiation Time . . . . .	119
Figure A.8:	Topology Error and Short Branches vs. Expected Treatment Initiation Time . . . . .	119
Figure B.1:	Probability Distributions on Ranked Tree Shapes . . . . .	138
Figure B.2:	Estimated $r$ vs. Cherry Fraction . . . . .	139
Figure B.3:	Histograms of Bitscores . . . . .	140
Figure B.4:	Bitscore Thresholds . . . . .	141
Figure B.5:	PASTA Alignment Sequence Lengths . . . . .	141
Figure B.6:	Human <i>Alu</i> Alignment and Phylogeny . . . . .	142
Figure B.7:	Human <i>Alu</i> Phylogeny Branch Support . . . . .	143
Figure B.8:	Matching Split Distance . . . . .	143

Figure B.9: Robinson–Foulds Distance . . . . .	144
Figure B.10: Branch Length Summary Statistics . . . . .	144
Figure B.11: Parameter Estimation Accuracy . . . . .	145
Figure B.12: True and Estimated Cherry Fraction . . . . .	145
Figure B.13: Molecular Clock on the <i>Alu</i> Phylogeny . . . . .	146
Figure C.1: ProACT Diagram . . . . .	149
Figure C.2: Number of Transmissions vs. Incident Branch Length . . . . .	149
Figure C.3: Raw ProACT Performance on Simulated Datasets . . . . .	150
Figure C.4: Optimal and Expected Raw Performance on Simulated Datasets . . . . .	151
Figure C.5: Genetic Linkage Score Functions . . . . .	152
Figure C.6: Kendall’s Tau-b Test vs. First ProACT Ordering . . . . .	152
Figure C.7: Proportion of Individuals in Infected States vs. Time . . . . .	154

## LIST OF TABLES

Table 1.1: Simulation Parameters . . . . .	21
Table 2.1: Experiments . . . . .	50
Table 2.2: Results on <i>Alu</i> . . . . .	59
Table 3.1: Varied Simulation Parameters . . . . .	75
Table 3.2: Kendall’s Tau-b Test $p$ -Values (Step Functions) . . . . .	82
Table A.1: Comparison with Existing Simulation Tools . . . . .	113
Table A.2: Post-Validation Tools . . . . .	113
Table A.3: Helper Scripts . . . . .	120
Table A.4: Simulation Parameters: Epidemiological Model . . . . .	121
Table A.5: Simulation Parameters: Evolutionary Model . . . . .	121
Table A.6: Real vs. Simulated Jensen–Shannon Divergence (JSD) . . . . .	122
Table A.7: Summary of Simulation Results . . . . .	122
Table C.1: Kendall’s Tau-b Test $p$ -Values (Sigmoid Functions) . . . . .	148
Table C.2: Full Simulation Parameters . . . . .	153

## ACKNOWLEDGEMENTS

I would like to thank all of the people who have helped me throughout my Ph.D. My colleagues in the Bioinformatics and Systems Biology Graduate Program and in the Computer Science and Engineering department have been great to work with, especially Jens Luebeck, Margaret Donovan, Bill Greenwald, Ben Pullman, and Argus Athanas. I enjoyed having the support of such excellent lab mates as Erfan Sayyari, Metin Balaban, Uyen Mai, Maryam Rabiee, Sina Malekian, and Chao Zhang. I would also like to thank Prof. Joel Wertheim, Manon Ragonnet-Cronin, Prof. Davey Smith, and Prof. Sanjay Mehta for teaching so much about the molecular biology and epidemiology of HIV. I am truly blessed to have been able to work with such amazing people.

I would not be where I am today (both figuratively and literally) without the mentorship of Prof. Vineet Bafna. I met him early on in my undergraduate career, and he has been the source of sage wisdom throughout my entire academic career thus far. It was largely because of him that I decided to continue to pursue graduate education in Bioinformatics, and I am thankful that his guidance led me down this path.

I am grateful of Prof. Pavel Pevzner for taking me under his wing at a time when I was somewhat lost regarding my career aspirations. Growing up, I always thought I wanted to become a medical doctor, but I always had a strong passion for technology. It was under his mentorship that I recognized the possibility to merge these two seemingly unrelated fields, and I was able to see the biomedical applications of Computer Science. He also introduced me to the field of Bioinformatics Education, and it was because of his guidance that I realized my passion for teaching, and I am honored he took a chance on me. He is truly a visionary, and I am happy to have been able to work with him.

When starting a Ph.D. program, one of the most stressful thoughts is “Will I find a good mentor?”, and I was extremely fortunate to have found Prof. Siavash Mirarab. I cannot thank him enough for all the time and effort he has taken to make my Ph.D. experience productive and

intellectually stimulating. From the long sessions in which we'd be working on an algorithmic problem together, to the hours he would spend helping me revise my manuscripts, the effort he took to mentor me has instilled within me far more than I could have imagined. I aspire to be as brilliant of a scientist as he one day, and I hope to be as great of a mentor as he was to me.

Chapter 1, in full, is a reprint of the material as it appears in “FAVITES: Simultaneous Simulation of Transmission Networks, Phylogenetic Trees, and Sequences” (2018). Moshiri, Niema; Ragonnet-Cronin, Manon; Wertheim, Joel; Mirarab, Siavash, *Bioinformatics*, bty921. The dissertation author was the primary investigator and first author of this paper.

Chapter 2, in full, is a reprint of the material as it appears in “A Two-State Model of Tree Evolution and its Applications to *Alu* Retrotransposition” (2017). Moshiri, Niema; Mirarab, Siavash, *Systematic Biology*, 67(3), 475-489. The dissertation author was the primary investigator and first author of this paper.

Chapter 3, in full, has been submitted for publication of the material as it may appear in “ProACT: Prioritization Using Ancestral Edge Lengths” (2019) Moshiri, Niema; Smith, Davey; Mirarab, Siavash, *Molecular Biology and Evolution*. The dissertation author was the primary investigator and first author of this paper.

Chapter 4, in full, has been submitted for publication of the material as it may appear in “TreeSwift: A Massively Scalable Python Tree Package” (2019) Moshiri, Niema, *SoftwareX*. The dissertation author was the primary investigator and sole author of this paper.

## VITA

2015	B. S. in Bioengineering: Bioinformatics, University of California, San Diego
2015-2016	Teaching Assistant, University of California, San Diego
2017	Associate Instructor, University of California, San Diego
2015-2019	Research Assistant, University of California, San Diego
2019	Ph. D. in Bioinformatics and Systems Biology, University of California, San Diego

## PUBLICATIONS

**Niema Moshiri** (2019). “Why Do I Look Like My Mom and Dad? Plants and Animals Inherit Traits From Parents,” *STEMTaught*

Metin Balaban, **Niema Moshiri**, Uyen Mai, and Siavash Mirarab (2019). “TreeCluster: Clustering Biological Sequences using Phylogenetic Trees,” *GLBIO 2019*, doi: 10.1101/591388

Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, **Niema Moshiri**, Mai H. Nguyen, Sara Brin Rosenthal, Fernando PÁlrez, and Peter W. Rose (2019). “Ten Simple Rules for Reproducible Research in Jupyter Notebooks,” *PLOS Computational Biology* (*in press*), arXiv: 1810.08055

**Niema Moshiri** and Liz Izhikevich (2018). *Design and Analysis of Data Structures*, Amazon KDP, ISBN: 1981017232

**Niema Moshiri**, Manon Ragonnet-Cronin, Joel O. Wertheim, and Siavash Mirarab (2018). “FAVITES: simultaneous simulation of transmission networks, phylogenetic trees, and sequences,” *Bioinformatics*, doi: 10.1093/bioinformatics/bty921

**Niema Moshiri** (2018). “TreeSwift: a massively scalable Python tree package,” *bioRxiv*, doi: 10.1101/325522

**Niema Moshiri**, Liz Izhikevich, and Christine Alvarado (2018). “Data Structures: An Active Learning Approach,” *edX*

**Niema Moshiri**, Phillip Compeau, and Pavel Pevzner (2017). “Analyze Your Genome!,” *edX*

Phillip Compeau, **Niema Moshiri**, and Pavel Pevzner (2017). “Introduction to Genomic Data Science,” *edX*

**Niema Moshiri** and Siavash Mirarab (2017). “A Two-State Model of Tree Evolution and Its Applications to Alu Retrotransposition,” *Systematic Biology*, 67(3), 475-489. doi: 10.1093/sysbio/syx088

ABSTRACT OF THE DISSERTATION

**Mathematical Modeling of Viral Evolution and Epidemiology**

by

Alexander Niema Moshiri

Doctor of Philosophy in Bioinformatics and Systems Biology

University of California San Diego, 2019

Professor Siavash Mirarab, Chair  
Professor Pavel A. Pevzner, Co-Chair

Phylogenetic trees can be used to study the evolution of any sequence that evolves, including viruses. In a viral epidemic, the history of transmission events defines constraints on the evolutionary history of the viral population. The spread of many viruses is driven by social and sexual networks, and because of the relationship between their evolutionary and transmission histories, phylogenetic inference from viral sequences can be used to improve the inference of patterns of the epidemic, which in turn may be able to enhance epidemiological intervention. The simultaneous simulation of viral transmission networks, phylogenetic trees, and sequences can provide a method to observe the effects of virus model parameters on the

epidemic as well as to study the accuracies and errors of transmission inference tools, but the success of such simulations relies on the existence of appropriate models. Further, the development of massively-scalable tools to analyze ultra-large datasets of viral sequences can aid epidemiologists in the real-time surveillance of the spread of disease. To enable viral epidemic simulation analyses, I developed FAVITES: a novel framework to simulate viral transmission networks, phylogenetic trees, and sequences, and I used FAVITES to study the effects of model parameters on epidemic outcomes. In an effort to better capture the unbalanced topologies commonly observed in retroviral phylogenies, I developed a novel evolutionary model (dual-birth), derived probabilistic distributions and theoretical expectations of trees sampled under the model, developed an approach to estimate model parameters given real data, and used the model to analyze *Alu* retrotransposons in the human genome. In order to potentially aid public health officials, I developed a scalable and non-parametric phylogenetic method of viral transmission risk prioritization, which I evaluated against current best-practice methods via simulation and real data. Lastly, I contributed to Bioinformatics education by developing multiple publicly-accessible adaptive online interactive texts.



# Introduction

Although they are typically associated with the study of the evolution of species, phylogenetic trees can be used to study the evolution of any sequence that evolves. For example, phylogenetic methods have been used to study the evolution of multicopy gene families [1], cancer genomes [2, 3], antibodies [4, 5, 6], segmental duplicates [7, 8], and transposable genomic elements [9, 10], which are all entities that evolve *within* the genome of a single species. Further, they can be used to study the evolution of viruses, both within and across hosts [11, 12, 13]. In the case of viruses, the history of transmission events constrains the evolutionary history, such as imposing a bottleneck at the time of each transmission [14].

The spread of many infectious diseases is driven by social and sexual networks [15], and reconstruction of their transmission histories from molecular data can greatly enhance intervention. For example, network-based statistics for measuring Human Immunodeficiency Virus (HIV) treatment effects can yield increased statistical power [16]; the analysis of the growth of HIV infection clusters can yield actionable epidemiological information for disease treatment and prevention [17, 18]; transmission-aware models can be used to infer rates of HIV evolution [13]. The ability to infer properties and patterns of the transmission history of a viral epidemic allows public health officials to intervene and attempt to prevent the spread of the virus. In the case of HIV, patients who adhere to Antiretroviral Therapy (ART) can become “virally suppressed,” meaning the virus is kept at bay, resulting in slower progression of the HIV disease as well as a significant reduction in transmission risk [19]. Thus, the ability to predict which individuals are most at-risk of transmitting the virus would provide public health officials actionable information: they can take measures to ensure high-risk individuals are able to continuously adhere to ART.

The ability to infer and reconstruct properties of a transmission network has been researched extensively in recent years [20, 21], and many tools exist that attempt to use molecular data to try to perform this inference [22]. For example, PhyloPart [23], Cluster Picker [24], and TreeCluster [25] infer transmission clusters using phylogenies inferred from viral sequences. HIV-TRACE, on the other hand, infers transmission clusters directly from sequences [26]. While

these tools have been used to analyze real datasets [27], the accuracies, errors, and limitations of these methods are still poorly understood.

By utilizing models of social contact networks, viral transmission, tree evolution, and sequence evolution together, epidemiologists can define complex probabilistic distributions composed of sub-models. These complex distributions can be sampled to simulate data representative of a virus of interest as it spreads through a population of interest, and the resulting data can be used to evaluate the accuracies of transmission network inference methods as well as to study trends and patterns of an epidemic as a function of the various model parameters to gain insights into the mechanisms driving the epidemic of interest [28]. However, many existing tools to perform such epidemic simulations have model assumptions that the user cannot relax or change. In Chapter 1, I will discuss FAVITES: a novel epidemic simulation framework I developed that provides flexibility in terms of the model assumptions about the epidemic, allowing the user to control the generative model in minute detail. The framework is defined by a series of interactions of abstract modules, and each *implementation* of a module defines the model assumptions. Thus, users are free to select whichever module implementations (and thus model assumptions) that best fit their epidemic of interest. In addition to presenting the tool, I will describe in detail a simulation experiment designed to emulate the San Diego HIV epidemic between 2005 and 2014, and I will use the simulated data to compare and contrast existing transmission clustering methods.

Of course, the ability to simulate an epidemic depends entirely on the existence of statistical models that appropriately describe the processes of the epidemic of interest. Models of tree evolution describe probability distributions over the space of tree shapes [29, 30], which can be used as the prior distribution in a Bayesian inference [31, 32, 33], to generate null distributions describing certain neutral processes [34, 35, 36], or to infer evolutionary parameters inherently of interest to the biologist [37]. Similarly, generalized epidemic models describe probability distributions over the space of transmission networks [38], and simulations that sample the

distributions defined by these stochastic models allows epidemiologists to study infection patterns of disease epidemics [39]. Further, network models describe probability distributions over graphs, and they can be used to capture features of networks of interactions (e.g. social interactions between humans) [40, 41, 42, 43]. Lastly, models of sequence evolution describe the mutation of a sequence over time [44, 45, 46, 47, 48], and they can be used to simulate the evolution of a sequence down a phylogeny [49] as well as to infer the evolutionary history of a set of sequences [50]. In the case of retroviruses and retrotransposons, which replicate via reverse transcription [51] and may undergo significant selection pressure [52], a neutral model of tree evolution like the Yule [29] or Coalescent [53] may not be appropriate. In Chapter 2, I will discuss the dual-birth model, a novel model of tree evolution that is motivated by the retrotransposition of *Alu* elements in the human genome [54]. I will derive various probabilistic distributions and theoretical expectations of trees sampled under the model, and I will then present two approaches for estimating model parameters from a given phylogeny. I will then present the results of an analysis of close to one million *Alu* sequences from the human genome in which I infer the dual-birth model parameters and present an estimate of the number of active *Alu* elements, a topic of much debate [55, 56, 57].

The goal of many transmission clustering analyses is to learn about the dynamics of a virus through a given population, often to try to predict which sub-populations may be spreading the virus more rapidly [16, 20, 58]. However, transmission clustering is essentially a way of summarizing the relationships between the sampled viral sequences, and instead of performing predictions and inferences on these summaries, what if we were able to infer properties of interest directly from the evolutionary relationships of the viruses? In Chapter 3, I investigate a single specific question: Given a set of viral sequences sampled from people living with HIV, can I predict which individuals are most at-risk of transmitting the virus in the future? In an attempt to address this question, I present ProACT, a tool that attempts to prioritize people living with HIV based on risk of future transmissions. ProACT depends only on the viral phylogeny and does not

require any demographic information from the patients, meaning it is not sensitive to error-prone survey data, and most importantly, it is less prone to bias.

A primary focus of mine is the ability to execute phylogenetic methods like ProACT in a massively-scalable fashion. The scalability of a computational tool is primarily dependant on two things: (1) the theoretical time complexity of the algorithm, and (2) the efficiency of the implementation of the algorithm. While developing FAVITES and ProACT, I found that, although the phylogenetic algorithms I designed were quite fast in theory, my implementations using existing tree-manipulation packages were much slower than I anticipated due to significant overhead in loading and initializing my ultra-large phylogenies. In Chapter 4, I will present TreeSwift, a new Python package for traversing and manipulating trees. I will describe its implementation design, demonstrate some of its features, and compare its execution time for various common tree algorithms against existing packages.

As can be seen, as the cost of sequencing decreases, the amount of viral sequence data available to researchers is growing rapidly, and as a result, the field of Epidemiology is becoming increasingly dependent on scalable computational methods. However, many researchers in the fields of Epidemiology and Molecular Biology have never received formal education in computation. In recent years, computational courses have started appearing in undergraduate Biology major curricula, and while this will provide computational skills to the next generation of biomedical and epidemiological researchers, it does not benefit the current generation of professionals. In Chapter 5, I will present my contributions to Bioinformatics education in the form of developing novel Massive Open Online Courses (MOOCs) and Massive Adaptive Interactive Texts (MAITs), and I will discuss the pedagogical philosophy I employed in developing my learning materials.

In summary, I show that modern studies in viral epidemiology require the ability to perform simulation experiments that can appropriately capture the virus and population of interest, which thus requires tools to run such simulations efficiently as well as statistical models that make

realistic assumptions. I also show that the ability to infer actionable epidemiological information from molecular data is an open problem. In this dissertation, I present a novel framework for epidemic simulations, provide a novel model of phylogenetic evolution (and derive probabilistic distributions and theoretical expectations of trees sampled under the model), demonstrate the effectiveness of a novel phylogenetic tool for prioritizing people living with HIV based on their risk of future transmissions, and introduce a novel package for performing tree traversals and manipulations efficiently on ultra-large phylogenies. I also discuss my contributions to Bioinformatics education.

# **Chapter 1**

## **FAVITES: Simultaneous Simulation of Transmission Networks, Phylogenetic Trees, and Sequences**

*Motivation* — The ability to simulate epidemics as a function of model parameters allows insights that are unobtainable from real datasets. Further, reconstructing transmission networks for fast-evolving viruses like HIV may have the potential to greatly enhance epidemic intervention, but transmission network reconstruction methods have been inadequately studied, largely because it is difficult to obtain “truth” sets on which to test them and properly measure their performance.

*Results* — We introduce FAVITES, a robust framework for simulating realistic datasets for epidemics that are caused by fast-evolving pathogens like HIV. FAVITES creates a generative model to produce contact networks, transmission networks, phylogenetic trees, and sequence datasets, and to add error to the data. FAVITES is designed to be extensible by dividing the generative model into modules, each of which is expressed as a fixed Application Program Interface (API) that can be implemented using various models. We use FAVITES to simulate HIV datasets and study the realism of the simulated datasets. We then use the simulated data to study the impact of the increased treatment efforts on epidemiological outcomes. We also study two transmission network reconstruction methods and their effectiveness in detecting fast-growing clusters.

*Availability and implementation* — FAVITES is available at <https://github.com/niemasd/FAVITES>, and a Docker image can be found on DockerHub (<https://hub.docker.com/r/niemasd/favorites>).

## 1.1 Introduction

The spread of many infectious diseases is driven by social and sexual networks [59], and reconstructing their transmission histories from molecular data may be able to enhance intervention. For example, network-based statistics for measuring the effects of ART in HIV can yield increased statistical power [16]; the analysis of the growth of HIV infection clusters can yield actionable epidemiological information for disease control [60]; transmission-aware models



can be used to infer HIV evolutionary rates [13].

A series of events in which an infected individual infects another individual can be shown as a *transmission network*, which itself is a subset of a *contact network*, a graph in which nodes represent individuals and edges represent contacts (e.g. sexual) between pairs of individuals. If the pathogens of the infected individuals are sequenced, which is the standard of HIV care in many developed countries, one can attempt to reconstruct the transmission network (or its main features) using molecular data. Some viruses, such as HIV, evolve quickly, and the phylogenetic relationships between viruses are reflective of transmission histories [61], albeit imperfectly [62, 63, 64].

Recently, multiple methods have been developed to infer properties of transmission networks from molecular data [23, 24, 26]. Efforts have been made to characterize and understand the promise and limitations of these methods: it is suggested that, when combined with clinical and epidemiological data, these methods can provide critical information about drug resistance, associations between sociodemographic characteristics, viral spread within populations, and the time scales over which viral epidemics occur [65]. More recently, these methods have become widely used at both local [27] and global scale [66]. Nevertheless, several questions remain to be fully answered regarding the performance of these methods. It is not always clear which method/setting combination performs best for a specific downstream use-case or for specific epidemiological conditions. More broadly, the effectiveness of these methods in helping achieve public health goals is the subject of ongoing clinical and theoretical research.

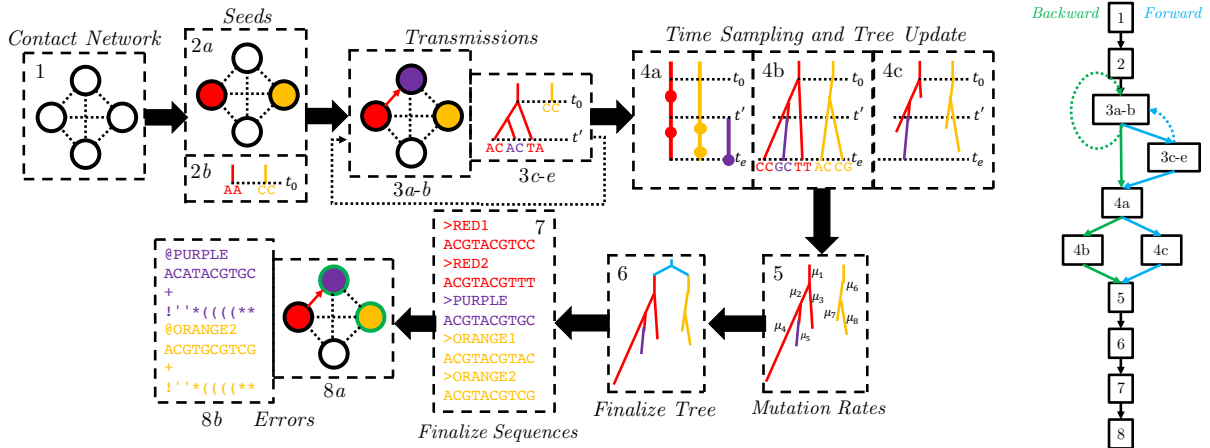
Accuracy of transmission networks is difficult to assess because the true order of transmissions is not known. Moreover, predicting the impact of parameters of interest (e.g. rate of treatment) on the epidemiological outcomes is difficult. In simulations, in contrast, the ground truth is known and parameters can be easily controlled. The simulation of transmission networks needs to combine models of social network, transmission, evolution, and ideally sampling biases and errors [67].

We introduce FAVITES (FrAmework for VIral Transmission and Evolution Simulation), which can simulate numerous models of contact networks, viral transmission, phylogenetic and sequence evolution, data (sub)sampling, and real-world data perturbations, and which was built to be flexible such that users can seamlessly plug in statistical models at every step of the simulation process. Previous attempts to create an epidemic simulation tool include epinet [68], TreeSim [69], outbreaker [70], seedy [71], and PANGEA.HIV.sim [28]. A detailed comparison of FAVITES with these tools can be found in Table A.1. One key distinction is that FAVITES simulates the full end-to-end epidemic dataset (social contact network, transmission history, incomplete sampling, viral phylogeny, error-free sequences, and real-world sequencing imperfections), whereas each existing tool simulates only a subset of these steps. Another key distinction is that FAVITES allows the user to choose among several models at each step of the simulation, whereas the existing tools are restricted to specific models. After describing the FAVITES framework, we compare its output to real data on a series of experiments, study the properties of HIV epidemics as functions of various model and parameter choices, and finally perform simulation experiments to study two transmission network reconstruction methods.

## 1.2 Materials and Methods

### 1.2.1 FAVITES Simulation Process

FAVITES provides a workflow for the simulation of viral transmission networks, phylogenetic trees, and sequence data (Fig. 1.1). It breaks the simulation process into a series of interactions between abstract modules, and users can select the module implementations appropriate to their specific context. In the statistical sense, the end-to-end process creates a complex composite generative model, each module is a template for a sub-model of a larger model, and different implementations of each module correspond to different statistical sub-models. Thus, the FAVITES workflow does not explicitly make model choices: each module *implementation*



**Figure 1.1:** FAVITES workflow. (1) The contact network is generated (nodes: individuals; edges: contacts). (2) *Seed* individuals who are infected at time 0 are selected (2a), and a viral sequence is chosen for each (2b). (3) The epidemic yields a series of transmission events in which the time of the next transmission is chosen (3a), the source and target individuals are chosen (3b), the viral phylogeny in the source node is evolved to the transmission time (3c), viral sequences in the source node are evolved to the transmission time (3d), and a viral lineage is chosen to be transmitted from source to destination (3e). Step (3) repeats until the end criterion is met. Step 3c–e are optional, as tree and sequence generation can be delayed to later steps. (4) Infected individuals are sampled such that viral sequencing times are chosen for each infected individual (4a), viral phylogenies (one per seed) are evolved to the end time of the simulation (4b), and viral phylogenies (one per seed) are pruned to reflect the viral sequencing times selected (4c). (5) Mutation rates are introduced along the branches of the viral phylogenies and the tree is scaled to the unit of expected mutations. (6) The seed trees are merged using a seed tree (cyan). (7) Viral sequences obtained from each infected individual are finalized. (8) Real-world errors are introduced on the error-free data, such as subsampling of the sequenced individuals (marked as green) (8a) and the introduction of sequencing errors (8b). The workflows of a typical forward (blue) and backward (green) simulation are shown as well.

makes those choices. The model for a FAVITES execution is defined by the set of module implementations chosen by the user.

FAVITES defines APIs for each module and lets implementation decide how to achieve the goal of the module. The APIs allow various forms of interaction between modules, which enable sub-models that are described as conditional distributions (via dependence on preceding steps) or as joint distributions (via joint implementation). Module implementations can simply wrap around existing tools, allowing for significant code reuse. The available implementations for each step are continuously updated; the full documentation of these implementations can be

found online.

Simulations start at time zero and continue until a user-specified stopping criterion is met. Error-free and error-prone transmission networks, phylogenetic trees, and sequences are output at the end. FAVITES has eight steps (Fig. 1.1) detailed below. Depending on the specific implementations, some of the steps may not be needed (we mark these with an asterisk), especially when the phylogeny is simulated backward in time. Also note that steps and modules are not the same; a module may be used in several steps and a step may require multiple modules.

### **Step 1: Contact Network**

The *ContactNetworkGenerator* module generates a contact network; vertices represent individuals, and edges represent contacts between them that can lead to disease transmission (e.g. sexual). Graphs can be created stochastically using existing models [72], including those that capture properties of real social networks [40, 42, 73] and those that include communities [41, 74]. For example, the Erdős–Rényi (ER) model [75] generates graphs with randomly-placed edges, the Random Partition model [74] generates communities, the Barabási–Albert (BA) model [42] generates scale-free networks whose degree distributions follow power-law (suitable for social and sexual contact networks), the Caveman model [41] and its variations [74] generate small-world networks, the Watts–Strogatz (WS) model [40] generates small-world networks with short average path lengths, and Complete graphs connect all pairs of individuals (suitable for some communicable diseases). We currently have many models implemented by wrapping around the NetworkX package [76]. In addition, a user-specified network can be used.

### **Step 2: Seeds**

The transmission network is initialized in two steps. *a*) The *SeedSelection* module chooses the “seed” nodes: individuals who are infected at time zero of the simulation. *b*\*) For each selected seed node, the *SeedSequence* module can generate an initial viral sequence.

Seed selection has many implemented models, including uniform random selection, degree-weighted random selection, and models that place seeds in close proximity. Seed sequences can be user-specified or randomly sampled from probabilistic distributions. To enable seed sequences that emulate the virus of interest, we implement a model that uses HMMER [77] to sample each seed sequence from a profile Hidden Markov Model (HMM) specific to the virus of interest. Profile HMMs are appropriate for sampling random sequences that are intended to resemble real sequences because they define a probabilistic distribution over the space of sequences, they can be flexible to insertions and deletions, and they can be sampled in a computationally efficient manner. We provide a set of such prebuilt profile HMMs constructed from Multiple Sequence Alignments (MSAs) of viral sequences.

When multiple seeds are chosen, we need to model their phylogenetic relationship as well. Thus, we also have a model that samples a *single* sequence from a viral profile HMM using HMMER, simulates a *seed tree* with a single leaf per seed individual (e.g. using Kingman coalescent or birth-death models using DendroPy [78]), and then evolves the viral sequence down the tree to generate seed sequences using Seq-Gen [49].

### **Step 3: Transmissions**

An iterative series of transmission events occurs under a transmission model until the *EndCriteria* module triggers termination (e.g. after a user-specified time or a user-specified number of transmission events). Each transmission event has five components.

*a)* The *TransmissionTimeSample* module chooses the time at which the next transmission event will occur and advances the *current* time accordingly, and *b)* the *TransmissionNodeSample* module chooses a source node and target node to be involved in the next transmission event. These two modules are often jointly implemented. Some of the current implementations use simple models such as drawing transmission times from an exponential distribution and selecting nodes uniformly at random. Others are more realistic and use Markov processes in which individuals

start in some state (e.g. Susceptible) and transition between states of the model (e.g. Infected) over time. These Markov models are defined by two sets of transition rates: *nodal* and *edge-based*. Nodal transition rates are rates that are independent of interactions with neighbors (e.g. the transition rate from Infected to Recovered), whereas edge-based transition rates are the rate of transitioning from one state to another given that a single neighbor is in a given state (e.g. the transition rate from Susceptible to Infected given that a neighbor is Infected). The rate at which a specific node  $u$  transitions from state  $a$  to state  $b$  is the nodal transition rate from  $a$  to  $b$  plus the sum of the edge-based transition rate from  $a$  to  $b$  given neighbor  $v$ 's state for all neighbors  $v$ . We use GEMF [39] to implement many compartmental epidemiological models in this manner, including sophisticated HIV models like the Granich *et al.* (2009) model [79] and the HPTN 071 (PopART) model [80].

$c^*$ ) The *NodeEvolution* module evolves viral phylogenetic trees of the source node to the current time using stochastic models of tree evolution [81]. We use DendroPy [78] for birth-death and use our own implementation of dual-birth [10] and Yule.

$d^*$ ) If models of the tree evolution or transmission models are dependent on sequences, the *SequenceEvolution* module is invoked here to evolve all viral sequences in the source node to the current time. Otherwise, sequence evolution is delayed until Step 7 (we assume this scenario).

$e^*$ ) The *SourceSample* module chooses the viral lineage(s) in the source node to be transmitted.

#### **Step 4: Time Sampling and Tree Update**

The patient sampling (i.e., sequencing) events are determined and phylogenetic trees are updated accordingly. Three sub-steps are involved.

$a$ ) For each individual, the *NumTimeSample* module chooses the number of sequencing times (e.g. a fixed number or a number sampled from a Poisson distribution), the *TimeSample* module chooses the corresponding sequencing time(s) (e.g. by draws from uniform or truncated

Gaussian or Exponential distributions, or by sampling right before the first transition of a person to a treated state), and the *NumBranchSample* module chooses how many viral lineages will be sampled at each sequencing time (e.g. single). A given individual may not be sampled at all, thus simulating incomplete epidemiological sampling efforts.

$b^*/c^*$ ) The *NodeEvolution* module is called to simulate the phylogenetic trees *given sampling times*. This step can be used *instead of* Step 3c to evolve only lineages that are sampled, thereby reducing computational overhead. If the tree is simulated in Step 3c, it will be pruned here to only include lineages that are sampled.

### **Step 5: Mutation Rates**

To generate sequences, rates of evolution must be assumed and in this step, the *TreeUnit* module determines such rates. For example, it may use constant rates or may draw from a distribution (e.g. Gamma). Applying rates on the tree from Step 4 yields a tree with branch lengths in units of per-site expected number of mutations.

### **Step 6\*: Finalize Tree**

We now have a single tree per seed. Some implementations of *SeedSequence* also simulate a tree connecting seeds, so the roots of per-seed trees have a phylogenetic relationship. In this case, this step merges all phylogenetic trees into a single global tree by placing each individual tree's root at its corresponding leaf in the seed tree (Fig. 1.1).

### **Step 7: Finalize Sequences**

The *SequenceEvolution* module is called to simulate sequences on the final tree(s). Commonly-used models of Deoxyribonucleic Acid (DNA) evolution including General Time-Reversible (GTR) model [48], and its reductions such as Jukes and Cantor (1969) (JC69) [44], Kimura (1980) (K80) [45], Felsenstein (1981) (F81) [46], and Tamura and Nei (1993) (TN93) [47],

are currently available as implementations of *SequenceEvolution*. FAVITES also includes the GTR+ $\Gamma$  model, which incorporates rates-across-sites variation [82]. It also includes multiple codon-aware extensions of the GTR model, such as mechanistic [83] and empirical [84] codon models. These modules internally use Seq-Gen [49] and Pyvolve [85].

## Step 8: Errors

Error-free data are now at hand. Noise is introduced onto the complete error-free data in two ways.

*a*\*) The *NodeAvailability* module further subsamples the individuals to simulate lack of accessibility to certain datasets. Note that whether or not an individual is sampled is a function of two different modules: *NodeAvailability* and *NumTimeSample* (if *NumTimeSample* returned 0, the individual is not sampled). Conceptually, *NumTimeSample* can be used to model when people are sequenced, while *NodeAvailability* can be used to model patterns of data availability (e.g. sharing of data between clinics).

*b*) The *Sequencing* module simulates sequencing error on the simulated sequences. In addition to sequencing machine errors, this can incorporate other real-world sequencing issues, e.g. taking the consensus sequence of a sample and introducing of ambiguous characters. FAVITES currently uses existing tools to simulate Illumina, Roche 454, SOLiD, Ion Torrent, and Sanger sequencing [86, 87], including support for ambiguous characters.

## Backward-in-Time Simulation

Thus far, we have assumed that trees are evolved forward-in-time: they begin with a single root lineage, and as time progresses, lineages split. However, backward-in-time models of tree evolution (e.g. coalescent) begin with  $k$  leaves, and as time regresses, these lineages coalesce. In FAVITES, if a backward-in-time model of tree evolution is chosen, Steps 3c–e and 4c can be skipped, and the full backward simulation can be performed at once in Step 4b (Fig. 1.1).



We use VirusTreeSimulator [28] for coalescent models with constant, exponentially-growing, or logistically-growing population size.

### **Sequence-Dependent Transmissions**

Steps 3c–e are required only if the choice of transmission events after time  $t$  depends on the past phylogeny or sequences up to time  $t$ . If the choice of future transmission recipients/donors and transmission times are agnostic to past phylogenies and sequences, these steps can be skipped and the tasks are delayed to Steps 4b and 7. Note also that if sequences are simulated in Step 3d, a mutation rate needs to be assumed early. In this case, a joint implementation of the *TreeUnit* and *SequenceEvolution* modules must be used such that per-time mutation rates are chosen in Step 3d, and the same mutation rates are used to scale the tree in Step 5.

### **Model Validation**

We provide tools to validate FAVITES outputs, by comparing the simulation results against real data the user may have (e.g. networks, phylogenetic trees, or sequence data) using various summary statistics (Table A.2). In addition to validation scripts, we have several helper scripts to implement tasks that are likely common to downstream use of FAVITES output (Table A.3).

## **1.2.2 Experimental Setup**

We have performed a set of simulations using the FAVITES framework. In these studies, we compare the simulated data against real HIV datasets, study properties of the epidemic as a function of the parameters of the underlying generative models, and compare two transmission cluster inference tools when applied to sequence data generated by FAVITES. All datasets can be found at <https://gitlab.com/niemasd/favites-paper-final>.

## **The Simulation Model**

We selected a set of “base” simulation models and parameters and also performed experiments in which they were varied. For each parameter set, we ran 10 simulation replicates. The base simulation parameters were chosen to emulate HIV transmission in San Diego from 2005 to 2014 to the extent possible. In addition, to show the applicability of FAVITES to other settings, we also performed a simulation with parameters learned from the HIV epidemic in Uganda from 2005 to 2014. For both datasets, we estimate some parameters from real datasets while we rely on the literature where such data are not available. We first describe base parameters for San Diego and then present changing parameters and Uganda parameters (see Tables A.4 and A.5 for the full list of parameters).

## **Contact Network**

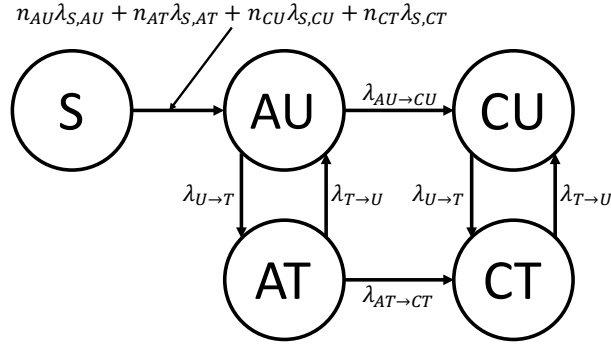
The contact network includes 100,000 individuals to approximate the at-risk community of San Diego. We set the base expected degree ( $\mathbb{E}_d$ ) to 4 edges (i.e., sexual partners over 10 years). This number is motivated by estimates from the literature (e.g.  $\approx 3$  in Wertheim *et al.*, 2017 [20] and 3–4 in Rosenberg *et al.*, 2011 [88]), and it is varied in the experiments. We chose the BA model as the base network model because it can generate power-law degree distributions [42], a property commonly assumed of sexual networks [89].

## **Seeds**

We chose 15,000 total infected seed individuals uniformly at random based on the estimate of total HIV cases in San Diego as of 2004 [90].

## **Epidemiological Model**

We model HIV transmission as a Markov chain epidemic model (see Section 1.2.1) with states Susceptible (S), Acute Untreated (AU), Acute Treated (AT), Chronic Untreated (CU), and



**Figure 1.2:** Epidemiological model of HIV transmission with states Susceptible (S), Acute Untreated (AU), Acute Treated (AT), Chronic Untreated (CU), and Chronic Treated (CT). The model is parameterized by the rates of infectiousness of AU ( $\lambda_{S,AU}$ ), AT ( $\lambda_{S,AT}$ ), CU ( $\lambda_{S,CU}$ ), CT ( $\lambda_{S,CT}$ ) individuals, and by the rate to transition from AU to CU ( $\lambda_{AU \rightarrow CU}$ ), the rate to transition from AT to CT ( $\lambda_{AT \rightarrow CT}$ ), the rate to start ART ( $\lambda_{U \rightarrow T}$ ), and the rate to stop ART ( $\lambda_{T \rightarrow U}$ ).

Chronic Treated (CT). All seed individuals start in AU, and transmissions occur with rates that depend for each individual on the number of neighbors it has in each state (Fig. 1.2). Note that this model is a simplification of the model used by Granich *et al.* (2009) [79].

We set  $\lambda_{AU \rightarrow CU}$  such that the expected time to transition from AU to CU is 6 weeks [91] and set  $\lambda_{AT \rightarrow CT}$  such that the expected time to transition from AT to CT is 12 weeks [92]. We set  $\lambda_{U \rightarrow T}$  such that the expected time to start ART is 1 year from initial infection [93], and we define  $\mathbb{E}_{ART} = 1/\lambda_{U \rightarrow T}$ . We set  $\lambda_{T \rightarrow U}$  such that the expected time to stop ART is 25 months from initial treatment [94]. For the rates of infection  $\lambda_{S,j}$  for  $j \in \{AU, CU, AT, CT\}$ , using the infectiousness of CU individuals as a baseline, we set the parameters such that AU individuals are 5 times as infectious [95] and CT individuals are not infectious (i.e., rate of 0). Cohen *et al.* (2011) found a 0.04 hazard ratio when comparing linked HIV transmissions between an early-therapy group and a late-therapy group [92], so we estimated AT individuals to be  $1/20$  the infectiousness of CU individuals. We then scaled these relative rates so that the total number of new cases over the span of the 10 years was roughly 6,000 [90], yielding  $\lambda_{S,AU} = 0.1125$ .

## Phylogeny

We estimate parameters related to phylogeny and sequences from real data. We used a MSA of 674 HIV-1 subtype B Polymerase (*pol*) sequences from San Diego [58] and a subset containing the 344 sequences that were obtained between 2005 and 2014. For both of these datasets, we inferred Maximum-Likelihood (ML) phylogenetic trees using the ModelFinder Plus feature [96] of IQ-TREE [97]. We then removed outgroups from the tree inferred from the full 674 sequence dataset and used LSD [98] to estimate the Time of the Most Recent Common Ancestor (tMRCA) and the per-year mutation rate distribution. The tMRCA was estimated at 1980. The mutation rate was estimated as 0.0012 with a standard deviation of roughly 0.0003, so to match these properties, we sampled mutation rates for each branch independently from a truncated Normal random variable from 0 to infinity with a location parameter of 0.0008 and a scale parameter of 0.0005 to scale branch lengths from years to expected number of per-site mutations.

In our simulations, a single viral lineage from each individual was sampled at the end time of the epidemic (10 years). The viral phylogeny in unit of time (years) was then sampled under a coalescent model with logistic viral population growth using the same approach as the the PANGEA-HIV methods comparison exercise, setting the initial population to 1, the per-year growth rate to 2.851904, and the time back from present at which the population is at half the carrying capacity ( $v.T50$ ) to -2 [28]. Each seed individual is the root of an independent viral phylogenetic tree, and these trees were merged by simulating a seed tree with one leaf per seed node under a non-homogeneous Yule model [99] scaled such that its height equals 25 years to match the 1980 estimate using SD. The rate function of the non-homogeneous Yule model was set to  $\lambda(t) = e^{-t^2} + 1$  to emulate short branches close to the base of the tree (see comparison to other functions in Fig. A.1).

## Sequence Data

We sampled a root sequence from a profile HMM generated from the San Diego MSA using HMMER [77]. We evolved it down the scaled viral phylogenetic tree under the GTR+ $\Gamma$  model using Seq-Gen [49] with parameters inferred by IQ-TREE (Table A.5).

## Varying Parameters

For San Diego, we explore four parameters (Table 1.1). For the contact network, in addition to the BA model, we used the ER [75] and WS [40] models. We also varied the expected degree ( $\mathbb{E}_d$ ) of individuals in the contact network between 2 and 16 (Table 1.1). For seed selection, we also used “Edge-Weighted,” where the probability that an individual is chosen is weighted by the individual’s degree. For each selection of contact network model,  $\mathbb{E}_d$ , and seed selection method, we study multiple rates of starting ART (expressed as  $\mathbb{E}_{ART}$ ). In our discussions, we focus on  $\mathbb{E}_{ART}$ , a factor that the public health departments can try to impact. Increased effort in testing at-risk populations can decrease the diagnosis time, and the increased diagnosis rate coupled with high standards of care can lead to faster ART initiation. Behavioral intervention could in principle also impact degree distribution, another factor that we vary, but the extent of the effectiveness of behavioral interventions is unclear [59].

**Table 1.1:** Simulation parameters (base parameters in bold)

<b>Parameter</b>	<b>Values</b>
Contact Network Model	<b>BA</b> , ER, WS
Expected Degree ( $\mathbb{E}_d$ )	2, <b>4</b> , 8, 16
Seed Selection	<b>Random</b> , Edge-Weighted
Mean Time to ART ( $\mathbb{E}_{ART}$ )	$\frac{1}{8}$ , $\frac{1}{4}$ , $\frac{1}{2}$ , <b>1</b> , 2, 4, 8 (years)

## Uganda Simulations

Our simulations with Uganda followed a similar approach to the base model used for San Diego but with different choices of parameters, motivated by Uganda. For inferring the reference phylogeny and mutation rates, we used a dataset of all 893 HIV-1 subtype D *pol* sequences in the Los Alamos National Laboratory (LANL) HIV Sequence Database that were sourced from Uganda and that were obtained between 2005 and 2014. All other Uganda parameters were motivated by McCreesh *et al.* (2017) [100], and the following are key differences from the San Diego simulation. The contact network had 10,000 total individuals (a regional epidemic), and 1,500 individuals were randomly selected to be seeds. For epidemiological parameters, we assumed the expected time to begin as well as stop ART to be 1 year [100]. A comprehensive list of simulation parameters can be found in Tables A.4 and A.5.

## Transmission Network Reconstruction Methods

We compare two HIV network inference tools: HIV-TRACE [26] and TreeCluster [25]. HIV-TRACE is a widely-used method [22, 20, 101] that clusters individuals such that, for all pairs of individuals  $u$  and  $v$ , if the TN93 distance is below the threshold (default 1.5%),  $u$  and  $v$  are connected by an edge; each connected component forms a cluster. When we ran HIV-TRACE, we skipped its alignment step because we did not simulate indels. TreeCluster clusters the leaves of a given tree such that the pairwise path length between any two leaves in the same cluster is below the threshold (default 4.5%), the members of a cluster define a full clade, and the number of clusters is minimized. Trees given to TreeCluster were inferred using FastTree 2 [102] under the GTR+ $\Gamma$  model. We used FastTree 2 because using IQ-TREE on these very large datasets (up to 80,000 leaves) was not feasible. TreeCluster is similar in idea to Cluster Picker [24], which uses sequence distances instead of tree distances (but also considers branch support). We study TreeCluster instead of Cluster Picker because of its improved speed. Our attempts to run PhyloPart [23] were unsuccessful due to running time.

## Measuring the Predictive Power of Clustering Methods

We now have two sets of clusters at the end of the simulation process (year 10): one produced by HIV-TRACE and one by TreeCluster. Let  $C^t$  denote the clustering resulting from removing all individuals infected after year  $t$  from a given final clustering  $C^{10}$ , let  $C_i^t$  denote a single  $i$ -th cluster in clustering  $C^t$ , and let  $g(C_i^t) = \frac{|C_i^t| - |C_i^{t-1}|}{\sqrt{|C_i^t|}}$  denote the growth rate of a given cluster  $C_i^t$  [103]. We then compute the average number of individuals who were infected between years 9 and 10 by the “top” 1,000 individuals (roughly 5% of the total infected population) who were infected at year 9, where we choose top individuals by sorting the clusters in  $C^9$  in descending order of  $g(C_i^9)$  (breaking ties randomly) and choosing 1,000 individuals in this sorting, breaking ties in a given cluster randomly if needed (e.g. for the last cluster needed to reach 1,000 individuals). As a baseline, we compute the average number of individuals who were infected between years 9 and 10 by *all* individuals, which is equivalent (in expectation) to a random selection of 1,000 individuals. Our metric, therefore, measures the risk of transmission from the selected 1,000 individuals. Our motivation for this metric is to capture whether monitoring cluster growth can help public health intervention efforts with limited resources in finding individuals with a higher risk of transmitting.

## 1.3 Results

### 1.3.1 Comparison to Real Phylogenies

To compare data simulated by FAVITES to real data, we use the aforementioned San Diego and Uganda phylogenies. Since the trees on real data are inferred trees (as opposed to true trees), we compare them to inferred trees on simulated data (built using FastTree 2 as running IQ-TREE on simulated data was not feasible). We randomly subsample the simulated dataset to match the number of sequences in the corresponding real dataset (344 for San Diego; 893 for

Uganda).

For San Diego, the mean patristic distance between sequences on inferred trees is respectively 0.087 and 0.089 for the real and base simulated datasets. The distributions of pairwise distances among inferred trees of real and simulated datasets have similar shapes, but distances from real data have higher variance (Fig. 1.3a). To quantify the divergence between the real and simulated distributions, we use the JSD, a number between 0 and 1 with 0 indicating a perfect match [104]. The JSD is only 0.023 for trees inferred from the San Diego base parameters (Table A.6). The Uganda simulations have a larger divergence (Fig. 1.3a) between real and simulated distributions (JSD: 0.082), with simulated data showing higher mean distances (means: 0.075 and 0.097). We observe similar patterns when we compute pairwise distances directly from sequences and apply phylogenetic correction using the JC69+ $\Gamma$  model (Table A.6; Fig. A.3). For all simulated datasets, the true trees have lower variance in pairwise distances compared to estimated trees; this is consistent with the stochasticity of sequence evolution and the added variance due to the inference uncertainty.

Our simulated trees, like real trees, include clusters of long terminal branches and short internal branches, especially close to the root (Fig. A.4). The branch length distributions are bimodal, with one peak close to 0 and another between 0.01 and 0.03 (Fig. 1.3b). However, the second mode for the real trees is larger than the second mode of real data; for example, for San Diego, the second peak is at 0.030 for real data and 0.023 for base simulated data. The JSD divergence between branch length distributions of real and simulated trees are 0.102 for San Diego (base) and 0.119 for Uganda. The distribution of branch lengths on true trees (as opposed to inferred trees) has a similar shape (Fig. 1.3b) but a shorter tail of long branches and a reduced JSD compared to real data (e.g. 0.044 for base San Diego; see Table A.6).



## Sensitivity to Parameters

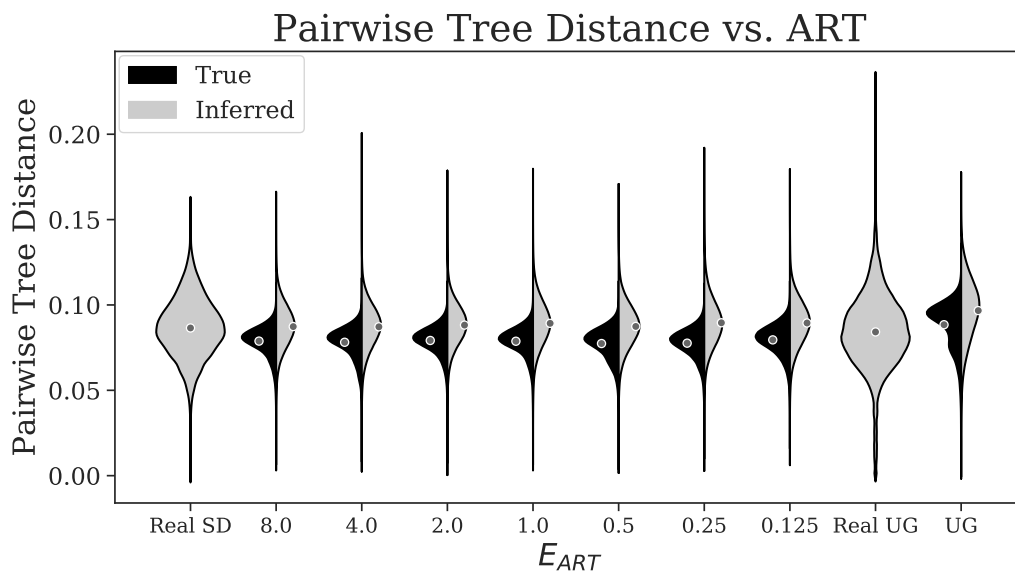
Even though mean branch lengths can change (between 0.0053 and 0.0080) as a result of changing  $\mathbb{E}_{ART}$  and  $\mathbb{E}_d$  (Fig. A.2), the overall distributions remain quite stable (Figs. 1.3b and A.3). Similarly, patristic distances are not sensitive to  $\mathbb{E}_{ART}$  (Fig. 1.3ab) nor to  $\mathbb{E}_d$  (Fig. A.3). In terms of branch lengths, the divergence from the real data changes only marginally as  $\mathbb{E}_d$  and  $\mathbb{E}_{ART}$  change (Table A.7). While the distributions are stable with respect to these epidemiological parameters, they are sensitive to others. For example, results are sensitive to the model of mutation rates. We draw mutation rates from a Truncated Normal distribution (fitted to real data) and obtain close matches to real data. However, other distributions (e.g. Exponential) yield significant deviation from real distributions (Fig. A.3). Because of these deviations, we have only used the truncated normal distributions for mutation rates everywhere.

### 1.3.2 Impact of Parameter Choices On the Epidemiology

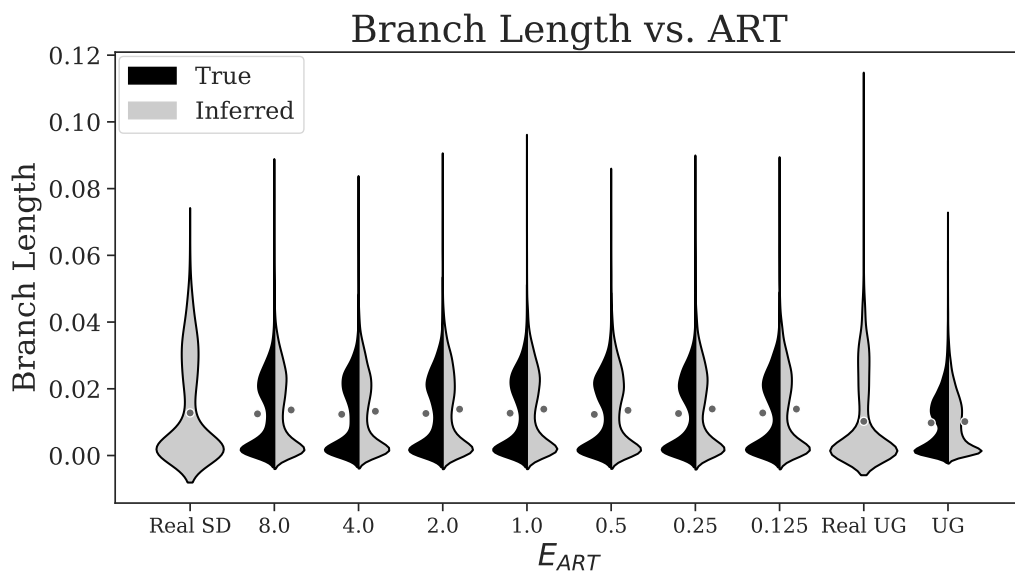
#### Infected Population

The number of infected individuals increases with time and the rate of growth is faster for larger  $\mathbb{E}_{ART}$  values (Fig. A.5). For all tested values of  $\mathbb{E}_{ART}$ , the number of infected individuals grows close to linearly (Pearson  $r \geq 0.966$ ), indicating that the large at-risk population has not saturated in the 10-year simulation period. As  $\mathbb{E}_{ART}$  decreases from 8 years to  $1/8$  years, the total number of infected individuals at the end of the simulation keeps decreasing (Fig. 1.4a). For example, with degree 4, the average final number of infected individuals in the 10 year period is 6686, 4134, and 1273 with  $\mathbb{E}_{ART}$  set to 1,  $1/2$ ,  $1/8$  year, respectively.

The model of contact network and the model of choosing the seed individuals have only marginal effects on these outcomes. Edge-weighting the seed selections yields a slightly higher (at most 12%) total number of infected individuals than the random selection (Table A.7). The BA model of contact network leads to a slightly higher infection count when compared to the

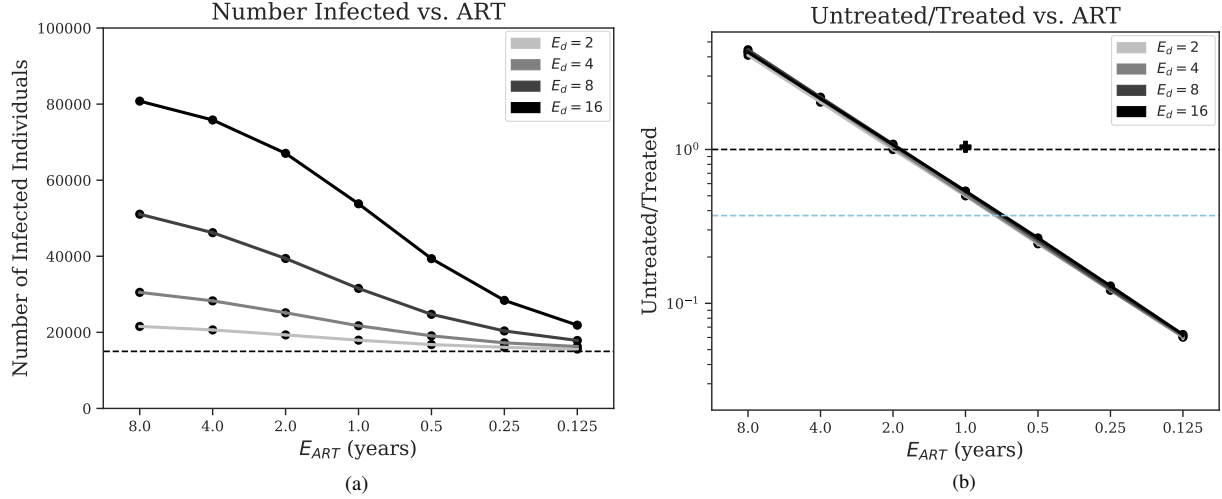


(a)



(b)

**Figure 1.3:** Kernel density estimates of the distributions of (a) patristic distances (path length) between all pairs of sequences and (b) branch lengths of real and simulated datasets for the San Diego (SD) and Uganda (UG) datasets. Averages are shown as dots (Fig. A.2). Black denotes distributions computed from true (simulated) trees and grey denotes distributions computed from trees inferred from sequences (IQ-TREE for real and FastTree 2 for simulated data). Note that real data only have inferred pairwise distances and branch lengths, as true branch lengths are not known.  $E_{ART}$  is the expected time to start ART.



**Figure 1.4:** Sensitivity analysis of epidemiological outcomes. We show (a) the total number of infected individuals, and (b) the ratio of the number of untreated vs. the number of treated individuals (log-scale), vs. expected time to begin Antiretroviral Therapy ( $\mathbb{E}_{ART}$ ) for the BA model with various mean contact numbers ( $\mathbb{E}_d$ ) with all other parameters set to base values. Untreated/treated = 1 is shown as a dashed black line, and the value of untreated/treated corresponding to the “90-90-90” goal [105] is shown as a dashed blue line ( $(1 - 0.9^3)/0.9^3 \approx 0.37$ ). The Untreated/Treated value corresponding to the simulated Uganda dataset has been shown as a + symbol on (b).

ER (at most 7%) and WS (at most 8%) models (Fig. A.6), but these differences are marginal compared to impacts of  $\mathbb{E}_{ART}$  and  $\mathbb{E}_d$  (which, when changed, leads to 43% and 152% change, respectively, in the number of infected people compared to the base parameters). Finally, Uganda simulations lead to higher infection count (64% versus 45%) compared to San Diego (Table A.7).

## Treated Population

The ratio of untreated to treated individuals is a function of  $\mathbb{E}_{ART}$  but not  $\mathbb{E}_d$  (Fig. 1.4b). Note that this ratio remains constant (at most 14.7% change) after year 4, has small changes in year 1 to 4, and experiences an initial period of instability for about 1 year (Fig. A.5), likely because all seeds are initially AU. With  $\mathbb{E}_{ART} = 1$  years, the ratio is on average 0.507 after year 2; decreasing/increasing  $\mathbb{E}_{ART}$  reduces/increases the portion of untreated people. The “90-90-90” campaign by the Joint United Nations Programme on HIV/AIDS (UNAIDS) [105] aims to have

90% of the HIV population diagnosed, of which 90% should receive treatment, of which 90% (i.e., 72.9% of total) should be virally suppressed. Reaching the 90-90-90 goals in the epidemic we model here requires  $\mathbb{E}_{ART}$  between  $1/2$  and 1 year (assuming that lack of viral suppression is fully attributed to lack of adherence). These results are stable with respect to model of contact network,  $\mathbb{E}_d$ , and seed selection approach (Figs. 1.4b and A.7). The only model choice that had a noticeable effect on the results is the use of the ER network model, which led to an increase in Untreated/Treated for  $\mathbb{E}_d \leq 4$  (Fig. A.7). We note that our simulated Uganda epidemic had twice the ratio of Untreated/Treated compared to base San Diego (Table A.7).

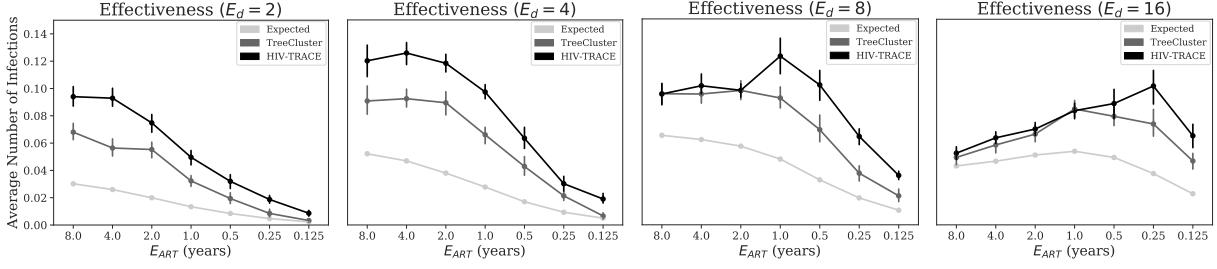
### 1.3.3 Evaluating Inference Methods

#### Phylogenetic Error

From simulated sequences, we inferred trees under the GTR+ $\Gamma$  model using FastTree 2 [102], and we computed the normalized Robinson–Foulds (RF) distance (i.e., the proportion of branches included in one tree but not the other [106]) between the true trees and their respective inferred trees (Fig. A.8). For all model conditions, the RF distance is quite high (0.36-0.58 for San Diego and 0.25-0.40 for Uganda). However, we note that our datasets include many extremely short branches, defined here as those where the expected number of mutations along the branch across the entire sequence length is lower than 1. In our simulations, we have between 16% and 30% of branches that are extremely short (Fig. A.8) and therefore hard to infer.

#### Clustering Methods

We measure the number of new infections caused by each person in the clusters with the highest growth rate and compare it with the same value for the total population (Fig. 1.5). Over the entire population, the average number of new infections caused by each person between years 9 and 10 is 0.028 for our base parameter settings. The top 1,000 people from the fastest growing



**Figure 1.5:** The effectiveness of clustering methods in finding high risk individuals. The average number of new infections between years 9 and 10 of the simulation caused by individuals infected at year 9 in growing clusters. We select 1,000 individuals from clusters, inferred by either HIV-TRACE or TreeCluster, that have the highest growth rate (ties broken randomly). As a baseline control, the average number of infections over all individuals (similar to expectations under a random selection) is shown as well. For a cluster with  $n_t$  members at year  $t$ , growth rate is defined as  $\frac{n_9 - n_8}{\sqrt{n_9}}$ . The columns show varying expected degree (i.e., number of sexual partners), and all other parameters are their base values.

TreeCluster clusters, in contrast, infect on average 0.066 new people.

Thus, the top 1000 people chosen among the growing clusters according to TreeCluster are more than twice as infectious as a random selection of 1000 individuals. HIV-TRACE performs even better than TreeCluster, increasing the per capita new infections among top 1,000 individuals to 0.097 for base parameters, a 3.46x improvement compared to the population average. As  $\mathbb{E}_{ART}$  decreases, the total number of per capita new infections reduces; as a result, the positive impact of using clustering methods to find the growing clusters gradually diminishes (Fig. 1.5). Conversely, reducing  $\mathbb{E}_{ART}$  leads to further improvements obtained using TreeCluster versus random selection and using HIV-TRACE versus TreeCluster.

Changing  $\mathbb{E}_d$  also impacts the results (Fig. 1.5). When  $\mathbb{E}_d = 2$ , slowing the epidemic down compared to the base case, both methods remain better than random, and HIV-TRACE continues to outperform TreeCluster. However, when  $\mathbb{E}_d$  is increased, the two methods first tie at  $\mathbb{E}_d = 8$ , and at  $\mathbb{E}_d = 16$ , TreeCluster becomes slightly better than HIV-TRACE for most  $\mathbb{E}_{ART}$  values (Fig. 1.5). The advantage compared to a random selection of individuals is diminished (improvements never exceed 70%) when the epidemic is made very fast growing by setting  $\mathbb{E}_{ART} \geq 2$  and  $\mathbb{E}_d = 16$ .

## 1.4 Discussion

Our results demonstrated that FAVITES can simulate under different models and can produce realistic data. A comparison of the fit between real and simulated data for Uganda and San Diego points to the importance of data availability. For San Diego, where more studies have been done and more sequence data were available, the fit between simulated and true data was generally good (Table A.6). For Uganda, we had to rely on several sources (e.g. data from McCreech *et al.* (2017) [100] and LANL), and we had a reduced fit between simulations and real data. Increased gathering and sharing of data, including sequence data, can in future improve our ability to parameterize simulations.

Although we only explored viral epidemics, FAVITES can easily expand to epidemics caused by other pathogens for which molecular epidemiology is of interest [107]. We also showed that TreeCluster and HIV-TRACE, when paired with temporal monitoring, can successfully identify individuals most likely to transmit, and HIV-TRACE performs better than TreeCluster under most tested conditions. The ability to find people with increased risk of onward transmission is especially important because it can potentially help public health officials better spend their limited budgets for targeted prevention (e.g. Pre-Exposure Prophylaxis (PrEP)) or treatment (e.g. efforts to increase ART adherence).

We studied several models for various steps of our simulations, but we did not exhaustively test all models: FAVITES currently includes 21 modules and a total of 169 implementations (i.e., specific models) across them, and testing all model combinations is infeasible. To simulate San Diego and Uganda, we aimed to choose the most appropriate set of 21 sub-models available in FAVITES to create the end-to-end simulations. Each of these 21 sub-models has its own limitations, as models inevitably do. However, it must be noted that limitations resulting from model assumptions are limitations of the specific example simulation experiment we performed in this manuscript, rather than limitations of the framework: FAVITES is designed specifically to

be flexible, allowing the use of different models for different steps. If better models are developed for each of these 21 modules, they can be easily incorporated. Like all statistical modeling, appropriate choice of model assumptions is essential to the interpretation of the simulation results, and it is important for the user to choose models appropriate to their specific epidemic of interest. To aid users, our extensive documentation provides descriptions for each module implementation and we provide model validation scripts.

For the simulation of HIV epidemics, novel statistical models can be created to address the unrealistic assumptions. For example, our contact network remains unchanged with time, whereas real sexual networks are dynamic. Our transmission model does not directly model effective prevention measures such as PrEP. Our sequences include substitutions, but no recombination. Moreover, the models of sequence evolution we used ignore many evolutionary constraints across sites. We also ignored infections from outside the network (viral migration), assumed full patient sampling, and we sampled all patients at the end time as opposed to varied-time sampling. While these and other choices may impact results, we note that our goal here was mainly to show the utility of FAVITES. We leave an extensive study of the impact of each of these factors on the results to future studies. Importantly, new models with improved realism to address these issues can easily be incorporated, and continued model improvement is a reason why we believe flexible frameworks like FAVITES are needed.

We observed relatively high levels of error in inferred phylogenies. This is not surprising given the low rate of evolution and length of the *pol* region (which we emulate). Further, our phylogenies include many super-short branches, perhaps due to our complete sampling. Many transmission cluster inference tools (e.g. PhyloPart, Cluster Picker, and TreeCluster) use phylogenies during the inference process and thus may be sensitive to tree inference error. Other tools like HIV-TRACE do not attempt to infer a full phylogeny (only distances). The high levels of tree inference error may be partially responsible for the relatively lower performance of TreeCluster compared to HIV-TRACE. Nevertheless, TreeCluster had higher per capita new

infections in its fastest growing clusters than the population average, indicating that the trees, although imperfect, still include useful signal about the underlying transmission histories.

Using FAVITES, we compared TreeCluster and HIV-TRACE in terms of their predictive power, and our results complement studies on real data [22]. Nevertheless, our simulations study has some limitations that should be kept in mind. A major limitation is that both methods we tested use a distance threshold internally for defining clusters. The specific choice of threshold defines a trade-off between cluster sensitivity and specificity, and the trade-off will impact cluster compositions. The best choice of the threshold is likely a function of epidemiological factors, and the default thresholds are perhaps optimal for certain epidemiological conditions, but not others. For example, we observed that, for a minority of our epidemiological settings, TreeCluster is more effective than HIV-TRACE in predicting growing clusters. A thorough exploration of all epidemiological parameters and method thresholds is left for future studies. On a practical note, FAVITES can enable public health officials to simulate conditions similar to their own epidemic and pick the best method/threshold tailored to their situation.

The approach we used for evaluating clustering methods, despite its natural appeal, is not the only possible measure. For example, the best way to choose high-risk individuals given clustering results at one time point or a series of time points is unclear. We used a strict ordering based on square-root-normalized cluster growth and arbitrary tie-breaking, but many other metrics and strategies can be imagined [103]. For example, we may want to order individuals within a cluster by some criteria as well and choose certain number of people per cluster inversely proportional to the growth rate of the cluster. We simply chose 1,000 people to simulate a limited budget, but perhaps reducing/increasing this threshold gives interesting results. A thorough exploration of the best method for each budget is beyond the scope of this work. Similarly, we leave a comprehensive study of the best strategies to allocate budgets based on the results of clustering and better ways of measuring effectiveness, to future work.



## 1.5 Acknowledgements

This work was supported by NIH subaward 5P30AI027767-28 to SM and NM and an NIH-NIAID K01 Career Development Award (K01AI110181), an NIH-NIAID R01 (AI135992), and a California HIV/AIDS Research Program (CHRP) IDEA Award (ID15-SD-052) to JOW. Computations were performed using XSEDE, supported by the NSF grant ACI-1053575.

Chapter 1, in full, is a reprint of the material as it appears in “FAVITES: Simultaneous Simulation of Transmission Networks, Phylogenetic Trees, and Sequences” (2018). Moshiri, Niema; Ragonnet-Cronin, Manon; Wertheim, Joel; Mirarab, Siavash, *Bioinformatics*, bty921. The dissertation author was the primary investigator and first author of this paper.

## **Chapter 2**

# **A Two-State Model of Tree Evolution and its Applications to *Alu* Retrotransposition**

Models of tree evolution have mostly focused on capturing the cladogenesis processes behind speciation. Processes that derive the evolution of genomic elements, such as repeats, are not necessarily captured by these existing models. In this paper, we design a model of tree evolution that we call the dual-birth model, and we show how it can be useful in studying the evolution of short *Alu* repeats found in the human genome in abundance. The dual-birth model extends the traditional birth-only model to have two rates of propagation, one for active nodes that propagate often, and another for inactive nodes, that with a lower rate, activate and start propagating. Adjusting the ratio of the rates controls the expected tree balance. We present several theoretical results under the dual-birth model, introduce parameter estimation techniques, and study the properties of the model in simulations. We then use the dual-birth model to estimate the number of active *Alu* elements and their rates of propagation and activation in the human genome based on a large phylogenetic tree that we build from close to one million *Alu* sequences.

## 2.1 Introduction

Phylogenetic trees can be used to study the evolution of not just species, but of any sequence that evolves. For example, multi-copy gene families [1, 108], cancer genomes [3, 2], antibodies [4, 5, 6], segmental duplicates [7, 8], and long or short interspersed nuclear elements [9] are all biological sequences that evolve, and many of these evolve *within* the genome of a single species. The process of diversification for many evolving entities can be characterized by propagation: an original copy of a sequence creates a new copy, and the two copies evolve independently by accumulating mutations. Phylogenetics provides a natural framework for studying such processes, but several challenges present themselves.

Given sufficiently long sequences and assuming our models of sequence evolution are reasonably accurate, we can recover the phylogenetic trees from sequence data with high accuracy [50, 109]. However, unlike species-tree reconstruction, in which the entire genome can

be used, reconstructing phylogenies of gene families, repeats, or antibodies is limited by the length of the evolving entity. As a result, high levels of uncertainty are to be expected in trees reconstructed from these types of sequences. These inherent limitations make accurate modeling of underlying processes crucial, perhaps even more so than species-tree reconstruction.

Models of tree evolution describe probability distributions over the space of tree shapes [29, 110, 30] and are useful in several ways. They can be used as the prior distribution in a Bayesian inference [31, 32, 33]. They can also generate null distributions describing certain neutral evolutionary process, which may then be rejected by trees inferred from the data [34, 35, 36]. Moreover, the diversification parameters are inherently of interest to the biologist [37]. Two well-known models of tree evolution are Yule (birth-only), in which each branch splits by a Poisson process with a constant rate, and birth-death, in which, in addition to birth, branches can go extinct with a constant rate. Each of these models have limitations and have inspired the development of several alternative models [111, 112, 113, 114, 115, 116].

A main feature of a tree evolution model is the expected tree shape, especially the tree balance (Fig. 2.1). The Yule model generates relatively balanced trees [117], more so than typically seen in phylogenetic trees [114]. Some systems, such as certain viruses, are especially known to have very unbalanced trees [118]. Most models of evolution are exchangeable, meaning that, after a split, the two branches are indistinguishable. When evolution works in a series of propagation events (i.e., where an element copies itself), there is a natural way in which one of the child branches corresponds to the parent branch [119]. The new copy may have properties that are different from the original branch, and as a result, non-exchangeable models may be more appropriate. For example, the new child may be initially incapable of propagation until it *activates*. In such situations, the tree will tend to be unbalanced. In the limit, if every new child is impotent, one would expect a *caterpillar*-like tree (Fig. 2.1).

In this paper, we study a non-exchangeable extension of the Yule model, which we name the dual-birth model. Each branch will split with one of two available rates. Branches that

correspond to elements that have in the past propagated are considered *active* and have a high rate of future propagation, whereas branches that have never propagated are considered *inactive*. With some rate, the Unlike some previous models (e.g. BiSSE [116]), after every birth event, one of the two children inherits the parent's rate while the other child has the opposite rate (i.e., the model is asymmetric in the terminology of Lambert and Stadler (2013) [119]). For this dual-birth model, we describe methods for sampling the tree distribution, derive probability distributions on the tree space, compute the expectation of various tree statistics, and introduce methods of estimating the model parameters from data. In extensive simulations, we study the behavior of the model and our estimators. We then use the model to study the evolution of *Alu* elements in the human genome.

*Alu* elements are a family of Short Interspersed Nuclear Elements (SINEs), each approximately 300 base pairs (bp) long, that abound in the genomes of supraprimates and that retrotranspose via Ribonucleic Acid (RNA) polymerase III-encoded RNAs [9]. There are approximately one million *Alu* elements in the human genome, meaning they comprise roughly 10% of the human genome. Although *Alu* elements have no known biological function of their own [120], studying and understanding their retrotransposition in the genome can provide key insight into their contributions to genetic disease [121] as well as useful information in the study of supraprimate evolution [122, 123, 124].

A topic of interest regarding *Alu* elements is the number and identity of repeats that are capable of propagating through retrotransposition [57, 125, 126]. Various hypotheses range from the single source model to the transposon model, where all elements are assumed to be equal in their ability to propagate [57]. We approach this question using phylogenetics and the dual-birth model. We build a tree for close to one million *Alu* elements. Using the properties of the dual-birth model, we estimate the number of *Alu* elements that have been active and estimate the rates of *Alu* propagation and activation.

## 2.2 Materials and Methods

### 2.2.1 Dual-Birth Model

The dual-birth model is similar to the Yule process, but unlike Yule, it is not *exchangeable*, meaning that left and right branches are not generated using identical processes. The dual-birth model is parameterized by two birth parameters:  $\lambda_a$  and  $\lambda_b$ . The generative process starts with a single root node, which immediately splits into two child branches *left*, denoted by  $a$ , and *right*, denoted by  $b$ . Further birth events occur on each child branch according to a Poisson process with the constant rate  $\lambda_a$  on branch  $a$  and the constant rate  $\lambda_b$  on branch  $b$ . Thus, *left* and *right* are governed by different rates. Each new node becomes the root of an identical process. The process can be terminated at any point in time. This generates an unlabeled *ordered*, also known as oriented [119], tree: each branch is labeled as either *left* or *right* (Fig. 2.1a). We define  $r = \lambda_a/\lambda_b$  and  $\lambda = \lambda_a + \lambda_b$ , which together identify  $\lambda_a$  and  $\lambda_b$ . When  $r = 1$ , the dual-birth process is reduced to the Yule process with a rate of  $\lambda/2$ .

#### Active/Inactive Elements

Consider a tree in which each branch corresponds to some entity, and the right child of any branch corresponds to the same entity as the parent. Thus, each split is a propagation of the parent entity. Moreover, entities are either *active* or *inactive*. A branch is active if it has produced an offspring before and is otherwise considered inactive. The right child of any branch is always active while the left one is inactive. Active entities propagate with rate  $\lambda_b$  (for “birth”), and inactive entities activate and simultaneously propagate with rate  $\lambda_a$  (for “activation”). Note that activation and the first propagation occur together (an alternative model could be that nodes activate mid-branch and wait for a birth event). Once an entity activates, it remains active (thus, there is no deactivation).

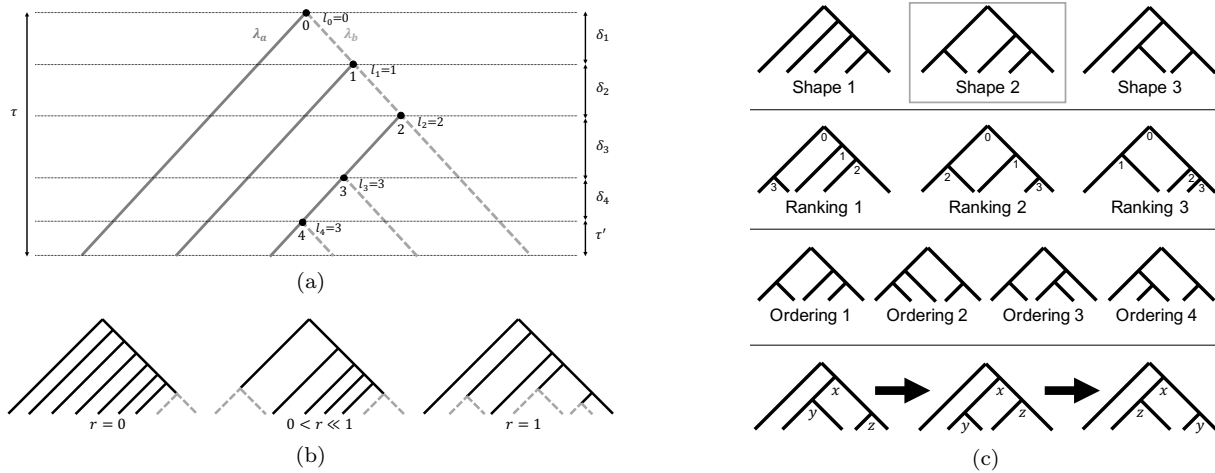
The dual-birth model can easily capture this scenario. If  $r = 1$  (i.e., the Yule model),

active and inactive nodes have the same rates of birth, and thus, their difference is inconsequential. When  $r < 1$ , new entities activate (i.e., propagate for the first time) with a rate  $\lambda_a$  that is lower than the rate  $\lambda_b$  with which nodes that are already active propagate (Fig. 2.1b). Allowing  $\lambda_a > \lambda_b$  would result in  $r > 1$ , which yields a model that is non-identifiable with the model that has rate  $1/r$ . Setting  $r > 1$  would correspond to a scenario where the rate of propagation *reduces* after the first activation, and we don't know of any scenario that justifies such reductions. Thus, our model defines  $\lambda_a \leq \lambda_b$  to remain identifiable.

One application of the dual-birth model is to study *Alu* elements, though the model may prove useful for other systems, such as retroviruses or gene families. Each *Alu* element appears at a specific position in the genome, and via retrotransposition, it can create a new copy of itself elsewhere in the genome, leading to a split in the repeat evolutionary tree. Each branch of the tree can thus be labeled by a position in the genome, which is the site at which the corresponding element resides. One child branch inherits the same position as the parent (and is thus active), and the other branch is the new copy, which is assumed to be initially inactive. The inactive state captures the observation that most *Alu* elements don't propagate [54]. The model allows for the chance that some inactive elements become active and start propagating, perhaps due to mutations or due to changes in their genomic context.

## Tree Balance

The Yule model generates balanced trees, more so than trees typically found in phylogenetic databases [34, 114, 115]. Similar to several other models of tree evolution [111, 114, 116], the dual-birth model provides a natural way to control tree balance (we provide an extensive comparison to other models in the Discussion section). Consider an extreme case in which only one element is ever active. The resulting tree is a caterpillar, which will include only one cherry (an internal node is called a cherry if both of its children are leaves, i.e., terminal nodes). This outcome can be naturally achieved in dual-birth by setting  $\lambda_a = 0$  (i.e.,  $r = 0$ ). When we expect



**Figure 2.1:** Dual-birth model. (a) A caterpillar tree with one cherry (node 4). The tree is generated by the dual-birth model; right branches (dashed light gray) are sampled from the Poisson process with rate  $\lambda_b$ , and left branches (solid dark gray) are sampled with rate  $\lambda_a$ . Internal nodes are ranked by distance to the root (ranks shown below nodes), and the tree is divided into time intervals between consecutive nodes. (b) With  $r = \lambda_a/\lambda_b = 0$ , only caterpillar trees can be generated; as  $r$  increases toward 1, the tree becomes more balanced and thus has more cherries (dashed light gray). (c) All three possible tree shapes with five leaves are shown on top; the second row shows  $\Psi$ , all possible rankings of tree shape 2; the third row shows  $\Omega$ , all orderings of tree shape 2. The last row demonstrates that, starting from a ranked ordered tree, one change of ranking followed by a change of ordering results in a tree identical to the original tree.

to have many more inactive nodes than active nodes, we would still expect to see an unbalanced tree with few cherries. This outcome, too, can be achieved by a natural choice of  $\lambda_a \ll \lambda_b$ , which results in  $r \ll 1$ . As  $r$  increases, the tree becomes gradually more balanced (Fig. 2.1b). With  $r = 1$ , the tree is as balanced as expected under the Yule model.

## 2.2.2 Theoretical Properties of the Dual-Birth Model

### Notations and definitions

A connected Directed Acyclic Graph (DAG) with no undirected cycles defines a tree. We only consider binary trees in which all nodes either have outdegree zero (leaves) or two (internal nodes). Two trees are considered to have the same *shape* if there exists a one-to-one mapping



between their nodes such that the head and tail of every edge in one tree map to the head and tail of exactly one edge in the other tree. In this paper, we care about the space of distinct tree shapes. In the tree-shape space, leaves are not distinguished (i.e., a tree shape is unlabeled). For simplicity of presentation, we represent a tree shape on  $n$  leaves using  $T = (V, E)$ , where  $V$  is the set of  $n - 1$  *internal* nodes and  $E$  is the set of internal edges  $(u, v)$  from parent node  $u$  to child node  $v$ . Note that terminal edges (connecting internal nodes to leaves) and leaves are not part of  $E$  and  $V$ , and as such, are implicit in the  $T$  formulation (each internal node has to have outdegree two). We use  $u_1$  and  $u_2$  to denote the children of  $u$ , and we use  $\otimes$  to denote a generic unlabeled leaf. Note that  $T$  defines a Partially Ordered Set (POSET) on  $V$ .

Recall a node  $v \in V$  is called a cherry if both of its children are leaves. A tree shape is called *caterpillar* if it has only one cherry (Fig. 2.1a); in contrast, a fully-balanced tree has exactly  $n/2$  cherries. The number of cherries of a tree is denoted as  $c(T)$ .

Let  $N = \{0, 1, \dots, n - 2\}$ . A bijective function  $\psi : V \mapsto N$  is a ranking of a tree  $T = (E, V)$  if for each edge  $(u, v) \in E$ , we have  $\psi(u) < \psi(v)$ . A *ranked tree shape* is defined as  $T^\psi = (T, \psi)$ . Each ranking is a topological sorting of the tree (i.e., is a linear extension of the POSET defined by the tree shape). We use  $\Psi(T)$  to denote the set of all possible rankings of the tree shape  $T$  (Fig. 2.1c).

An *ordered tree shape* is a tree shape in which left and right child nodes are distinguished (i.e., the tree is oriented). More precisely,  $\omega : V \mapsto \{0, 1\}$  is a valid order for a tree shape  $T$  iff  $\omega(u_1) + \omega(u_2) = 1$  for every  $(u, u_1), (u, u_2) \in E$  and  $\omega(r) = 1$  for the root node  $r$ . We call  $v$  a left child/node when  $\omega(v) = 0$  and otherwise call it a right child/node. A branch leading to a left (right) child is called a left (right) branch. An ordered tree shape is denoted by  $T^\omega = (T, \omega)$ . Note that, in this definition, leaves are not directly assigned a left/right side. Leaves below a cherry are indistinguishable; leaves that are sister to internal nodes are considered to have the opposite side of their sibling. For example, in Figure 2.1a, the leaf directly below the root is considered a left node because its sister, the node ranked 1, is a right node. Also,  $\Omega(T^\psi)$  denotes the set of all

possible orderings that are valid for  $T^\Psi$  (Fig. 2.1c).

A ranked ordered tree shape is defined by  $T_\omega^\Psi = (T, \psi, \omega)$ . For ease of notation, we define  $\omega_i = \omega(\psi^{-1}(i))$  (the order of the node ranked  $i$ ). For  $0 < i < n$  and the ranked ordered tree  $T_\omega^\Psi$ , we define  $l_i(T_\omega^\Psi) = 1 + \sum_{k=1}^{i-1} \omega_k$  and it is easy to show that  $l_i(T_\omega^\Psi)$  gives the number of left branches  $(u, v)$  with  $\omega(u) < i$  and  $\omega(v) \geq i$ . In other words,  $l_i(T_\omega^\Psi)$  gives the number of left terminal branches if the tree  $T_\omega^\Psi$  is terminated at the time when node  $i$  is created. Where clear by the context, we simply write  $l_i$  (Fig. 2.1a). We define  $n_l = l_{n-1}$  and  $n_r = n - n_l$ ; these definitions can be intuitively considered to show the number of left and right terminal branches, respectively, if we assign an order to all terminal branches (e.g.  $n_r = n_l = 3$  in Fig. 2.1a).

We refer to a tree shape with ultrametric branch lengths as a weighted shape. A weighted shape is defined by  $t = (T, \delta, \tau)$ , where  $\delta : E \mapsto \mathbb{R}$  gives the length of internal branches and  $\tau$  gives the distance from the root to all leaves; note that  $\tau$  has to be larger than the largest distance to the root from any internal node. Node ages define a unique ranking on any weighted shape. A weighted shape  $t$  can also be assigned an order,  $\omega$ , and will be denoted by  $t^\omega$ . For  $e = (u, v) \in E$ , we refer to  $\delta(e)$  by  $\delta_i$  where  $i = \psi(v)$  (e.g.  $\delta_1 \dots \delta_4$  in Fig. 2.1a).

## Probability Distributions

We now derive probability distributions on tree shapes conditioning on fixed  $n$ . Here we give the main results and provide the proofs in Section B.1.1.

**Theorem 1.** *Let  $X$  be a random variable (r.v.) over ordered ranked tree shapes and distributed according to the dual-birth model with parameter  $r = \lambda_a/\lambda_b$ . Then,*

$$\Pr(X = T_\omega^\Psi; n) = \frac{r^{n_r-1}}{\prod_{i=1}^{n-2} ((r-1)l_i + i + 1)} \quad (2.1)$$

Computing Equation 2.1 simply requires knowing the number of right leaves ( $n_r$ ) and the number of its left branches if the tree is terminated at each node  $i$  ( $l_i$ ); all of these can be computed

in time  $O(n)$  for an ordered tree. Figure B.1 shows the perfect match between Equation 2.1 and observed frequencies in simulations for all ranked ordered tree shapes with  $n = 6$  and shows that, with  $r \ll 1$ , the caterpillar tree shape has a high probability.

The left/right order of nodes cannot be estimated from sequence data, and thus, it would be more useful to compute the probability distribution over unordered ranked tree shapes. Since all orderings of a *ranked* tree are distinct, the probability of a ranked tree simply needs to marginalize over all possible orderings. Thus,

**Corollary 1.** *For  $Y$ , an r.v. over ranked tree shapes with  $n$  leaves and distributed according to the dual-birth model,*

$$\Pr(Y = T^\Psi; n) = \sum_{\omega \in \Omega(T^\Psi)} \Pr(T_\omega^\Psi) \quad (2.2)$$

where  $\Omega$  gives the set of all orderings of  $T^\Psi$ .

This computing requires an exponential number of computations to iterate all orderings (the recursive formula for that iteration is given in Equation B.5. Whether efficient algorithms for computing this probability exist is unclear to us. See Figure B.1 for an example distribution and matching simulations.

Next, we turn to computing the probability distribution over unranked shapes. This can be done by enumerating all possible rankings of the unranked tree and summing up their probabilities. The set of all rankings,  $\Psi(T)$ , is simply the set of all the linear extensions of the POSET defined by the tree shape. However, a final complication needs to be addressed. Recall that leaves are unlabeled. For a non-cherry symmetric node  $u$  (i.e., sub-tree shapes below  $u_1$  and  $u_2$  are identical), take any ordering of any ranking of nodes below  $u$ . Now swap  $\omega(u_1)$  and  $\omega(u_2)$  and also swap the rankings of nodes below  $\omega(u_1)$  with the rankings of the identical nodes under  $\omega(u_2)$  (Fig. 2.1c); this would produce an identical tree shape. However, our process will count this identical tree shape twice. To account for this, we need to divide the total probability by two for every non-cherry symmetric node. Thus,

**Corollary 2.** For  $Z$ , an r.v. over tree shapes with  $n$  leaves and distributed according to the dual-birth model,

$$\Pr(Z = T; n) = \frac{1}{2^{\sigma(T)}} \sum_{\psi \in \Psi(T)} \Pr(T^\psi) \quad (2.3)$$

where  $\Psi$  gives all rankings of  $T$  and  $\sigma(T)$  is the number of non-cherry symmetric nodes in  $T$ .

### Weighted Trees

Given an ordered weighted tree shape  $T_\omega^\psi$ , we can easily compute the probability density function (p.d.f) for the length of each of its internal branches. Recall  $\delta_i$  is the time between internal nodes ranked  $i - 1$  and  $i$  (i.e., an interval), which is simply the length of a specific branch. Given the tree shape,  $\delta_i$  follows an exponential distribution with rate  $\lambda_i = \lambda_a l_i + \lambda_b (i + 1 - l_i)$ . This is because the branch length is simply the minimum of all exponential r.v.s active in the corresponding interval, which itself, is an exponential r.v. with the total rate. Furthermore, since each interval is independent of the other intervals given  $T_\omega^\psi$ , the joint probability density of  $T_\omega^\psi$  and a set of internal branch lengths can be computed by multiplying the probability of  $T_\omega^\psi$  (Eq. 2.1) by the probability density of every branch length given  $T_\omega^\psi$ . Finally, to compute the joint probability density of a given tree and all its branch lengths, we need to also multiple the probability of no births in the final interval of length  $\delta_{n-1} = \tau - \sum_1^{n-2} \delta_i$ ; this is the probability of no events for an exponential with rate  $l_{n-1}$  in time  $\delta_{n-1}$ ; i.e.,  $e^{-l_{n-1} \delta_{n-1}}$ .

### Expected Number of Cherries and Active Leaves

We now ask the following question: how many cherries and how many active nodes are expected in a dual-birth tree generated with rate ratio  $r$ ?

A parsimony analysis is constructive here. Activation events can be considered evolutionary changes. Given an unordered tree, the most parsimonious ordering is one that implies the minimum number of activation events. The number of activations is simply the number of left

branches that have children. By making every internal node that is sister to a leaf a right node and arbitrarily ordering the rest, one can show that the most parsimonious ordering has exactly one activation for each cherry in the tree, counting the root as an activation (Lemma 1). The parsimony analysis would only be relevant for  $r \ll 1$ , when one would expect very few activations. As  $r$  increases, the tree becomes more balanced, and we would expect more cherries. We now formalize this intuition.

**Theorem 2.** *For a tree shape  $Z$  generated by the dual-birth model with  $r = \lambda_a/\lambda_b$ , let  $C = c(Z)/n$  be an r.v. capturing the fraction of cherries; then,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(C) = \frac{\sqrt{r}}{1 + r + \sqrt{r}} \quad (2.4)$$

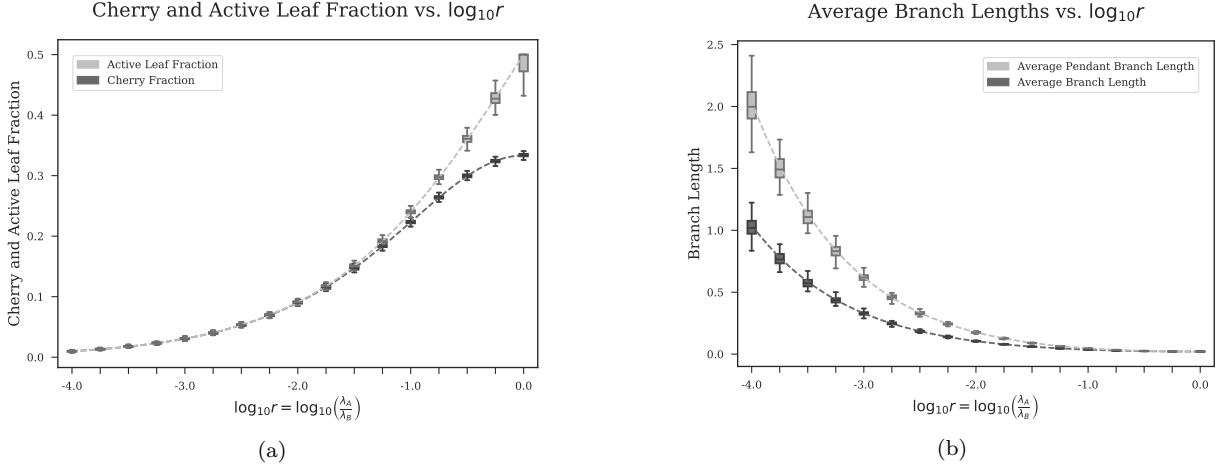
**Corollary 3.** *For an r.v.  $N_r$  capturing the fraction of right (i.e., active) leaves in tree shape  $T$ ,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(N_r) = \frac{\sqrt{r}}{1 + \sqrt{r}} \quad (2.5)$$

The proofs can be found in Section B.1.1. As Figure 2.2a shows, these expectations closely match simulation results. As  $r$  increases, the expected frequency of cherries increases until it reaches its peak at  $1/3$  for  $r = 1$ . The number of active elements follows a similar pattern and reaches its peak at  $1/2$  for  $r = 1$ . Thus, under the Yule model, only half the nodes will be expected to be active (recall that an active element is defined as one that has already propagated). Also note  $r = x$  and  $r = \frac{1}{x}$  differ only in what elements are labeled left or right, and thus, they are indistinguishable for unordered trees. As noted before,  $r > 1$  does not have a meaningful interpretation in biological processes that we consider; thus, we focus on  $0 \leq r \leq 1$ .

### Expected Branch Length

A natural quantity of interest under any model of tree evolution is the expected length of a random branch. For example, under the Yule model, branches are exponentially distributed with



**Figure 2.2:** Theoretical expectations of (a) cherry fraction (dashed dark gray line) and active leaf fraction (dashed light gray line), and (b) branch length (dashed dark gray line) and pendant branch length (dashed light gray line) versus simulated distributions (in box plots) using 100 replicates with  $n = 4096$ ,  $\lambda = 48$ , and varying values of  $r$  ( $x$ -axis) from  $1/1024$  to  $1$ . Note that the number of cherries is the maximum parsimony estimate of the number of active elements, and the most parsimonious estimate works well for low values of  $r$ .

rate  $1/2\lambda$  [32]. In our model, the expected branch length depends on both  $\lambda$  and  $r$ . In all the results given below, we assume all the leaves are sampled.

**Theorem 3.** *For a weighted tree shape  $t$  generated by the dual-birth model with parameters  $r$  and  $\lambda$  conditioned on having  $n$  leaves, let  $D$  be an r.v. giving the length of a random branch in  $t$ ; i.e.,  $D = \delta_I$  for  $I \sim \mathcal{U}(1, n - 2)$ . Then,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(D) \rightarrow \frac{1}{2\lambda} \left( \frac{r+1}{\sqrt{r}} \right) = \frac{1}{\lambda_a} \frac{\sqrt{r}}{2} \quad (2.6)$$

The proof can be found in Section B.1.1. For a fixed  $\lambda$ , increasing  $r$  in  $(0, 1]$  reduces the expected branch lengths, resulting in the minimum value under the Yule model (Fig. 2.2b).

### Expected Terminal Branch Length

Under the Yule model, terminal and internal branch lengths have the same expected length [32]. For  $r \ll 1$ , we expect that inactive entities result in long terminal branches and

relatively short internal branches. These expectations can be confirmed in simulations. As expected, close to  $r = 1$ , the mean terminal branch length is close to the mean branch length but the two values gradually diverge as  $r$  decreases (Fig. 2.2b). Therefore, the difference between average terminal and internal branch lengths is a function of  $r$ , and therefore, a closed-form formula for the expected terminal branch length would be useful in building an estimator of  $r$ . While we don't have a proven result for the terminal branch length, based on simulations, we present a conjecture. Note that this conjecture was purely reached based on our intuition and trial-and-error, starting from Equation 2.6 and modifying the denominator until a close match to the empirical values was obtained.

**Conjecture 1.** *For a weighted tree shape  $t$  generated by the dual-birth model with parameters  $r$  and  $\lambda$  conditioned on having  $n$  leaves, let  $L$  be an r.v. giving the length of a random terminal branch in  $t$ . Then,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(L) \rightarrow \frac{1}{\lambda_a} \left( \frac{\sqrt{r}}{1 + 2\sqrt{r} - r} \right) \quad (2.7)$$

Figure 2.2b shows that the conjectured results closely match the observed mean terminal branches for a wide range of  $r$  values. Note that changing  $\lambda_a$  simply scales all lengths, so our simple simulations have explored *all* free parameters of the dual-birth model (albeit, only for  $r \in [10^{-4}, 1]$  range). Regardless of whether our conjecture is true (which cannot be proven by simulations), the close match of Equation 2.7 and simulated results means that we can use it to provide an approximate estimator of  $r$ .

## Parameter Estimation

Theorem 2 enables us to estimate the  $r$  parameter for a given tree. Given a tree with  $c$  cherries, and for  $x = c/n$ , solving for  $r$  in Equation 2.4 results in the following relationship

(Fig. B.2):

$$\hat{r}(x) = \left( \frac{1 - x - \sqrt{(x+1)(1-3x)}}{2x} \right)^2 \quad (2.8)$$

for  $x \leq \frac{1}{3}$  and else  $\hat{r}(x) = 1$ .

An alternative estimator can be designed by combining Theorem 3 and Conjecture 1 for expected total and terminal branch length. Given a tree with an average branch length of  $d$  and an average terminal branch length of  $l$ , solving Equations 2.6 and 2.7 for  $r$  and  $\lambda_a$ , we can design the following estimator for large  $n$ :

$$\hat{r}(b, l) = \left( 1 - \sqrt{2 \left( 1 - \frac{d}{l} \right)} \right)^2 \quad (2.9)$$

Further approximating the total average branch length to be the mean of internal branch lengths ( $i$ ) and terminal branch lengths (a good approximation for large  $n$ ), we can further simplify Equation 2.9 to:

$$\hat{r}(i, l) = \left( 1 - \sqrt{1 - \frac{i}{l}} \right)^2 \quad (2.10)$$

Having estimated  $r$ , we can easily use Equation 2.6 to get an estimate of  $\lambda$  from the observed mean branch length for large  $n$ . Note that, absent a proof for Conjecture 1, Equation 2.9 should be treated as an approximate estimator. Also note that this approximate estimator assumes the given tree itself was generated by the dual-birth model (e.g. it is not the result of subsampling the tips of a tree generated by the dual-birth model). We discuss statistical properties of both estimators in Section 2.4.2.

### Sampling the Dual-Birth Model

When conditioning on  $n$ , the number of tips, a simple algorithm can be used to sample the space of ordered weighted tree shapes defined by the dual-birth model.

We start with a single-node tree and iteratively add new nodes until the tree has  $n$  leaves.



We use a heap to keep a list of current leaves sorted by their distance to the root. In each iteration, we add two child nodes to the highest leaf in the tree (i.e., the leaf closest to the root); we sample from two exponential distributions with rates  $\lambda_a$  and  $\lambda_b$  for the left and right child's branch lengths, respectively. The two new nodes are added to the heap of leaves, and the parent is removed from the heap. Once the loop has terminated, we truncate the tree by shrinking all terminal edges except the one attached to the leaf that is closest to the root such that all leaves are equidistant to the root.

Hartmann *et al.* have described various strategies for sampling trees with  $n$  leaves [81]. Our sampling procedure falls under what they have termed Simple Sampling Algorithm (SSA). As they point out, the SSA procedure produces the right distribution on tree topologies for pure birth models, like ours, because, once the process reaches  $n$  tips for the first time, it never goes back to having fewer tips. A remaining question is what distribution should be used to decide the time between when  $n$  leaves first become present until we stop the simulation (call it  $\tau'$ ). Let  $h(\tau')$  be the p.d.f of that waiting time. If the time between when we have  $n$  leaves and the the birth of leaf  $n + 1$  is given by  $x > \tau'$ , then  $\tau'$  should be uniformly sampled between zero and  $x$ . Moreover, the probability of sampling each tree with  $n$  leaves should be proportional to  $x$ , the time it remains an  $n$ -leaf tree. Thus, as Hartmann *et al.* show, by summing over all possible values of  $x$ , we get:

$$\begin{aligned} h(\tau') \propto \int_{x=\tau'}^{\infty} x \cdot h(\tau'|x) \cdot g_{n-1}(x) \, dx &= \int_{x=\tau'}^{\infty} x \cdot \frac{1}{x} \cdot g_{n-1}(x) \, dx \\ &= \int_{x=\tau'}^{\infty} g_{n-1}(x) \, dx \end{aligned} \tag{2.11}$$

where  $g_{n-1}$  is the p.d.f of a random variable (r.v.) for  $x$ . In our model, this r.v. is equivalent to the minimum of  $n$  exponential r.v.s,  $l_{n-1}$  of which have rate  $\lambda_a$  and the rest have rate  $\lambda_b$ ; thus,  $g(x)$  is the p.d.f of an exponential with rate  $\lambda_{n-1} = \lambda_a l_{n-1} + \lambda_b(n - l_{n-1})$ . It is easy to see that Equation 2.11 simplifies to  $h(\tau') \propto g_{n-1}(\tau')$ . Thus, the correct waiting time between the last birth event and the end of the simulation is identical to the waiting time for the birth event that would

create  $n + 1$  leaves.

Note that, if we were conditioning on  $\tau$  (the tree height) instead of  $n$ , the same procedure would remain correct, except we would continue until all leaves have at least the required height and would then cut branches that are longer than  $\tau$ .

## 2.2.3 Simulation Setup

### Datasets

Given a set of parameters, we use our implementation of the fixed- $n$  sampling procedure to generate 20 replicate “true” trees. We then deviate each tree from ultrametricity by multiplying each branch of the tree by a multiplier sampled from a gamma distribution with shape and rate both set to  $\alpha$  (with an expected value of 1). For each true tree, we then use INDELible [127] and the GTR+ $\Gamma$  model [48] to simulate a multiple sequence alignment with no indels, which is later used to infer the tree. The simulation parameters are  $n$ ,  $r$ ,  $\lambda$ , and deviation from ultrametricity (i.e., the shape of the Gamma distribution,  $\alpha$ ). For sequence evolution, the parameters to select are  $k$  (the sequence length), the GTR parameters, and the Gamma rate across sites. We also vary model of sequence evolution used for inference.

**Table 2.1:** Experiments (default parameters in bold)

#	Parameter	Parameter Values
1	$r$ (const. bl)	$10^{-4}, 10^{-3}, \mathbf{10^{-2}}, 10^{-1}, 10^0$
2	Model	JC69, K80, HKY85, GTRCAT, <b>GTR+<math>\Gamma</math></b>
3	$\lambda$	33.866, 84.664, <b>169.328</b> , 338.655, 846.64
4	$k$	50, 100, 200, <b>300</b> , 600, 1200, 2400, 4800
5	$n$	25, 50, 250, 500, <b>1000</b> , 2000, 4000
6	$\alpha$ (clock)	2.952, 5.904, <b>29.518</b> , 147.591, 295.182, $\infty$

We perform six *experiments*, each varying a single parameter (Table 2.1). The exception is  $r$ , for which we modify both  $r$  and  $\lambda$  to keep expected branch length constant. Each experiment is centered around a default set of parameters chosen to emulate our *Alu* dataset (details in

Section B.2.1).

## Methods

We infer trees from the simulated sequence data using FastTree-II [102] and RAxML [128]. We estimate  $r$  using the cherry-based (Eq. 2.8) and length-based (Eq. 2.9) estimators.

## Error Measurement

We measure the accuracy of inferred tree topologies using the normalized RF distance [106], which is equal to the proportion of the branches that are different between the true and inferred trees. To account for the sensitivity of RF distance to rogue tips, especially for caterpillar trees, we also compute the Matching Split (MS) metric, implemented in TreeCmp [129]. We compute differences between the log-likelihood scores of true and inferred trees using RAxML. To measure the accuracy of our estimates of  $\lambda$  and  $r$ , we compute the log-ratio of true versus inferred values for both parameters and show the resulting distribution.

### 2.2.4 Human *Alu* Dataset

Most analyses of *Alu* elements have relied upon the classification of elements into subfamilies and using consensus sequences. Because the subfamily classification is potentially incomplete [55], we analyze a large dataset of 885,011 *Alu* repeats.

## Data Acquisition

We use the Dfam database [130] to search for *Alu* repeats. We first create a database containing only Human *Alu* profile HMMs from Dfam. We then use nhmmer (via Dfam's `dfamscan.pl` script) to scan the hg19 reference genome using this subset of Dfam. nhmmer computes a bitscore for each result, which is a metric of how well the sequence matches its respective profile HMM in comparison to how well a random sequence would match the same

model. Our motivation to use Dfam profile HMMs to scan for *Alu* sequences is their sensitivity to detect sequences with deviation from the subfamily consensus, but this same sensitivity also allows for false hits. To combat this, for each *Alu* subfamily, we filter out all sequences with low bitscores. We use unique bitscore thresholds for each *Alu* subfamily because of the heterogeneity of bitscore distributions across subfamilies (see Figs. B.3 and B.4 for our choices of thresholds).

### **Alignment and Tree Inference**

We estimated a MSA on the set of 936,664 bitscore-filtered *Alu* sequences using PASTA [131]. Some of the sequences in the resulting MSA were short (Fig. B.5), which could negatively impact tree inference. To combat this potential risk, we filtered out any sequences that had less than 200 non-gap characters in the MSA. Our final dataset contained 885,011 full-length *Alu* sequences. Our MSA included some extremely gappy sites, as is expected for any dataset including these many sequences. We masked all sites in the PASTA alignment where 99% or more of characters were gaps. Prior to masking, there was a total of 266,699,287 non-gap characters in the MSA; masking reduced the number to 264,144,814 non-gap characters in the MSA (99.04% retention). Since gaps are treated as missing data in our tree inference methods (and not as phylogenetically informative indels), the removal of 1% of the data should result in minimal loss of phylogenetic information. The consensus sequence of the final alignment (Fig. B.6a) included many conserved sites. We used FastTree 2 [102] to infer a tree on the masked alignment under the GTR+ $\Gamma$  model. The resulting tree was not well-supported (Fig. B.7), a fact that is not surprising considering the short sequence length and low divergence. We also used RAxML [128] to infer a tree on the masked alignment under the GTR+CAT model. Our attempts to infer a tree under the GTR+ $\Gamma$  model using RAxML were unsuccessful.

Finally, to test the stability of our estimates, we performed a series of subsampling experiments in which we performed 20 subsampling replicates for  $n = 1,000, 10,000, \text{ and } 100,000$  sequences, inferred trees using FastTree 2, and estimated parameters from those trees just as

before.

## 2.3 Results

### 2.3.1 Simulations: Dual-Birth Model

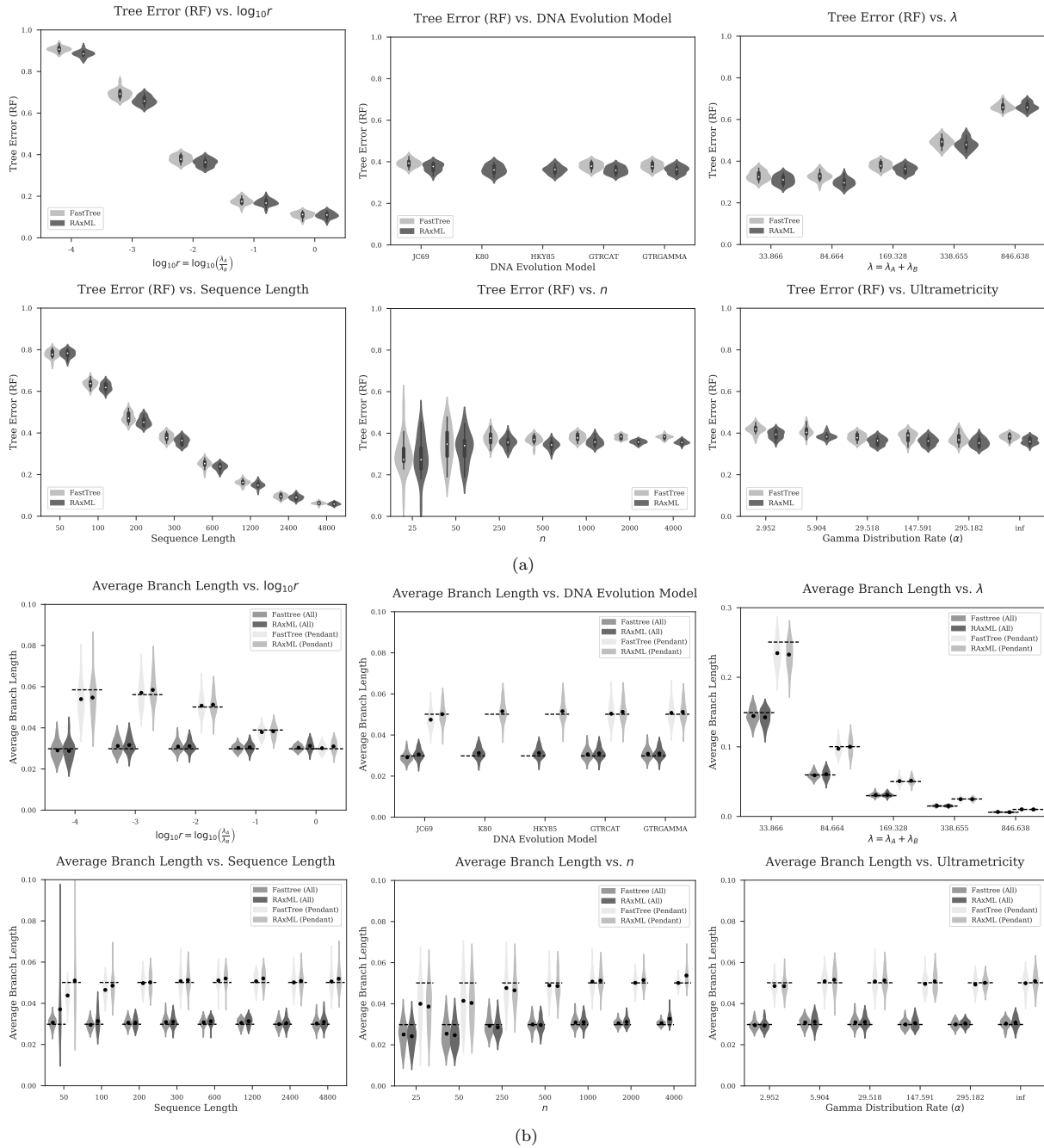
We study the effects of parameters on tree reconstruction accuracy and the ability to estimate  $r$ .

#### Tree Accuracy

The topological tree accuracy was heavily influenced by  $r$ ,  $\lambda$ , and sequence length (Fig. 2.3a). Shortening branch lengths (by increasing  $\lambda$ ) increased the error (Fig. 2.3a, upper-right) as expected. Reassuringly, increasing sequence length reduced the error; while with the default 300 sites, the topological error was 38%, the error reduced to 6% with 4,800 sites (Fig. 2.3a, lower-left).

Most interestingly, the topological error depended on the parameter  $r$  (Fig. 2.3a, upper-left). When  $r = 1$  (i.e., the Yule model), tree estimation error was relatively low. As we reduced  $r$ , which progressively made the true trees less balanced, the topological error quickly increased (Fig. 2.3a, upper-left). With  $r = 10^{-4}$ , where the tree is almost fully unbalanced, the RF error ranged between 85% and 94%. Similar patterns were observed when we used the MS [129] measure of error (Fig. B.8). These extremely high levels of error for unbalanced trees are interesting considering the fact that the sequence length and the expected branch length are kept fixed.

Interestingly, the number of leaves,  $n$ , mostly affected the variance of the topological error. As  $n$  increases, the average tree error remained relatively constant, but its variance gradually reduced (Fig. 2.3a, lower-center). Deviations from the clock had very small impact on the topological tree accuracy (Fig. 2.3a, lower-right). The choice of the sequence evolution model



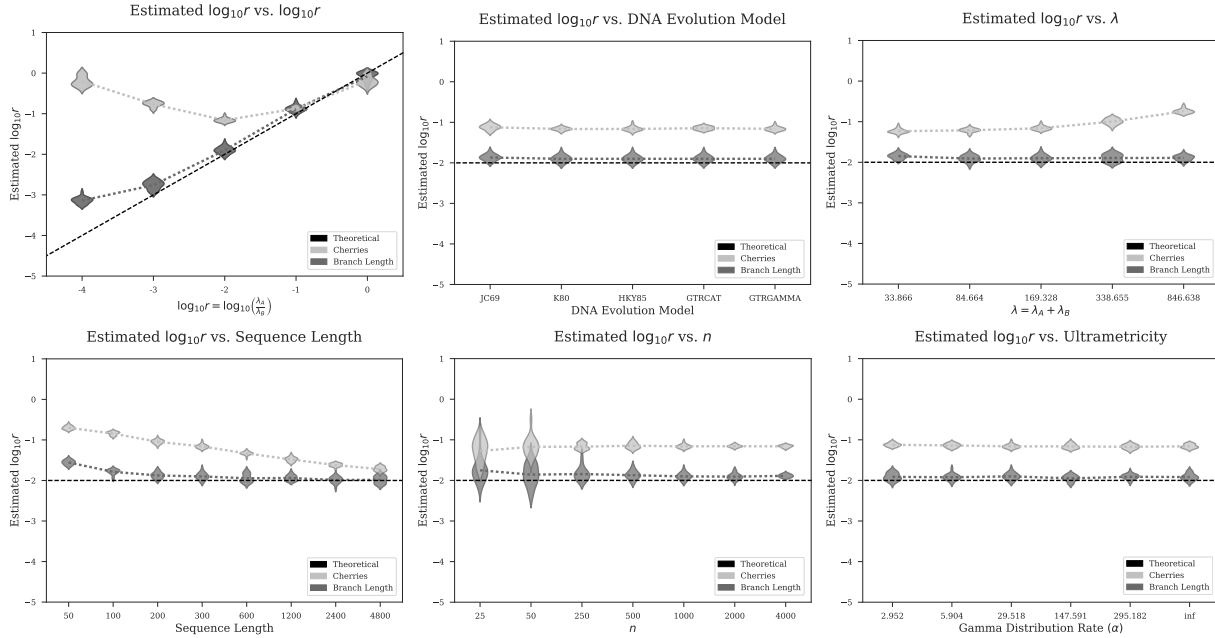
**Figure 2.3:** Tree inference error. Violin plots are shown for (a) the RF distance between true and estimated trees, and (b) mean branch lengths and mean pendant branch lengths computed for each tree (the dashed lines show the theoretical and conjectured expectations, and the dots show empirical averages). Note that FastTree 2 does not implement the K80 and Hasegawa, Kishino, and Yano (1985) (HKY85) models.

similarly had minimal impact on accuracy (Fig. 2.3a, upper-center). Note that the GTR+ $\Gamma$  model was used for simulation (with parameters given in the supplement); thus, all other results include model misspecification.

To make sure the difficulty in correctly resolving unbalanced trees is not simply due to insufficient search in ML tools, we compared likelihood scores of inferred trees and their corresponding true trees (Fig. B.9). Two interesting patterns were observed. The RAxML tree consistently had better scores than the true tree, indicating that lack of accuracy was not simply due to insufficient search. The difference in log-likelihood scores narrowed as  $r$  increased. These patterns are consistent with the explanation that likelihood scores computed on limited data are progressively less predictive of tree accuracy as the trees become less balanced. It is well-known that trees that include a mix of long and short branches, or generally, high heterogeneity of branch length, are hard to estimate, even in a likelihood framework [132, 133]. Decreasing  $r$  increased branch length heterogeneity in our dataset (Fig. B.10); the increased heterogeneity may be a cause of the large number of sites required for accurate estimation using maximum likelihood.

Unlike tree topology, the estimated average branch length and average terminal branch length were relatively accurate and robust to the parameters choice (Fig. 2.3b). However, two interesting and related patterns should be noticed. For  $r = 10^{-4}$ , both the terminal and overall branch lengths had slightly lower empirical means compared to the theoretical results or the conjecture (Fig. 2.3b, upper-left). This may partially be due to the fact that our theoretical results/conjectures are asymptotic in  $n$ , so the estimators may be biased for limited  $n$ . Consistent with this explanation, we observed that for  $r = 10^{-2}$ , with small  $n$ , empirical branch length averages were consistently lower than the theoretical values, but that they gradually increased and reached the theoretical expectations around  $n = 500$  (Fig. 2.3b, lower-center). The required  $n$  for the asymptotic expectations to be accurate will likely depend on  $r$ , and  $r$  values in the  $10^{-4}$  range likely require  $n > 1000$ .

Overall, RAxML consistently outperformed FastTree 2 with respect to tree accuracy by a



**Figure 2.4:** Parameter estimation accuracy. Violin plots are shown for the estimated  $r$ , using the cherry-based estimator and the branch-length-based estimator, for each of the experiments. True values are shown as dashed black lines.

small margin.

### Accuracy of $\hat{r}$

We focus on RAxML trees here, but note that FastTree trees give similar results (Fig. B.11). We start with the cherry-based  $r$  estimator. Unlike estimates based on true trees that were highly accurate (Fig. B.11), when trees inferred from sequence data are used, the cherry-based estimator is often not accurate (Fig. 2.4). When  $r = 1$ , estimates from cherry fraction are close to true values. However, for small  $r$ , the cherry fraction can be dramatically overestimated (Fig. B.12), and as a result, the estimated  $r$  can be orders of magnitude larger than the true value (Fig. 2.4). For example, when the true value of  $r$  is  $10^{-4}$ , RAxML inferred around 30% cherry fraction (i.e.,  $\hat{r} \approx 1$ ) instead of 0.99% (Fig. B.12). Since on true trees the estimator works very well (Fig. B.11), the overestimation is clearly due to tree inference error. Consistent with this explanation, as the length of the sequence increases, the cherry fraction and  $\hat{r}$  gradually converge to the true values;



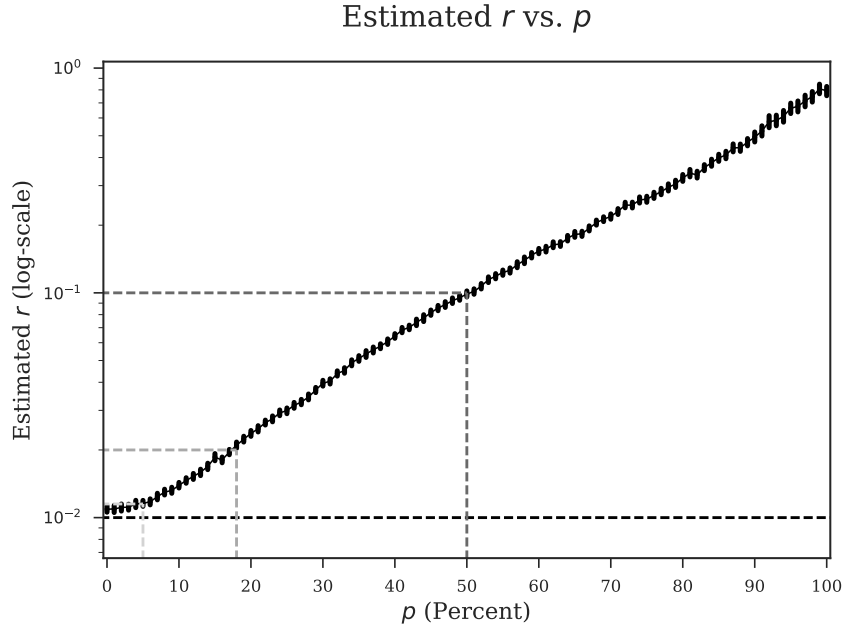
with 4,800 sites, the estimated  $r$  is within 27% of the true value (Fig. 2.4, bottom-left).

Unlike the cherry-based estimates, the length-based estimates of  $r$  were generally quite accurate (Fig. 2.4). While the estimator showed patterns that indicated it may be biased, it gave reasonable estimates of  $r$  for most conditions. The length-based method tended to slightly overestimate  $r$  in most conditions, but the overestimation was substantial only for  $r = 10^{-4}$ ; even for this most difficult case, however, estimates were still within one order of magnitude from the true value. Also, when sequences were extremely short (50 bp),  $r$  was substantially overestimated but was still within an order of magnitude of the true  $r$ . Even though the estimator is based on asymptotic results on  $n$ , reducing  $n$  to small values still maintained relatively high accuracy; only at  $n \leq 50$  did the variance of the estimator start to increase such that distributions of the estimate spanned more than an order of magnitude. Overall, the estimates are in the correct order of magnitude for most conditions, and are especially accurate for  $10^{-3} \leq r \leq 1$  for  $k = 300$ , and this range is even wider for larger  $k$ .

### 2.3.2 Simulations: Model Violations

Our results so far were based on trees that completely followed the dual-birth model (save for the enforced divergences from the ultrametricity). We now explore the performance under conditions in which the model generating the true tree diverges from our model. Specifically, we explore the following model: instead of forcing each node to have one child with rate  $\lambda_a$  and another with rate  $\lambda_b$ , with some small probability  $p$ , we allow both children to be active right away and thus have the rate  $\lambda_b$ . Setting  $p = 0$  recaptures the dual-birth model, but increasing  $p$  gradually introduces more model violations;  $p = 1$  simply gives the Yule model.

We performed simulations in which we used the default experiment parameters (Table 2.1) but varied  $p$  from 0 to 1. As expected, the error in  $\hat{r}$  increases as  $p$  approaches 1 (Fig. 2.5). The length-based estimator was robust to relatively low levels of model violation. For example, with  $p = 0.05$ , the average estimated  $r$  was 0.0116, which is very close to the true value of 0.01.



**Figure 2.5:** Model violations. Length-based estimates of  $r$  vs.  $p$ , the probability that both children of a given branch are active (i.e., have rate  $\lambda_b$ ) based on 20 replicates of simulations per  $p \in [0, 1]$  with  $n = 1000$ ,  $r = 10^{-2}$ ,  $\lambda = 169.328$ . Dashed light gray line:  $p = 0.05$ ; dashed medium gray line:  $\hat{r} = 2 \times r$ ; dashed dark gray line:  $\hat{r} = 10 \times r$ .

Further increasing  $p$  up to 0.17, the average estimate of  $r$  remained within two times the true value; errors reached an order of magnitude only at  $p = \frac{1}{2}$ . Interestingly, in log-scale, there was a somewhat linear relationship between  $p$  and  $\hat{r}$ .

### 2.3.3 Human *Alu* Analysis

We study two questions: How many *Alu* elements are active? At what rates do inactive *Alu* elements become active and active elements propagate? Assuming that *Alu* evolution has followed the dual-birth model, we use the length-based estimator (Eq. 2.9) to estimate the parameters shown in Table 2.2 from the full *Alu* dataset as well as from replicate subsampled data. The parameter  $r$  is estimated to be  $0.006 \approx 10^{-2.2}$  using the complete dataset. The estimated  $r$  gradually decreases with subsampled datasets, and with 1,000 sequences,  $r$  is estimated to be  $0.0034 \approx 10^{-2.5}$ . Note that these changes remain well within an order of magnitude.

**Table 2.2:** Results on *Alu*

<b>Sampling</b> <sup>†</sup>	1,000	10,000	100,000	885,011*	885,011 <sup>+</sup>
$n_r$	$0.055 \pm 0.001$	$0.060 \pm 0.000$	$0.072 \pm 0.000$	0.072	0.072
$\hat{r}$	$0.0034 \pm 0.0001$	$0.0040 \pm 0.0001$	$0.0059 \pm 0.0000$	0.0060	0.0060
$\lambda$	$127.64 \pm 2.32$	$124.35 \pm 1.24$	$111.70 \pm 0.56$	118.23	122.76
$\lambda_a$	$0.44 \pm 0.01$	$0.50 \pm 0.00$	$0.66 \pm 0.00$	0.70	0.73
$\lambda_b$	$127.21 \pm 2.32$	$123.85 \pm 1.24$	$111.04 \pm 0.56$	117.53	122.03
$D$	$0.0671 \pm 0.0007$	$0.0636 \pm 0.0004$	$0.0585 \pm 0.0002$	0.0550	0.0531
$L$	$0.1206 \pm 0.0014$	$0.1133 \pm 0.0007$	$0.1019 \pm 0.0004$	0.0958	0.0924

<sup>†</sup>Rows: Sampling (number of taxa), estimated portion of active *Alus*, model parameters ( $r$ ,  $\lambda$ ,  $\lambda_a$ ,  $\lambda_b$ ), mean branch length, and mean terminal branch length. The last two column are for the full final dataset (\*FastTree 2 and <sup>+</sup>RAXML). All other columns are the average of 20 subsampling replicates with the given number of taxa.

Based on the full dataset, we estimate the percentage of active *Alu* elements (Eq. 2.5) to be approximately 7.2% if either FastTree 2 or RAXML is used. Recall that an element is active if it has ever propagated. Thus, we estimate that 7% of *Alu* repeats have propagated at least once. Progressively reducing the number of sequences consistently reduces the estimated number of active elements, but it never falls below 5.5%.

### Rate of Activation and Propagation

We can also estimate  $\lambda$  (using Eq. 2.6). The rates we infer (Table 2.2) are in the unit of expected mutations. To convert them to the unit of time, we use a simple approach that requires several approximations and assumptions. We use a linear-time implementation of midpoint rooting [134] to root our estimated tree and then compute the maximum root-to-tip distance, which is 1.270. Assuming a molecular clock (see Fig. B.13 for deviations), we assume that this value corresponds to approximately 65 million years since the origin of *Alu* repeats [54] and multiply our estimates by the ratio of the tree depth in mutation units to time units. The results are  $\lambda_a = 1.426 \times 10^{-8}$  activation events per year per inactive element and  $\lambda_b = 2.384 \times 10^{-6}$  propagation events per year per active element, meaning each *Alu* element becomes active with a rate of roughly once every 70 million years, and once active, it propagates with a rate of roughly

two and a half times every million years. Note that these rates are for each element, and the total rates are much higher. Also, note that these are rates of an exponential distribution, and thus, individual activation and propagation events may occur in much shorter or longer time frames.

## 2.4 Discussion

We start by comparing the dual-birth model to alternative models. We then discuss several important points regarding our model and its application to *Alu* repeats.

### 2.4.1 Comparison to Other Models

The beta-splitting model of Aldous (1996) is one of the earliest models to provide a way to control tree balance [111]. The model starts from a predetermined number of leaves and recursively divides the set of leaves into two sets; at each step, the number of leaves in each set is determined by draws from a parameterized distribution. Adjusting the parameter enables generating trees with varying levels of balance. Our model is distinct from beta-splitting in several ways. Beta-splitting, unlike our model, generates distributions over unordered tree shapes and also does not define branch lengths. Moreover, unlike our model or Yule that generate the tree by a natural Markov process, beta-splitting starts by deciding the final number of tips and thus does not have a clear biological interpretation (as Aldous noted).

The alpha model of Ford (2005) is parameterized by a single parameter  $\alpha \in [0, 1]$ , where  $\alpha = 0$  gives the Yule model,  $\alpha = 1/2$  gives the uniform distribution, and  $\alpha = 1$  gives a perfect caterpillar tree [113]. The alpha model starts with a single-leaf tree and iteratively adds a new leaf to the middle of an edge in the tree. terminal edges are given weight  $1 - \alpha$  and all other edges are given weight  $\alpha$ , and the edge to which a new leaf will be added is chosen via these weights. The alpha model, unlike our model, does not define branch lengths, similarly to beta-splitting, and also doesn't have a clear biological interpretation.

An improvement of the beta-splitting model is the Blum and François (2006) (BF) model [114]. The BF model has a root speciation rate  $\lambda$ , and for a given branch with speciation rate  $\kappa$ , one child branch has speciation rate  $p\kappa$  and the other has speciation rate  $(1-p)\kappa$ , where  $p$  is either a fixed constant or is randomly chosen from a symmetric distribution on  $[0, 1]$ . Because the rate of a given branch must equal the sum of the rates of its two children, as the tree becomes larger, rates will progressively shrink and branches will become longer (unlike the dual-birth model). Blum and François are not concerned with this property because only the tree topology matters to them. Doubling the rates of child branches in the BF model with a fixed  $p$  can maintain the overall rate and gives a model that Kirkpatrick and Slatkin first introduced [35].

Just like the dual-birth model, the Kirkpatrick and Slatkin (1993) (KS) model can be parameterized by  $\lambda$  and  $r$  (which they call  $x$ ) to produce a fixed ratio  $r$  between the left and right branch rates and can also produce unbalanced trees. However, a main difference remains. Consider rates of the leaves in a balanced four-taxon tree. In the dual-birth model, two terminal branches have the rate  $\lambda_a$  and two have the rate  $\lambda_b$ . However, in the KS model, two have the rate  $2\frac{\lambda_a\lambda_b}{\lambda}$ , one will have the rate  $2\frac{\lambda_a^2}{\lambda}$ , and the other will have the rate  $2\frac{\lambda_b^2}{\lambda}$ . In both models, the sum of the rates of terminals is  $2\lambda$ , but in the KS model, this total rate is distributed differently. For larger trees, the terminal rates become even more unevenly distributed, whereas in the dual-birth, we always have two rates at terminals, corresponding to our two states. There is no natural way in which the KS or BF models can be mapped to the “active” and “inactive” states. To our knowledge, the KS model has only been used to simulate unbalanced trees in order to test the power of tree balance metrics in detecting deviations from the Yule model.

Jones (2011) explores age-dependent models in which a species lives for some time and then either goes extinct or produces exactly two descendant species, where the ratio of extinctions to speciations is given by a fixed number  $\rho$  [115]. The probability that a species  $i$  lives for at least time  $t$  is given by a function  $S(t)$ . Note that the function  $S(t)$  is dependent on time and not state, whereas the probability that a given species  $i$  lives for at least time  $t$  under the dual-birth model is

dependent on state (*active* or *inactive*) and is independent of time.

Maddison *et al.* (2007) propose a two-state model, known as the BiSSE model, in which each state has its own birth and extinction rates, and entities can transition across the two states under specified transition rates [116]. A key distinction between the dual-birth model and the BiSSE model is that, under the BiSSE model, *both* children of a node inherit the parent's state, but under the dual-birth model, the two children must have different states. Thus, simply setting one state transition rate (active to inactive) to 0 in the BiSSE model does not produce the dual-birth model. Moreover, the BiSSE model assumes that state change and speciation are completely independent of one another, whereas in the dual-birth model, a state change from inactive to active must coincide with a birth event. The BiSSE model is designed to study the impact of traits on speciation processes, and therefore, the inheritance of the state (e.g. a trait) by both progenies is natural. However, it does not provide a clear advantage in the study of propagating elements like *Alu* elements, where one of the child branches is a continuation of the parent and the other is not.

Lambert and Stadler (2013) study a wide range of macroevolutionary models and determine which models lead to a uniform distribution on ranked tree shapes [119]. The dual-birth model we introduce is an example of a model in which the speciation rate depends on a fully-heritable trait (*active*) with asymmetric speciation: one child, *right*, is the “new” child and does not inherit the mother *active* trait at all, and the other child, *left*, corresponds to the mother and completely inherits the mother *active* trait. Based on results from Lambert and Stadler, the dual-birth model does not induce a uniform distribution on ranked tree shapes, a fact that will be corroborated by the probability distribution we derive for ranked tree shapes generated under the model (Eq. 2.2 and Fig. 2.1c).

Finally, Steel and McKenzie (2001) propose a two-state extension of the Yule process [112]. In their model, unlike ours, states are used to enable a birth rate that varies throughout a branch, increasing gradually as the branch becomes longer.

## 2.4.2 Properties of the Dual-Birth

### Statistical Properties of the Estimators

Based on Theorem 3, it is easy to prove that the cherry-based estimator (Eq. 2.8) is a statistically consistent estimator of  $r$  if  $n$  is allowed to grow infinitely and if the true phylogenetic tree is known. Alternatively, if the tree is inferred from sequence data under the true model using maximum likelihood, allowing both  $n$  and the alignment length to grow to infinity will render Equation 2.8 a statistically consistent estimator. For limited  $n$  and alignment length, this estimator is not necessarily unbiased; in fact, our simulations showed clear evidence of severe biases in the number of cherries in trees inferred from limited data, and hence biased estimates of  $r$ . Only with very large alignment lengths (e.g. 4,800) did our estimates of  $r$  start to become accurate using the number of cherries. Requiring such long alignments can often be problematic. For example, SINEs are typically no more than several hundred bases long, and any tree inferred from such short datasets is prone to high estimation error. This shortcoming motivated the design of the length-based estimator.

Since Equation 2.9 is a conjecture, the length-based estimator is not presently proven statistically consistent. If Conjecture 1 is ever proven correct, the estimator can be also be proven statistically consistent for increasing  $n$  and the correct phylogeny. The length-based estimator may be biased, especially for small  $n$ . Nevertheless, it seems to provide a relatively robust estimator in our wide-ranging simulations.

### Model Limitations

The dual-birth model can be improved in several ways. Most importantly, it can be imagined that, as an active element evolves, it can deactivate and lose propagation capability. This change of state from active to inactive is not possible in our current model. Modeling deactivation would enable the estimation of the number of elements that are active at any specific point in

time, including at the present time. As the model currently stands, the estimated number of active elements should be best interpreted as the number of elements that have been ever active. A related but distinct improvement is allowing deaths in addition to births. Moreover, the fact that all elements are born into an inactive state, have identical rate of activation at birth, and an identical rate of birth are all obvious limitations of the model.

### **Unsolved Questions**

While we derived equations for several distributions and expectations, many theoretical questions remain unanswered, including the following. Can the exponential time calculations of tree distributions be simplified using closed form formulas or more efficient algorithms (e.g. dynamic programming)? What is the probability distribution of the number of leaves in the left or right of a given node? Relatedly, what are the distributions of other statistics of tree shape [135, 136]? We computed the expected branch length, but we did not derive the exact distribution of branch lengths. Although we conjecture a formula for the expected length of terminal branches and demonstrate its accuracy via simulation, we have not proven its correctness. Further, the cherry-based approach to estimate  $r$  is often inaccurate because of the error-prone topology of inferred trees. It would be interesting to see if such estimates could be corrected by considering Bayesian distributions over the trees or by using branch bootstrap support.

A main application of tree shape models is to define the prior distribution in a Bayesian tree inference [31, 32]. The dual-birth model could be used for this purpose as well, and such an approach may help in addressing the issue of the low accuracy of inferred trees for very unbalanced trees. Intuitively, if  $r$  is estimated to be small, the unbalanced trees will be given a higher prior probability. While we have derived many of the required distributions, the practical application of the dual-birth model as a prior model requires further development. The main issue is that computing the probability of unordered trees requires iterating all orderings, which will not be practical for trees of even moderate size. It may be possible to develop clever



dynamic programming algorithms to speed up the computations. Further, the best choice for hyperparameters for  $r$  and  $\lambda$  also need to be explored. However, if these difficulties could be overcome, the Markov Chain Monte Carlo (MCMC) approach can also be used to estimate the  $r$  parameter, and such approaches may outperform our current estimators.

### 2.4.3 *Alu* Repeats

#### The $r \leq 1$ Assumption

We estimated  $r \approx 0.006$  and that approximately 7% of nodes are active. Note that a transposon model of *Alu* propagation corresponds to  $r \approx 1$ , where a new *Alu* is as active as existing ones, and in expectation, half the repeats have propagated at least once. Recall that  $r = x$  and  $r = \frac{1}{x}$  are indistinguishable for trees inferred from the data, so  $r \leq 1$  is an assumption. But note that  $r > 1$  would imply that, once an *Alu* has propagated, its rate of transposition *reduces*. Such a model is not one of the debated hypotheses and is not necessarily sensible: no reasonable scenario that we can imagine would reduce the rate of propagation after the first propagation, but would keep it constant afterwards. Thus,  $r > 1$  is dismissed *a priori* in our analyses. In situations where  $r > 1$  and  $r < 1$  both present reasonable hypotheses, our phylogenetic approach will not be able to distinguish between the two scenarios.

#### Accuracy

Our simulation results indicated that the  $r$  parameter can be estimated with relatively high accuracy in most cases. However, we note that the estimates are never quite exact and have a range that spans between half to a full order of magnitude (Fig. 2.4). Thus,  $r$  estimates should be treated as ballpark estimates and interpreted to give the right order of magnitude. Our estimate of  $r = 0.006 \approx 10^{-2.2}$ , therefore, should be interpreted as stating that, based on our model and our length-based estimator, there is a two to three orders of magnitude change between the rate

of propagation of active and inactive elements. In our simulations, the estimator has reduced accuracy for very low values of  $r$  (close to  $10^{-4}$ ) but estimates around  $10^{-2}$  are quite accurate, if slightly overestimated. As  $r$  changes between 0.001 and 0.01, our estimate of the active number of elements would change between 3% and 9%.

## Interpretation

We have no independent way of estimating the number of active elements from our dataset. Estimates of the number of active elements in the literature are wide and varied. For example, Price *et al.* (2004) used whole-genome *Alu* data to estimate the total number of active elements to have been *at least* 143 throughout the history of *Alu* elements [55]; Wang *et al.* (2006) used human polymorphism data to estimate the number of currently-active *Alu* elements to be at least 31 [137]; Wacholder and Pollock (2016) introduced a novel Bayesian transposable element ancestral reconstruction method and used it to estimate a lower-bound of 1,386 *Alu* elements to have ever been active [138]; Batzer and Deininger (2002) did not provide a specific estimate of the number of active elements, but they stated that “only a few human *Alu* elements, the so-called ‘master’ or source genes, seem to be retrotransposition competent” [54]. These wide variations are partially because mechanisms of propagation and spread are not fully understood. Moreover, these studies are looking for a strong evidence of transposition capability and do not rule out the possibility that others are able to propagate. For example, the 1,386 lower bound given by Wacholder and Pollock is based on the observation that these many distinct elements currently include a mutation that inactivates them, and hence, should have been created by those many active element [138]. Our estimates of 7% is higher than these values found in the literature, but we emphasize that our estimate is not a lower bound. Future work should validate these estimates using alternative approaches, perhaps by comparing various primate genomes or providing estimates for other species.

Whether or not a new *Alu* insertion survives to become dominant in a population depends

on many factors, including whether the element is under selective pressure. The dual-birth model is not trying to capture population-level heterogeneity nor specific causes of birth, death, or survival of elements. In other words, in our model, a birth event corresponds to a new repeat that has successfully spread through a population (either due to drift or by selection). Thus, our estimated rates of propagation should be interpreted in this light and not as the rate with which a new *Alu* element is inserted in individual members of the population.

## 2.5 Data Availability

Data available from the Dryad Digital Repository: <https://doi.org/10.5061/dryad.13n52>

Code available from the GitHub repository: [github.com/niemasd/Dual-Birth-Model](https://github.com/niemasd/Dual-Birth-Model)

## 2.6 Acknowledgements

This work was supported by National Science Foundation (NSF) [IIS-1565862 to S.M.]; and National Institutes of Health (NIH) subaward [5P30AI027767-28 to S.M. and N.M.]. We thank Prof. Pavel Pevzner for fruitful discussions, which provided the motivation for the approach.

Chapter 2, in full, is a reprint of the material as it appears in “A Two-State Model of Tree Evolution and its Applications to *Alu* Retrotransposition” (2017). Moshiri, Niema; Mirarab, Siavash, *Systematic Biology*, 67(3), 475-489. The dissertation author was the primary investigator and first author of this paper.

## **Chapter 3**

### **ProACT: Prioritization Using Ancestral**

### **Edge Lengths**

In HIV epidemics, the majority of the structure of the transmission network is dictated by just a few individuals. Public health intervention, such as ensuring people living with HIV adhere to ART and are continually virally-suppressed, can help control the spread of the virus. However, such intervention requires utilizing the limited public health resource allocations. As a result, the ability to determine which individuals are most at-risk of transmitting HIV could allow public health officials to focus their limited resources on these individuals. Molecular epidemiology suggests an approach: prioritizing people living with HIV based on patterns of transmission inferred from their sampled viral sequences. In this paper, we introduce ProACT (**P**rioritization using **A**n**C**es**T**ral edge lengths), a novel phylogenetic approach for prioritizing individuals living with HIV. ProACT uses a simple idea: ordering individuals by their terminal branch length in the phylogeny of their virus. In simulations and also on a dataset of HIV-1 subtype B *pol* sequences obtained in San Diego, we show that this simple strategy improves the effectiveness of prioritization compared to state-of-the-art methods that rely on monitoring the growth of transmission clusters defined based on genetic distance.

### 3.1 Introduction

The transmission of HIV resembles scale-free networks [66], in which the majority of the structure of the network is dictated by just a few individuals, a phenomenon likely resulting from the scale-free properties of sexual contacts and injection drug use along which HIV is transmitted [58, 139]. As a result, public health intervention may be more effective when targeted at people living with HIV who are more likely to grow the transmission network. However, the best method to target individuals for specific interventions remains an open question, and the best strategy will likely depend on the specific intervention planned.

A potential form of intervention aiming to reduce future transmissions is to target HIV-Positive Individuals ( $H^+$ Is). For example, ART suppresses the HIV virus in the majority of cases,

stops the progression of the disease, and prevents onward transmission to an uninfected sexual partner, provided the H<sup>+</sup>I continuously adheres to the treatment [92]. In addition to reducing risk of transmission at the molecular level, adherence to ART is associated with a reduction of risky behavior as well [140]. While the initiation of ART is routine (or even universal) in most advanced health care systems, not every case of ART initiation leads to a sustained suppression of the virus through time. H<sup>+</sup>Is who start ART but fail to sustain it or who are otherwise unsuppressed can still infect others. Thus, a possible intervention is to ensure known H<sup>+</sup>Is are kept on ART and are continually suppressed, a task that requires allocation of public health resources. If people at risk of losing their suppression could be predicted accurately, the public health system could focus their limited resources on these individuals, administering several types of interventions: followups to ensure sustenance of ART, increased testing to ensure suppression, and, if all else fails, offering PrEP to their sexual partners. However, these are all costly interventions and cannot be undertaken for every known H<sup>+</sup>I. Thus, a natural question surfaces: which individuals are most at-risk of transmitting HIV?

Predicting tendency for future transmissions is difficult and is fraught with danger if undertaken primarily based on demographic or behavioral traits. Molecular epidemics suggest an alternative method: prioritizing H<sup>+</sup>Is for intervention solely based on patterns of transmission inferred from HIV sequence data [141, 142, 143, 21, 103, 16, 66, 144]. The inference of transmission networks using phylogenetic or distance-based methods has been the subject of much research [64, 26, 24, 23]. However, in this work, instead of being concerned with inferring exact patterns of transmissions, we ask the following question: given molecular data from each Sampled HIV-Positive Individual (SH<sup>+</sup>I), presumably all with access to ART, which individuals are most at-risk of transmitting the virus?

Prioritizing care based on molecular epidemics has been studied recently. Wertheim *et al.* (2018) present a method for prioritizing SH<sup>+</sup>Is based on performing transmission clustering (i.e., grouping individuals with low viral genetic distance into “transmission clusters”) and ordering

clusters by growth rate [103]. On a large dataset from New York, they show that the approach is able to predict individuals who will have relatively larger numbers of transmission links in the near future. Moshiri *et al.* (2018) have studied the same question in simulations and have shown that monitoring cluster growth can be used for predicting future transmissions substantially better than a random guess, whether clusters are defined using genetic distances or using phylogenetic methods [145]. Most recently, Balaban *et al.* (2019) showed in simulations that using a cluster-monitoring approach similar to that of Wertheim *et al.* (2018) but defining clusters using a min-cut optimization problem gives a small but consistent improvement over defining clusters using genetic distances [25].

In this paper, we introduce a new method for ordering SH<sup>+</sup>Is based on their phylogenetic relationships. Instead of relying on clustering individuals and then ordering clusters based on their growth, we seek to order individuals without clustering and without reliance on parametric models. Instead, we seek to simply exploits patterns in the phylogeny, and in particular, in branch lengths.

## 3.2 New Approaches

ProACT (**P**rioritization using **A**n**C**es**T**ral edge lengths) takes as input the inferred phylogenetic relationships between sampled HIV viruses (e.g. from the *pol* region), rooted using an outgroup or clock-based methods (e.g. midpoint or MinVar-root [134]). ProACT simply orders SH<sup>+</sup>Is in order of incident branch length of their associated virus, and it breaks ties based on incident branch lengths of parent nodes, then those of grandparent nodes, etc. We first motivate the approach and then present a formal definition of the method.

We note that ProACT is motivated and tested in a context similar to the present day health care systems that enjoy enough resources to provide ART to all SH<sup>+</sup>Is (recall that we call a H<sup>+</sup>I a SH<sup>+</sup>I if their sequence is also sampled). Thus, each SH<sup>+</sup>I is assumed to be given ART at a time

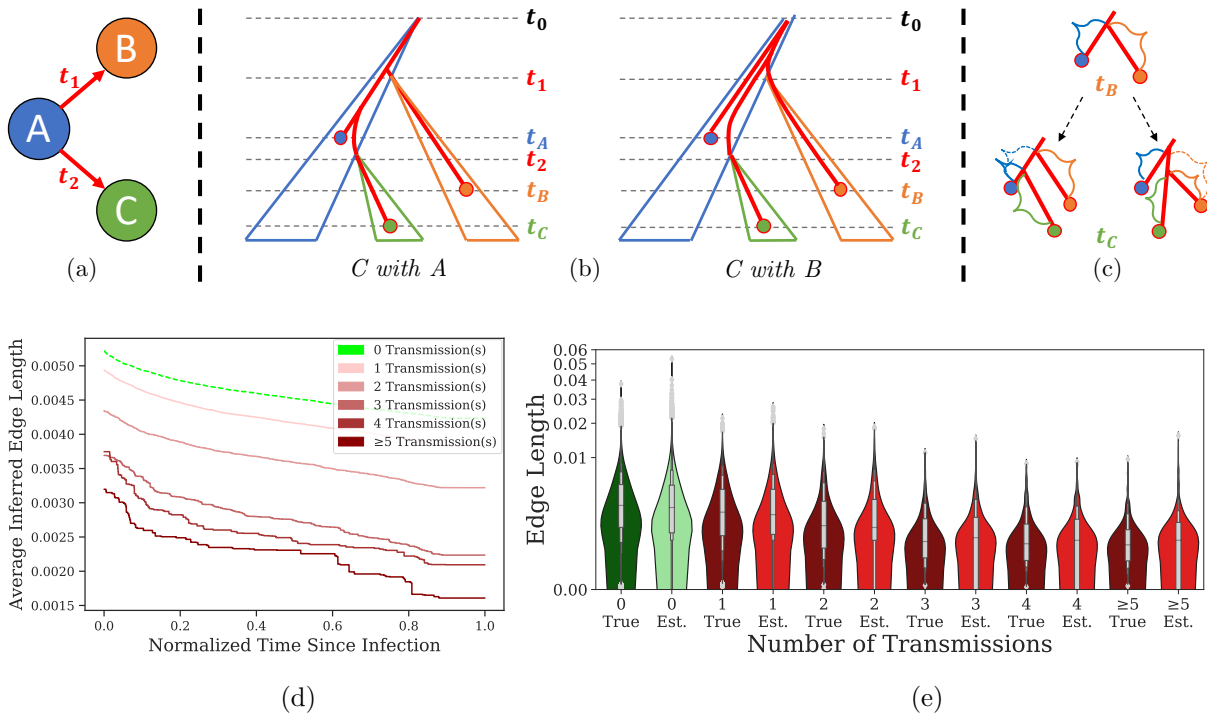
close to when their HIV is sequenced, but they may fail to be suppressed for the remainder of their life. These conditions describe the common practice of care in many advanced and (increasingly) developing countries.

### 3.2.1 Motivating the Approach

We start with the observation that, in simulations (described in detail below), when a phylogeny is inferred from sequences obtained at a given time point in an epidemic, the more a node transmits, the shorter its incident branch length tends to be (Figs. 3.1d–e and C.2). Using the Kendall’s tau-b test [146], in a ten-year epidemic simulation (details described below), we found a statistically significant anticorrelation between the incident branch lengths of individuals sampled within the first 9 years of the epidemic and the number of individuals they infected over the final year of the epidemic. This held for true ( $\tau = -0.0431$ ,  $p \ll 10^{-10}$ ) and inferred ( $\tau = -0.0354$ ,  $p \ll 10^{-10}$ ) phylogenetic trees. Though not obvious, this observation can be explained by the constraints placed upon the viral phylogeny by the transmission history (Fig. 3.1a–c).

In the context of HIV epidemiology in many advanced countries, SH<sup>+</sup>Is are typically sampled upon beginning ART. Let’s assume for simplicity that every individual in the given dataset has at some point initiated ART, meaning future transmissions by individuals in the dataset must happen only if the source stops ART or is otherwise unsuppressed. Given a viral phylogeny containing all known SH<sup>+</sup>Is, if, in the future, individual  $u$  in the dataset transmits to individual  $v$ , there are two possible scenarios regarding the placement of the leaf corresponding to  $v$  in the existing (true) phylogeny: (1)  $v$  is placed on the edge incident to  $u$ , so the edge incident to  $u$  will shorten, or (2)  $v$  is not placed on the edge incident to  $u$ , so the edge incident to  $u$  will remain the same length. Although Scenario 2 is possible, Scenario 1 is far more likely [147], and note that the terminal branch lengths do not increase in either scenario. Thus, as time goes by, the terminal branch can only shorten or stay fixed, and it will most often shorten because of new transmissions by the SH<sup>+</sup>I associated with that terminal branch. This pattern, easily observed in simulations





**Figure 3.1:** The effect of new transmissions on incident branch lengths. (a) Individual A transmits to individual B and C at times at  $t_1$  and  $t_2$ , respectively. (b) Viral samples are obtained from individuals A, B, and C at times  $t_A$ ,  $t_B$ , and  $t_C$ . The viral phylogeny of samples is constrained by each transmission event's bottleneck, and the most likely phylogeny matches the transmission history (Left), but in the less likely deeper coalescence, it may not match (Right). (c) Moving from the phylogeny observed at time  $t_B$  to the phylogeny at time  $t_C$ , the branch length incident to individual A shortens upon the addition of individual C in the likely event that the coalescence of the lineage from C with the lineage from A is more recent than its coalescence with the lineage from B (Left), or the branch length incident to individual A remains constant in the event of a less likely deeper coalescence (Right). Regardless, the length of the branch incident to individual A never increases. In simulation, we can observe this trend: as time progresses, the incident branch length of each individual tends to decrease, both in true (Fig. C.1) and inferred (d) phylogenies, and as the number of transmissions from a given individual increases, the distribution of incident edge length tends to decrease, both in true and inferred phylogenies, labeled "True" and "Est.," respectively (e).

(Fig. 3.1d), leads to shorter branches for SH<sup>+</sup>Is who have transmitted recently.

Note that SH<sup>+</sup>Is who transmit are unsuppressed. The first time they infect others, their terminal branch length is likely to decrease, and further transmissions further decrease their terminal branch lengths (Fig. 3.1d). Thus, one expects nodes with smaller incident branch length to be more likely to have transmitted since their sampling time. Moreover, they are also likely to transmit in the near future because they are likely not to be suppressed. The higher probability of a lack of suppression makes them a good candidate for intervention.

### 3.2.2 Formal Description

ProACT takes as input a *rooted* phylogenetic tree  $T$  of viral samples. Let  $bl(u)$  denote the incident branch length of node  $u$ , and assume the incident branch length of the root of  $T$  is 0. Let  $a(u)$  denote the vector of ancestors of node  $u$  (including  $u$ ), where  $a(u)_1$  is  $u$ ,  $a(u)_2$  is the parent of  $u$ ,  $a(u)_3$  is the grandparent of  $u$ , etc. Let  $r(u)$  denote the length of the path from node  $u$  to the root of  $T$ , i.e.,  $r(u) = \sum_{v \in a(u)} bl(v)$ . ProACT sorts the leaves of  $T$  in ascending order of  $bl(a(u)_1)$ , with ties broken by  $bl(a(u)_2)$ , then by  $bl(a(u)_3)$ , etc. Note that, for two leaves  $u$  and  $v$ ,  $|a(u)|$  may be less than  $|a(v)|$ , in which case, for all  $|a(u)| < i \leq |a(v)|$ ,  $\frac{r(u)}{|a(u)|-1}$  (i.e., average branch length along the path from  $u$  to the root of  $T$ ) is compared with  $bl(a(v)_i)$  instead. If two nodes are equal in all comparisons, if the user provides sample times, the earlier sample time is given higher priority; otherwise, ties are broken arbitrarily. Because sorting is needed, for a tree with  $n$  leaves, assuming branch lengths are fairly unique, the ProACT algorithm runs in  $O(n \log n)$  time. Scalable methods exist both for the inferring [102, 148] and rooting [134] very large trees.

## 3.3 Results

We evaluate ProACT on simulated and real data.

**Table 3.1:** Varied HIV simulation parameters. Values for the base model condition are shown in bold.

<b>Parameter</b>	<b>Values</b>
ART Start Rate ( $\lambda_+$ , year <sup>-1</sup> )	<b>1</b> , 2, 4
ART Stop Rate ( $\lambda_-$ , year <sup>-1</sup> )	0.12 (0.25x), 0.24 (0.5x), <b>0.48 (1x)</b> , 0.96 (2x), 1.92 (4x)
Expected Degree ( $E_d$ )	<b>10</b> , 20, 30

### 3.3.1 Simulation Results

In order to test ProACT’s efficacy, we performed a series of simulation experiments in which we used FAVITES [145] to generate a sexual contact network, transmission network, viral phylogeny, and viral sequences emulating HIV transmission in San Diego from 2005 to 2014 (Material and Methods). We have simulated nine model conditions (Table 3.1) by starting from a base model condition and varying the rate of ART initiation ( $\lambda_+$ ), rate of ART termination ( $\lambda_-$ ), and the expected degree of the sexual network ( $E_d$ ). We subsequently inferred and rooted a phylogeny of all sequences obtained during the first 9 years of the simulation. Then, ProACT was run on the true and inferred full trees and subsampled trees.

To measure the efficacy of a given prioritization, we compute the Cumulative Moving Average (CMA) of the number of infections caused by the top individuals in the prioritization during the tenth year of the simulation (our outcome measure). The higher the CMA for the top individuals in a prioritization, the higher the number of future transmissions from these *top* individuals. Sorting individuals by their outcome measure (known to us in simulations) enables us to compute the optimal CMA curve. Also, the mean number of transmissions gives us the expected value of the CMA for a random prioritization. Across experimental conditions, the maximum and random expectation vary, so to enable proper comparison of *effects of prioritization* across conditions, we also report an adjusted CMA normalizing above the random prioritization and over the optimal prioritization (Eq. 3.1; see Materials and Methods). Thus, for this Adjusted Transmissions/Person metric, 1 indicates the optimal ordering and 0 indicates ordering that is no

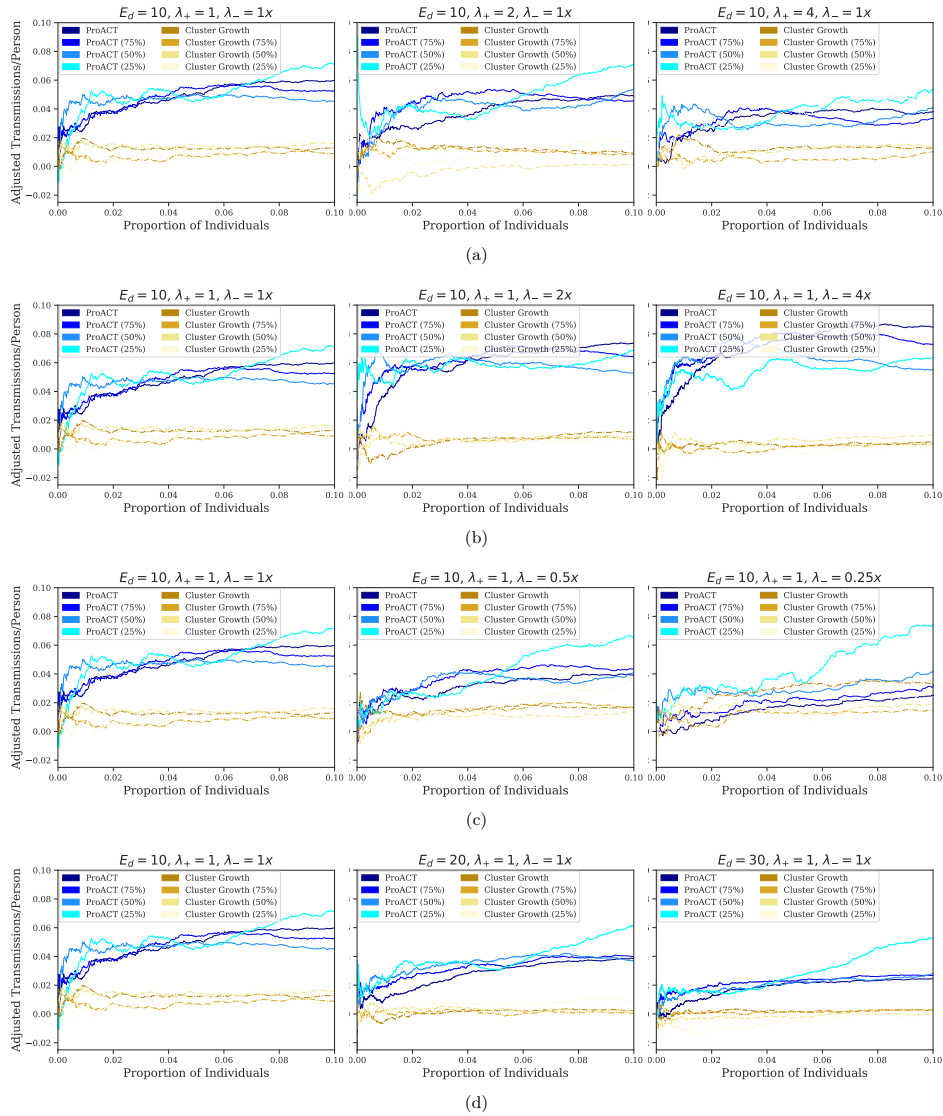
better than random.

### **ProACT Outperforms Random Prioritization Across Conditions**

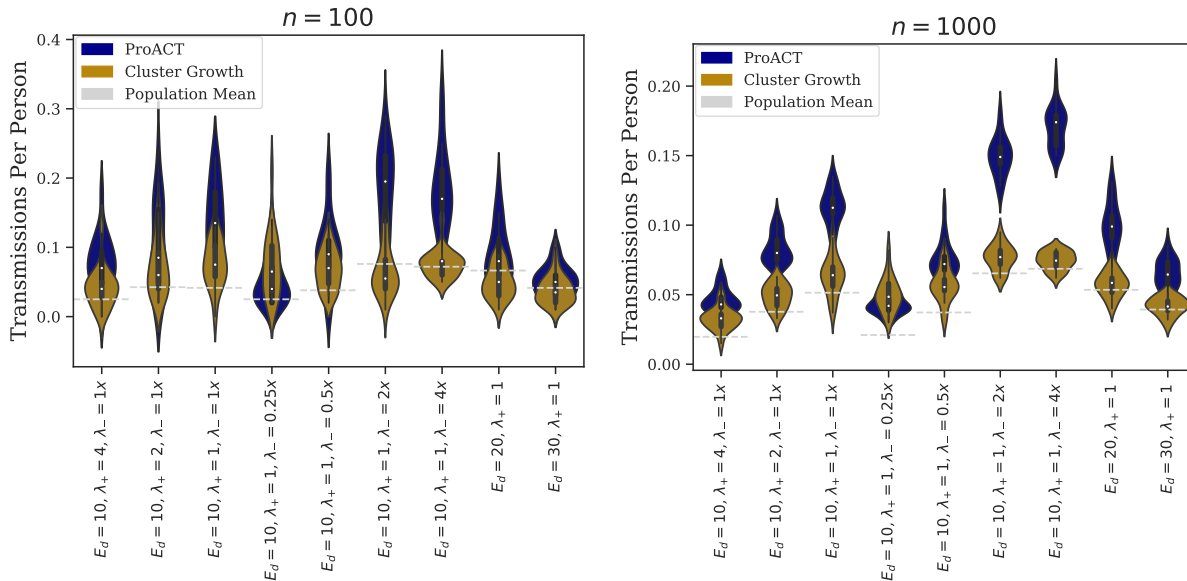
Across all experimental parameters, ProACT performed much better than one would expect from a random ordering (Fig. 3.2). As we increased the proportion of top individuals selected, ProACT's CMA initially increased (e.g. for up to 7% top individuals in the base model condition) and subsequently flattened out. The most clear signal for benefits of prioritization (e.g, a high CMA) is obtained for up to 10% top-priority individuals (though exact values depend on the model condition). As the number of selected individuals increases beyond 10%, however, because the metric of efficacy is CMA, the efficacy of a selection will eventually converge towards the efficacy of a random selection by definition (Fig. C.4).

### **ProACT Outperforms Cluster Growth**

As mentioned, Wertheim *et al.* (2018) present a method for prioritizing SH<sup>+</sup>Is by clustering individuals based on viral genetic distance, tracking the size of each cluster over time, and prioritizing clusters in descending order of the growth rate [103]. The approach can be easily extended to also order individuals (i.e., individuals belonging to clusters with high growth rates are prioritized higher; see Materials and Methods for details). ProACT consistently outperformed prioritization using cluster growth for various parameter choices (Figs. 3.2–3.3). The only exception was when the rate of stopping ART was lowered all the way to 0.25x, which corresponds to expected time of ART termination of 8.3 years. In this condition where adherence was at its highest, prioritization by cluster growth outperformed ProACT when using the full dataset.



**Figure 3.2:** ProACT performance on datasets simulated using FAVITES. CMA of adjusted number of transmissions per person across the first decile of prioritized SH<sup>+</sup>Is for each simulation parameter set. The horizontal axis depicts the quantile of highest-prioritized SH<sup>+</sup>Is (e.g.  $x = 0.01$  denotes the top percentile), and the vertical axis depicts their adjusted average number of transmissions per person (1 indicates the optimal ordering, and 0 indicates an ordering that is no better than random). In our simulations, we varied three parameters of interest: (a) the rate of ART initiation ( $\lambda_+$ ), (b-c) the rate of ART termination ( $\lambda_-$ ), and (d) the expected degree of the sexual network ( $E_d$ ). The simulations were 10 years in length, prioritization was performed 9 years into the simulation, and the adjusted average number of transmissions per person was computed during the last year of the simulation. The curves labeled “Cluster Growth” denote prioritization by inferring transmission clusters using HIV-TRACE [26] at year 9 of the simulation and sorting clusters in descending order of growth rate since year 8. The curves labeled with percentages denote subsampled datasets. All curves were calculated using 20 simulation replicates.



**Figure 3.3:** Efficacy on datasets simulated using FAVITES. Average of the raw number of transmissions per person for the top  $n$  individuals in a prioritized list vs. simulation parameter set across various values of  $n$ . The violin plots depicted are across 20 replicates and contain box plots with distribution medians shown as white dots and distribution means shown as dashed grey lines.

### Impact of Simulation Parameters

As the rate of stopping ART ( $\lambda_-$ ) increased (i.e., with lower adherence), the gap between ProACT and cluster growth grows. For example, the mean number of transmissions per person among the top 1,000 individuals chosen using ProACT and cluster growth were respectively 0.1702 and 0.0745 (a 1.28x improvement) for the condition with  $\lambda_- = 4x$ . This 1.28x improvement gradually decreases to 0.95x, 0.78x, 0.31x, and -0.15x as we reduce the rate or ART termination to 2x, 1x, 0.5x, and 0.25x. As  $\lambda_-$  decreased, ProACT’s performance compared to optimal ordering tended to decrease, whereas cluster growth’s performance compared to optimal ordering tended to increase; however, ProACT continued to outperform cluster growth for all but the  $\lambda_- = 0.25x$  condition (Fig. 3.2b–c).

As the rate of starting ART ( $\lambda_+$ ) increased (i.e., with faster diagnoses), the performance of ProACT compared to optimal ordering very slightly degrades (Fig. 3.2a). As a result, the

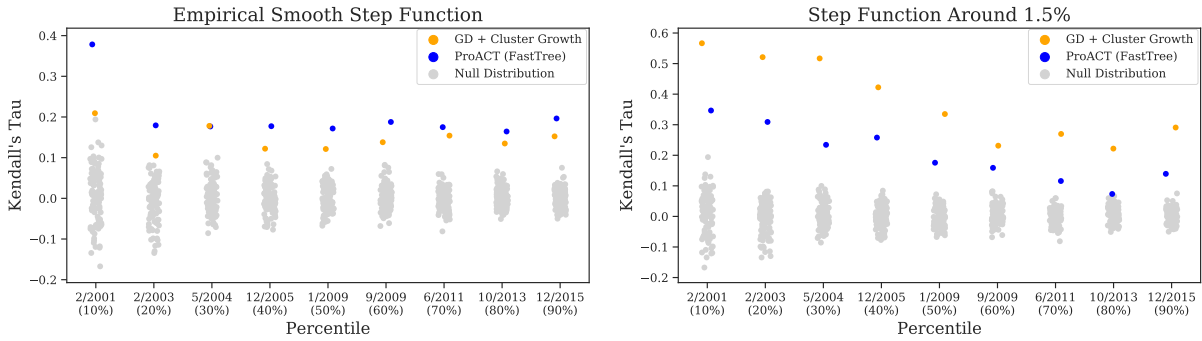
gap between ProACT and cluster growth decreases slightly: when observing the mean number of transmissions per person among the top 1,000 individuals chosen by each method, ProACT experiences a 0.78x, 0.70x, and 0.37x improvement over cluster growth when  $\lambda_+$  is set to 1x, 2x, and 4x, respectively. Note that as expected, increasing  $\lambda_+$  reduced the raw number of new infections caused per capita (Fig. C.3) overall and among top-priority individuals (Fig. 3.3).

Effects of the expected number of sexual contacts per person ( $E_d$ ), which controls the speed of spread is also interesting (Figs. 3.2d and 3.3). As  $E_d$  increased, the efficacy of both approaches decreased, but ProACT continued to consistently perform many times better than cluster growth.

### **Impact of Incomplete Sampling**

Subsampling the total dataset to include  $3/4$ ,  $1/2$ , or  $1/4$  of the total population of SH<sup>+</sup>Is did not have a major impact on the performance of ProACT compared to the optimal ordering (Fig. 3.2). Inevitably, the raw number of new infections decreased as the dataset was subsampled (Fig. C.3). However, what remained relatively constant was the benefit of ProACT and cluster growth with respect to optimal and random ordering (e.g. the adjusted metric).

Despite the general robustness, some interesting effects were observed. With  $\lambda_+ = 2x$ , ProACT's performance remained quite similar across all levels of subsampling, whereas prioritization by cluster growth was negatively impacted by less sampling, especially at the  $1/4$  level (Fig. 3.2a). Interestingly, for  $\lambda_- < 1x$ , ProACT's performance on  $1/4$  sampled datasets *improved* relative to more complete sampling. However, the efficacy of prioritization by cluster growth remained fairly consistent for  $\lambda_- < 1x$  (Fig. 3.2b–c). Similarly, the performance of ProACT compared to optimal ordering improved with  $1/4$  sampled datasets when sexual contact degree increased to  $E_d \geq 20$  (Fig. 3.2d).



**Figure 3.4:** Kendall’s tau-b test results for ProACT ordering on real data using two riskiness score functions: an empirical smooth step function and a strict step function around 1.5%. The full San Diego dataset was split into two sets (*pre* and *post*) at each decile (shown on the horizontal axis). The individuals in *pre* were ordered using ProACT and by cluster growth, and they were given a riskiness “score” computed using a riskiness score function (see Materials and Methods). Kendall’s tau-b correlation coefficient was computed for each ordering with respect to the optimal possible ordering (i.e., sorting in descending order of riskiness score). The null distribution was visualized by randomly shuffling the individuals in *pre*, and test *p*-values are shown in Table 3.2.

### 3.3.2 Real San Diego Dataset

We next analyzed a dataset of 926 HIV-1 subtype B *pol* sequences obtained in San Diego between 1996 and 2018. To evaluate ProACT accuracy, we divided the data into deciles, with each decile defining two sets: *past* (sequences up to the decile) and *future* (sequences after the decile). We inferred a phylogeny from the sequences present in the *past* set using FastTree 2 [102], and we used ProACT to order all SH<sup>+</sup>Is in this set. We then evaluated how the outcome measure correlates with the position of each individual in the ordering. We quantify the correlation using Kendall’s tau-b, a rank correlation coefficient adjusted for ties [146]. Values range between -1 and 1, with -1 signifying perfect inversion, 1 signifying perfect agreement, and 0 signifying the absence of association.

On real datasets, unlike the simulated data, the desired outcome measure, the number of new transmissions per person, is not known. Instead, we have to use inferred relationships. HIV-TRACE (used in our cluster growth approach) defines a pair of SH<sup>+</sup>Is as “genetically linked” if their sequences are very similar (TN93 distance below 1.5%). We similarly use the TN93



sequence similarity as an outcome measure, but in addition to using a fixed threshold, we also use smoother functions (Fig. C.5). We measure the number of linked individuals using a step function (1 if TN93 distance is below 1.5% and 0 otherwise) and an empirical smooth step function determined by fitting a mixture of three Gaussians to the distribution of pairwise TN93 distances (Material and Methods). We also explore an analytical smooth step function (parameterized sigmoid). Note that, when the step function is used, our outcome measure (computed for future transmissions) is exactly the same as what the cluster growth method uses for prioritizing (albeit, using past data). Thus, it is reasonable to expect the step function will favor cluster growth. As we move to smoother functions of distance to count genetic links, our measure is expected to become less biased in favor of HIV-TRACE.

Using both ProACT and cluster growth to prioritize individuals results in orderings of individuals with positive Kendall's tau-b correlations to the number of future genetic links regardless of the time (i.e., decile) and the function used to count genetic links (Fig. 3.4). These correlations are statistically significant in almost all cases (Table 3.2 and Fig. 3.4). The correlation coefficient ranges between 0.4 (ProACT; 10% time) and 0.1 (cluster growth; 20% time) for empirical function, and between 0.6 (cluster growth; 10% time) and 0.1 (ProACT; 80% time) for the step function.

The comparison between ProACT and cluster growth depends on the choice of the function to count links. When counting the number of links using the step function, prioritization by cluster growth consistently outperforms ProACT for all deciles of the dataset. These results are not surprising, given that we count HIV-TRACE links both to prioritize and to evaluate. However, according to the empirical smooth step function learned from the TN93 distances, ProACT outperforms cluster growth in all except one time point, where they are tied.

To further test whether the smoothness of the link-counting function applied to TN93 distances is a factor in deciding the relative accuracy of methods, we used a sigmoid function to replace the step function while keeping the inflection point at 1.5% (Fig. C.5). We observed

**Table 3.2:** Kendall’s tau-b test for a null hypothesis that a given prioritization yields a total outcome measure no better than random. We show  $p$ -values for the real San Diego dataset for the first through ninth deciles using two outcome measure functions. Tests that failed to reject the null hypothesis with (uncorrected)  $p$ -value  $< 0.00138$  (corresponding to  $\alpha = 0.05$  with a Bonferroni multiple hypothesis testing correct with  $n = 36$ ) are marked with †.

Empirical Smooth Step Function

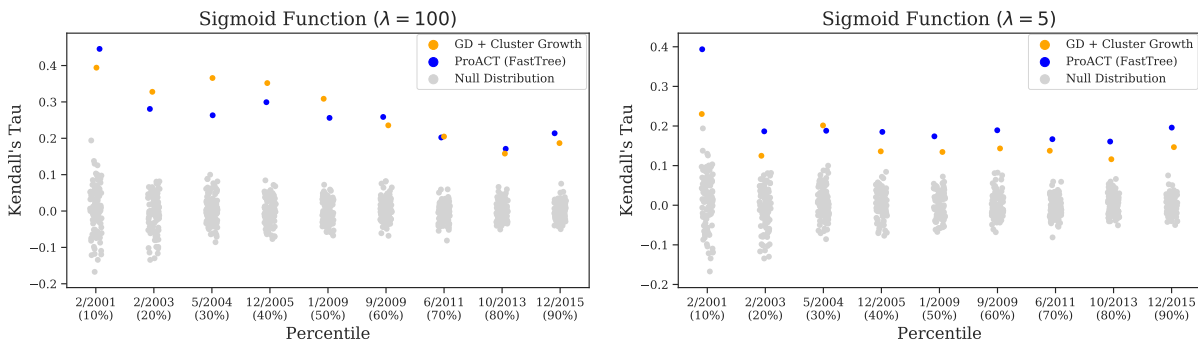
Percentile	GD + Cluster Growth	ProACT (FastTree)
10%	† $2 \times 10^{-3}$	$5 \times 10^{-8}$
20%	† $2 \times 10^{-2}$	$1 \times 10^{-4}$
30%	$5 \times 10^{-6}$	$6 \times 10^{-6}$
40%	$2 \times 10^{-4}$	$2 \times 10^{-7}$
50%	$5 \times 10^{-5}$	$2 \times 10^{-8}$
60%	$6 \times 10^{-7}$	$2 \times 10^{-11}$
70%	$2 \times 10^{-9}$	$1 \times 10^{-11}$
80%	$2 \times 10^{-8}$	$1 \times 10^{-11}$
90%	$2 \times 10^{-11}$	$1 \times 10^{-17}$

Step Function Around 1.5%

Percentile	GD + Cluster Growth	ProACT (FastTree)
10%	$4 \times 10^{-12}$	$1 \times 10^{-5}$
20%	$1 \times 10^{-19}$	$5 \times 10^{-8}$
30%	$3 \times 10^{-28}$	$3 \times 10^{-7}$
40%	$7 \times 10^{-25}$	$2 \times 10^{-10}$
50%	$2 \times 10^{-19}$	$1 \times 10^{-6}$
60%	$8 \times 10^{-12}$	$1 \times 10^{-6}$
70%	$1 \times 10^{-17}$	$1 \times 10^{-4}$
80%	$5 \times 10^{-14}$	† $7 \times 10^{-3}$
90%	$2 \times 10^{-25}$	$4 \times 10^{-7}$

that as the outcome measure function becomes more smooth, ProACT’s performance improves with respect to prioritization by cluster growth (Fig. 3.5, Table C.1). Based on the more smooth sigmoid function ( $\lambda = 5$ ), ProACT outperforms cluster growth in all but one case where they are tied. Thus, simply counting distances close to 1.5% as partial links leads to evaluations that favor ProACT.

As time increases, both methods experience seemingly downward trends in their tau coefficients, but the null distribution of tau coefficients also tightens (Fig. 3.4). Thus, both methods consistently do significantly better than expected by random chance and there is no clear



**Figure 3.5:** Kendall’s tau-b test results for ProACT ordering on real data using the sigmoid riskiness score functions with  $\lambda = 100$  and  $\lambda = 5$ . The full San Diego dataset was split into two sets (*pre* and *post*) at each decile (shown on the horizontal axis). The individuals in *pre* were ordered using ProACT and by cluster growth, and they were given a riskiness “score” computed using a riskiness score function (see Materials and Methods). Kendall’s tau-b correlation coefficient was computed for each ordering with respect to the optimal possible ordering (i.e., sorting in descending order of riskiness score). The null distribution was visualized by randomly shuffling the individuals in *pre*, and test *p*-values are shown in Table C.1.

relationship between *p*-values of individual tool and time (Table 3.2). However, both for the step function and the sigmoid functions, ProACT’s relative performance with respect to cluster growth tends to improved over time.

### 3.4 Discussion

We start by discussing observed results and then comment on practical implications of this paper both for public health and for future research in molecular epidemics.

#### 3.4.1 Discussion of Results

In our simulations, ProACT was least effective in conditions with very low rate of ART termination ( $\lambda_-$ ) which correspond to very high adherence. As expected, the total number of new infections originated from SH<sup>+</sup>Is is low when adherence is high (Fig. C.3) and neither method is much better than random clustering. This observation is consistent with the motivation we presented for the ProACT algorithm. Recall that the motivation relied on identifying SH<sup>+</sup>Is who

have stopped being suppressed. If all known SH<sup>+</sup>Is have been started on treatment and none ever stops treatment, prioritization loses its practical relevance, and relatedly, ProACT loses its statistical power. We saw a similar effects when we increased the rate of ART ( $\lambda_+$ ), which is also not surprising as increasing  $\lambda_+$  is in effect similar to reducing  $\lambda_-$ .

When we reduced sampling, we did not observe reductions in effectiveness of ProACT and occasionally even observed improvements. These results have to be interpreted in the context of our adjusted metric, which measures benefits over random and below optimal ordering. The per capita number of new infections from high-priority SH<sup>+</sup>Is was *lowered* when we subsampled the datasets (Fig. C.3). Thus, as expected, when some SH<sup>+</sup>Is are missing from the dataset available to a particular analysis, the overall effectiveness of identifying top priority SH<sup>+</sup>Is reduces. However, the effectiveness reduces equally for the optimal ordering and the ProACT method is not impacted any worse than optimal ordering is. In fact, ProACT is in some cases impacted a bit less harshly than optimal ordering, hence the improvements in adjusted outcome with  $1/4$  sampling. One should also keep in mind that choosing  $x\%$  highest priority individuals from the full datasets results in  $4x$  as many individuals as choosing the top  $x\%$  of the  $1/4$  subsampled dataset.

The reader is reminded that SH<sup>+</sup>Is are H<sup>+</sup>Is who are also diagnosed, and in our model, are immediately sequenced and put on ART (which they may or may not sustain). Thus, *full sampling* refers to a case where all *diagnosed* individuals are included in the dataset and H<sup>+</sup>Is who are not diagnosed are never in our sampling. In other words, the full sampling case should not be misunderstood as including undiagnosed people. Rather, lack of full sampling corresponds to a case where some SH<sup>+</sup>Is are known to *some* clinic but are not included in the study, perhaps due to a lack of sequencing or data sharing.

ProACT far outperformed random ordering and also ordering by cluster growth in simulations. However, we note that, despite the strong performance, there is much room left for future improvement: ProACT consistently ranges in its outcome measure between 2% and 10% of the theoretically optimal efficacy when selecting up to 10% of top-priority SH<sup>+</sup>Is. Thus, there

is great room for improvement in identifying high-value individuals compared to our method according to the simulation results. It will be unrealistic to expect that any statistical method based solely on sequence data (and perhaps also commonly available metadata, e.g. sampling times) will be able to come close to the optimal ordering. Nevertheless, it remains likely that methods better than ProACT could in fact be developed.

### 3.4.2 Implications of Results

In this paper, in addition to introducing ProACT, we formalized a useful approach for thinking about the effectiveness of public health intervention in molecular epidemics. Instead of focusing on the accuracy of methods of reconstructing phylogenetic trees or transmission networks, a question fraught with difficulties, we asked a more practical question. Given molecular epidemic data, can the methods, whether phylogenetic or clustering-based, prioritize SH<sup>+</sup>Is for increased attention by public health? The idea of using molecular epidemics for prioritization is of course not a new idea. For example, as we mentioned, Wertheim *et al.* (2018) presented a method to prioritize SH<sup>+</sup>Is based on the growth rates of their transmission clusters [103]. Vasylyeva *et al.* (2018) performed a phylogeographic analysis to reconstruct HIV movement among different locations in Ukraine in order to infer region-level risk prioritization [149]. Much earlier even, Mellors *et al.* (1996) predicted HIV patient prognosis by quantifying HIV RNA in plasma [150]; predicted prognosis can subsequently be used as a prioritization rank. However, we hope that our formal definition of the problem as a computational question (i.e., prioritization), in addition to our extensive simulations and developed metrics of evaluation will stir further work in this area. As stated before, it seems likely that more advanced methods than our simple prioritization approach can improve performance beyond ProACT in the future.

ProACT prioritizes individuals, not clusters. Prioritizing treatment followup or partner tracing for individuals based on their perceived risk of future transmission promises to be perhaps more effective than targeting clusters. However, such targeted approaches also pose ethical

questions that have to be considered. For example, we may not want the algorithm to be biased towards particular demographic attributes. ProACT does not use *any* metadata in its prioritization, reducing risks of such biases. It simply uses the viral phylogeny, which, compared to other types of data, may lead to fewer biases. Nevertheless, it is possible that factors such as the depth of the sampling of a demographic group can in fact change branch length patterns in the phylogeny and make ProACT less or more effective for certain demographic groups. These broader implications of individual prioritization and impacts of demographics on the performance of ProACT should be studied more carefully in future.

One may wonder whether ordering by branch lengths will result in orderings that fail to change with time and reflect the changes in the epidemic. To answer this question, on the San Diego Primary Infection Resource Consortium (PIRC) data, we asked how fast the ProACT ordering changes as time progresses. To do so, we computed Kendall's tau-b correlations to the ProACT ordering obtained using only the first decile of the dataset (Fig. C.6). There was a strong but diminishing correlation with the initial ordering. The correlations started at 1 (as expected) and gradually decreased in the ninth decile to 0.522. The results show that as desired, ProACT orders do in fact change with time, albeit gradually. The gradual change implies that certain individuals remain high-priority as time progresses. In practical use, ProACT ordering should be combined with clinical knowledge about the status of individual patients. For example, high priority individuals according to ProACT can be given lower priority if they manage to constantly remain suppressed with multiple followups. More broadly, the ProACT ordering should be considered one more tool for prioritizing clinical care, but valuable clinical knowledge, not incorporated into the algorithm, should also be exploited.

Finally, a question faced by public health officials is whether the cost of targeting diagnosed individuals for followups and partner tracing is worth the reduction in future cases. The answer to that question will inevitably depend on who is targeted. For example, in our default simulation case, targeting individuals randomly would at most reduce 0.0529 transmissions per

chosen person in the next 12 months, whereas targeting top 1000 individuals according to ProACT would at most reduce 0.119 transmissions. Thus, prioritization can in fact change the cost-benefit analyses. Moreover, given a prioritization, one can use simulations to predict the outcome measure for the top  $x$  individuals (similar to Fig. 3.2) and use metrics such as Quality-Adjusted Life-Year (QALY) to estimate how many top individuals should be targeted for the cost to justify the benefits.

## 3.5 Materials and Methods

### 3.5.1 Simulated Datasets

We used FAVITES to simulate a sexual contact network, transmission network, viral phylogeny, and viral sequences emulating HIV transmission in San Diego from 2005 to 2014 [145].

Transmissions were modeled using a compartmental epidemiological model with 5 states: Susceptible (S), Acute HIV Untreated (AU), Acute HIV Treated (AT), Chronic HIV Untreated (CU), and Chronic HIV Treated (CT). Individuals in state S (i.e., uninfected) can only transition to state AU. Each infected state  $x \in \{AU, AT, CU, CT\}$  defines a “rate of infectiousness”  $\lambda_{S,x}$ : given an uninfected individual  $u$  in state S who has  $n_x$  sexual partners in state  $x \in \{AU, AT, CU, CT\}$ , the transition of  $u$  from S to AU is a Poisson process with rate  $\lambda_u = \sum_{x \in \{AU, AT, CU, CT\}} n_x \lambda_{S,x}$ . To mimic reality, where ART significantly reduces the risk of transmission, rates are chosen such that  $\lambda_{S,AU} > \lambda_{S,CU} > \lambda_{S,AT} > \lambda_{S,CT} \approx 0$ . At the beginning of the epidemic simulation, all initially uninfected individuals are placed in state S, and all initially infected (i.e., “seed”) individuals are distributed among the 4 infected states according to their steady-state proportions. This model is a simplified version of the model proposed by Granich *et al.* (2009) [79].

For the most part, we used the base parameters used in Moshiri *et al.* (2018) that sought to model San Diego [145], with the following modifications to better capture reality. See Table C.2 for the full set of parameters of the default condition.

## **Sexual Contact Network**

To capture the scale-free nature of the sexual contact network, Moshiri *et al.* (2018) used the BA model [42]. In addition to the scale-free property, in HIV sexual networks, we typically observe many densely-connected communities [151], a property the BA model fails to directly model. To have control over the number of communities, we simulated sexual contact networks such that networks contained 20 BA communities, each with 5,000 individuals. In the base condition, the expected degree of connection between an individual and somebody *within* their community was chosen to be 10, and the expected degree between an individual and somebody *outside* their community was chosen to be 1. Each community was simulated separately using the BA model and connections between communities were chosen uniformly at random, akin to the ER model [43]. Estimates from the literature put the number of contacts at 3–4 during a single year [88]. Because our simulated sexual contacts remain static over the 10 year simulation period, we explore mean degrees between 10 and 30.

## **Epidemic Initialization**

In Moshiri *et al.* (2018), at the start of the epidemic, all infected individuals were in state AU [145]. Here, instead, we randomly distribute initially infected individuals according to expected proportions of the states. To find these proportions, we ran simulations in which all seed individuals were in state AU, and we observed the proportion of individuals in each state over time, which reached a steady-state fairly early in the simulations (Fig. C.7).

## **Time of Sequencing**

In Moshiri *et al.* (2018), viral sequences are obtained from individuals exactly at the end time of the 10-year simulation period [145]. In reality, however, HIV patients are typically sequenced when they first visit a clinic to receive ART. Thus, it is expected that the terminal branch lengths of trees simulated in Moshiri *et al.* (2018) are artificially longer than would be



expected. Instead, we sample viral sequences from individuals the first time they begin ART (i.e., the first time they enter state AT or CT). Our current simulation better captures standards of care in advanced health care systems.

### **Simulated Data Analysis**

For each simulated sequence dataset, using FastTree 2 [102], a phylogenetic tree was inferred under the GTR+ $\Gamma$  model from the sequences obtained in the first 9 years of the simulation. These trees were then MinVar-rooted using FastRoot [134], and ProACT was run on the resulting trees.

### **3.5.2 San Diego Dataset**

To test ProACT on real data, we used a MSA of 926 HIV-1 subtype B *pol* sequences from San Diego collected by the UC San Diego PIRC. PIRC is one of the largest longitudinal cohorts of SH<sup>+</sup>Is in the United States. By design, PIRC strives to include acute infections (as much as 40% of recruited individuals are during acute or early stages of infection). Access to the data was obtained through a proposal submitted to PIRC.

A phylogenetic tree was inferred from the MSA under the GTR+ $\Gamma$  model using Fast-Tree 2 [102], and the resulting tree was MinVar-rooted using FastRoot [134]. For each decile, using TreeSwift [152], the full tree was pruned to only contain samples obtained up to the end of that decile. ProACT was run on each of the resulting trees.

### **3.5.3 Evaluation Procedure**

#### **Simulated Data**

To measure the efficacy of a given ProACT selection, because the true transmission histories are known in simulation, we simply average the number of infections caused by the

individuals in the selection in the last year of simulation (i.e, after prioritization) to obtain a raw outcome measure.

Let  $A = \{1, \dots, n\}$  denote the first,  $\dots$ ,  $n$ -th sampled individual in the current time step (years 1–9 in our simulations). For each individual  $i$ , let  $c(i)$  denote the number of individuals directly infected by  $i$  in the next time step (year 10 in our simulations). Given any set of individuals  $s \subseteq A$ , let  $C(s) = \frac{1}{|s|} \sum_{i \in s} c(i)$  denote the average  $c(i)$  for all individuals  $i \in s$ .

Let  $x = (x_1, \dots, x_n)$  denote an ordering of  $A$ . The (unadjusted) CMA of  $x$  up to  $i$  is  $C(\{x_1, \dots, x_i\})$ . Let  $o = (o_1, \dots, o_n)$  denote the ordering of  $A$  in which elements are sorted in descending order of  $c(i)$  (i.e., the optimal ordering), with ties broken arbitrarily. We defined the adjusted CMA of  $x$  up to  $i$  as

$$\frac{C(\{x_1, \dots, x_i\}) - C(A)}{C(\{o_1, \dots, o_i\}) - C(A)}. \quad (3.1)$$

We use Equation 3.1 to measure the effectiveness of a selection of the top  $i$  individuals from each ordering of all individuals. We explore  $i$  for 1 to 10% of the total number of samples (i.e.,  $\frac{|A|}{10}$ ).

## Real Data

The sequences were sorted in ascending order of sample time and, for each decile, they were split at the decile to form two sets: *pre* and *post*. A phylogenetic tree was inferred from the sequences in *pre* under the GTR+ $\Gamma$  model using FastTree 2 [102] and MinVar-rooted [134]. Using the resulting tree, ProACT ordered the samples. Then, pairwise distances were computed between each sequence in *pre* and each sequence in *post* under the TN93 model [47] using the `tn93` tool of HIV-TRACE [26].

A natural function to compute the riskiness score of a given individual  $u$  in *pre*, similar to that proposed by Wertheim *et al.* (2018) [103], is to simply count the number of individuals in *post* who are genetic links to  $u$ , i.e.,  $\sum_{v \in post} [d(u, v) \leq 1.5\%]$ . In other words, the score function is simply a step function with value 1 for all distances less than or equal to 1.5% and 0 for all other distances. However, the selection of 1.5% as the distance threshold, despite being

common practice in many HIV transmission clustering analyses, is somewhat arbitrary, and a step function exactly at this threshold may be overly strict (e.g. should a pairwise distance of 1.51% be ignored?).

To generalize this notion of scoring links, we utilized three analytical score functions. The first is the aforementioned step function  $f_1(d) = [d \leq 1.5\%]$ . The second is a sigmoid function  $f_2(d) = \frac{\lambda+1}{\lambda^{d/0.15} + \lambda}$  with the choice of  $\lambda = 100$  and  $\lambda = 5$  (Fig. C.5). The third is an empirical scoring function learnt from the data by fitting a mixture model of three Gaussian random variables onto the distribution of pairwise TN93 distances  $f_3(d) = \frac{p_1(x)}{p_1(x)+p_2(x)+p_3(x)}$ , where  $p_1(x)$  is the Probability Density Function (PDF) of the Gaussian component with smallest mean and  $p_2(x)$  and  $p_3(x)$  are the remaining Gaussian components (Fig. C.5). Specifically, the three Gaussian fits were parameterized by  $(\mu_1 = 0.0191, \sigma_1 = 0.0103)$ ,  $(\mu_2 = 0.0609, \sigma_2 = 0.0118)$ , and  $(\mu_3 = 0.118, \sigma_3 = 0.0468)$ , respectively.

For each of these function, for each decile to define *pre* and *post*, we performed a Kendall’s tau-b test to compare the prioritization approaches [146]. To generate a null distribution in Figure 3.4, we randomly shuffled the individuals in *pre* repeatedly; note however that the *p*-values reported in Table 3.2 are the theoretical *p*-values computed by the tau-b test, not empirically estimated from our repeated shuffling.

## 3.6 Acknowledgments

We thank Susan B. Little for providing the San Diego HIV sequence dataset used in this study. We also thank Joel O. Wertheim and Sanjay R. Mehta for fruitful discussions that helped motivate the development of ProACT.

This work was supported by the National Institutes of Health (5P30AI027767-28, AI100665, AI106039, and MH100974) and a developmental grant from the University of California, San Diego Center for AIDS Research (P30 AI036214), supported by the National Institutes of Health.

Chapter 3, in full, has been submitted for publication of the material as it may appear in “ProACT: Prioritization Using Ancestral Edge Lengths” (2019) Moshiri, Niema; Smith, Davey; Mirarab, Siavash, *Molecular Biology and Evolution*. The dissertation author was the primary investigator and first author of this paper.

## **Chapter 4**

### **TreeSwift: A Massively Scalable Python**

### **Tree Package**

Phylogenetic trees are essential to evolutionary biology, and numerous methods exist that attempt to extract phylogenetic information applicable to a wide range of disciplines, such as epidemiology and metagenomics. Currently, the three main Python packages for trees are Bio.Phylo, DendroPy, and the ETE Toolkit, but as dataset sizes grow, parsing and manipulating ultra-large trees becomes impractical for these tools. To address this issue, I developed TreeSwift, a user-friendly and massively scalable Python package for traversing and manipulating trees that is ideal for algorithms performed on ultra-large trees.

## 4.1 Motivation and Significance

Phylogenetic trees are essential to evolutionary biology, and phylogenetic methods are applicable to a wide range of disciplines, such as epidemiology [24, 22] and metagenomics [153, 154, 155]. However, the datasets analyzed by these methods are growing rapidly as sequencing costs continue to fall, emphasizing the need for scalable methods of tree traversal and manipulation. Beyond the analysis of real datasets, phylogenetic approaches can be utilized in the analysis of potentially massive datasets generated by simulation experiments [145].

Methods for performing phylogenetic analyses such as clustering [25] and rerooting [134] are typically presented as a series of higher-level tree traversals and manipulations. The developers of these tools do not commonly implement basic tree processing from scratch: they typically utilize existing tree packages to handle low-level tasks and instead implement their algorithms as a series of calls to functions of these packages. As a result, the performance of such a tool depends not only on the time complexity of its algorithm, but also on the performance of the underlying tree package.

Currently, the three main Python packages for trees are the Bio.Phylo module of Biopython [156], DendroPy [78], and the ETE Toolkit [157]. The three tools are simple to integrate into new methods, include a plethora of functions that cater to most phylogenetics needs, and

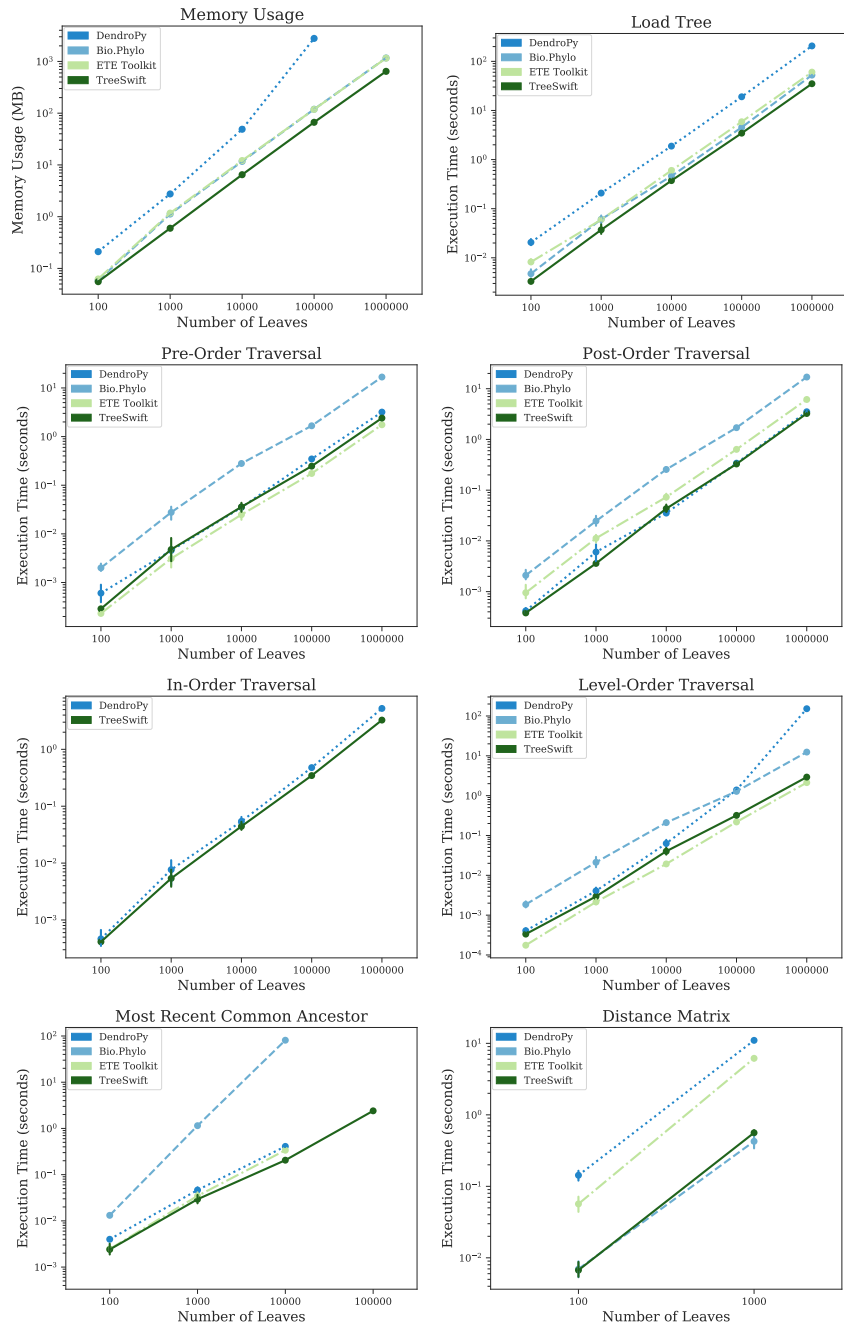
are fast for reasonably-sized trees. However, as dataset sizes grow, parsing and manipulating ultra-large trees becomes impractical. I introduce TreeSwift, a scalable cross-platform Python package for traversing and manipulating trees that does not require any external dependencies, and I compare its performance against that of Bio.Phylo, DendroPy, and the ETE Toolkit.

## 4.2 Software Description

### 4.2.1 Software Overview

TreeSwift is a pure-Python package that has no required external dependencies and which has been tested on Python versions 2.6–2.7 and 3.3–3.7. It is also compiled and hosted on PyPI, meaning it can easily be installed with a single `pip` command without any need for administrative privileges or any advanced knowledge. This is essential to contrast against the current state-of-the-art, ETE Toolkit, which requires the Six and NumPy Python libraries to install if the user has administrative privileges or Anaconda/Miniconda to install if the user doesn't, and BioPython, which requires a C compiler and the NumPy Python library as well as the computer fluency to compile tools from source using a `Makefile`.

A key feature of TreeSwift is its simplicity in class design in order to reduce time and memory overhead of loading, traversing, and manipulating trees. The entire package consists of just two classes: a `Node` class, which contains the data and local relationships, and a `Tree` class, which handles manipulation and traversal on the `Node` objects. A key distinction between TreeSwift and DendroPy is that DendroPy stores bipartition information to enable efficient comparisons between multiple trees that share the same set of taxa, but because TreeSwift is designed for the fast traversal and manipulation of individual trees (and not for the comparison of multiple trees), TreeSwift forgoes this feature to avoid the accompanied overhead, resulting in a much lower memory footprint and faster execution of equivalent functions (Fig. 4.1).



**Figure 4.1:** Runtimes of DendroPy, Bio.Phylo, the ETE Toolkit, and TreeSwift for a wide range of typical tree operations using trees of various sizes, as well as memory consumption after loading a tree. The truncation of a given tool’s plot implies lack of scalability beyond that point, and the entire lack of a given tool implies lack of implementation of the tested functionality. Timing was performed on a computer running CentOS release 6.6 (Final) with an Intel(R) Xeon(R) CPU E5-2670 0 at 2.60GHz and 32 GB of RAM.



## 4.2.2 Software Functionalities

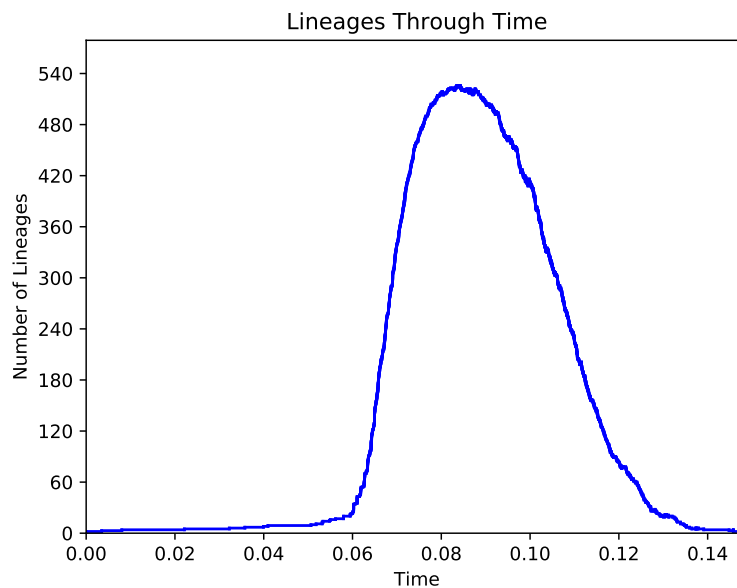
TreeSwift supports loading trees in the Newick, Nexus, and NeXML file formats via the `read_tree_newick`, `read_tree_nexus`, and `read_tree_nexml` functions, respectively. Inputs to these functions can be strings, plaintext files, or gzipped files, and TreeSwift handles the nuances of parsing them internally to maintain user-friendly operability.

TreeSwift provides generators that iterate over the nodes of a given tree in a variety of traversals, including pre-order, in-order, post-order, level-order, and root-distance-order. TreeSwift also allows for the modification of the structure of a given tree by simply modifying the `Node` objects of the tree. These built-in generators and modifiers intend to provide developers a simple yet efficient manner in which to implement their own algorithms such that they only need to consider higher-level details of the traversal process.

TreeSwift also provides the ability to compute various summarizing statistics of a given tree, such as tree height, average branch length, patristic distances between nodes in the tree, treeness [158], and the Gamma statistic [159]. Beyond numerical statistics to describe trees, TreeSwift can also generate a visual summary of a tree in the form of a Lineages Through Time (LTT) plot [160], a feature not currently implemented in any other Python tree package.

## 4.3 Illustrative Example

In the following example, I load a tree from a gzipped file, compute the minimum distance from each node in the tree to a leaf, print the minimum leaf distance of the root, and create a LTT plot (Fig. 4.2).



**Figure 4.2:** Example LTT plot generated using TreeSwift.

```
from treeswift import read_tree_newick

tree = read_tree_newick("my_huge_tree.nwk.gz")

min_leafdist = dict()

for u in tree.traverse_postorder():

    if u.is_leaf():

        min_leafdist[u] = 0

    else:

        min_leafdist[u] = min(min_leafdist[c]+c.edge_length for c in u.children)

print("Minimum leaf distance from root: %f" % min_leafdist[tree.root])

tree.lineages_through_time(color="blue")
```

---

## 4.4 Impact

The key impact of TreeSwift is its significant performance improvement over existing Python tree packages (Fig. 4.1). For almost all tested tree operations, TreeSwift performed tasks significantly faster than all existing tools (by orders of magnitude at times), and it was the only tool that not only had all tested functions implemented, but that also was able to scale to the largest of tested datasets. Further, TreeSwift's memory consumption was significantly lower than all existing tools. Thus, phylogenetic tools written in Python can utilize TreeSwift for scalability.

Further, TreeSwift was designed to be simple to use. As can be seen in the example code in Section 4.3, a user with minimal Python experience can generate a LTT plot in just 3 lines of Python code. Even complex tree algorithms can be implemented cleanly by utilizing TreeSwift's traversal generators [25].

It must be emphasized that, although TreeSwift was designed with the field of phylogenetics in mind, the package is general in that it can be utilized with any arbitrary tree structure, including those in non-phylogenetic applications [152]. Thus, its utility can extend well beyond its intended phylogenetics audience.

## 4.5 Conclusions

In this article, I presented TreeSwift, a pure-Python package for loading, traversing, and manipulating trees in a massively-scalable manner. The current version implements a wide range of typical tree operations, and due to its simple design, I hope to engage other developers to further expand TreeSwift's capabilities to target a larger suite of potential applications.

## 4.6 Acknowledgements

This work was supported by NIH subaward 5P30AI027767-28 to NM. I would like to acknowledge Siavash Mirarab for his mentorship. I would also like to acknowledge Jeet Sukumaran and Mark Holder, as DendroPy provided much motivation during TreeSwift’s development.

Chapter 4, in full, has been submitted for publication of the material as it may appear in “TreeSwift: A Massively Scalable Python Tree Package” (2019) Moshiri, Niema, *SoftwareX*. The dissertation author was the primary investigator and sole author of this paper.

# **Chapter 5**

## **Bioinformatics Education**

With rapid advances in sequencing technologies, the entire field of biology has largely shifted to depend upon the ability to analyze ultra-large datasets. As a result, the ability to perform basic computation has become a necessary prerequisite for successful biological research, yet it is only barely beginning to enter official undergraduate biology curricula as a required topic. Further, these skills are required not only by undergraduate biologists, but by graduate students, post-docs, and even faculty members and professionals, yet these individuals may not have the ability to enroll in undergraduate Computer Science courses. In an attempt to address this gap in education availability, I have dedicated significant effort to develop MAITs for use in MOOCs as well as in flipped in-person classrooms.

## **5.1 Introduction**

### **5.1.1 Bioinformatics Education: The New Frontier**

With the introduction of Next Generation Sequencing (NGS) technologies, researchers gained the ability to perform large-scale sequencing experiments at extremely high throughput with relatively low costs [161]. Due to the massive sizes of the datasets that are produced in such experiments, basic computational education has become increasingly necessary for successful biological research. While professors at top universities have started introducing bioinformatics courses into undergraduate curricula in recent years [162, 163, 164], access to such courses is typically restricted to students who have the ability to *enroll* in undergraduate courses at these top universities. However, high tuition costs disproportionately prevent low-income and minority students from entering such universities [165], leading to disparity in terms of who actually has access to such learning materials. Further, undergraduate students are not the only audience of interest for courses in such topics: graduate students, post-docs, and even faculty and professionals who received formal training in biological and biomedical sciences without any computational coursework are in need of these bioinformatics courses. In addition to difficulties

faced by students, due to the rapid growth of the popularity of computational courses [166], instructors of such courses tend to struggle to scale their courses to accommodate large class sizes.

### **5.1.2 The MOOC Revolution**

With the creation of companies like Coursera and edX, university professors started to develop MOOCs: tuition-free courses taught over the internet to a large number of students. What started as just a handful of courses, such as *Machine Learning* by Andrew Ng (2012) [167], eventually blew up, and all major universities started releasing MOOCs on a wide range of subjects [168]. Much research went into how to design these courses [169, 170, 171, 172]. Further, MOOCs seemed to attract increased participation by residents of countries in which higher education is extremely rare, far more significant representation of women than in universities, a large proportion of individuals who are either unemployed or seeking to change field of employment, and a considerable number of individuals simply taking courses for interest [173]. However, their reception was generally mixed: some enjoyed the freedom of filling their education gaps at their own pace [174], whereas others were pessimistic about their educational value [175]. Many complaints were aimed at the passive learning encompassed in traditional MOOCs, in which students simply watch a series of lecture videos and answer simple multiple choice quizzes embedded throughout.

### **5.1.3 From MOOCs to MAITs**

Phillip Compeau and Pavel Pevzner released the first ever bioinformatics MOOC, *Bioinformatics Algorithms* (2014) [176], and with it, a new technology to revolutionize online learning: the MAIT, an online text that has integrated quizzes, numerical problems, and even coding challenges to allow the learner to directly interact with the content and to allow the instructor

to enable active learning, even in a remote and automated setting [177]. The challenges are adaptive in that they provide the student uniquely-tailored feedback based on the student's specific misconception, and the text itself is adaptive in that the user can take his or her own unique "learning path." For example, a biology student would have the ability to take optional "detours" on prerequisite computer science topics such as time complexity, whereas a computer science student would have the ability to take optional "detours" on prerequisite biology topics such as the Central Dogma. These carefully-written MAITs were the foundation upon which the *Bioinformatics Algorithms* MOOCs were built, and the adaptivity and interactivity was generally well-received by the learners.

#### **5.1.4 "Bioinformatics" Means Nobody Gets Left Behind**

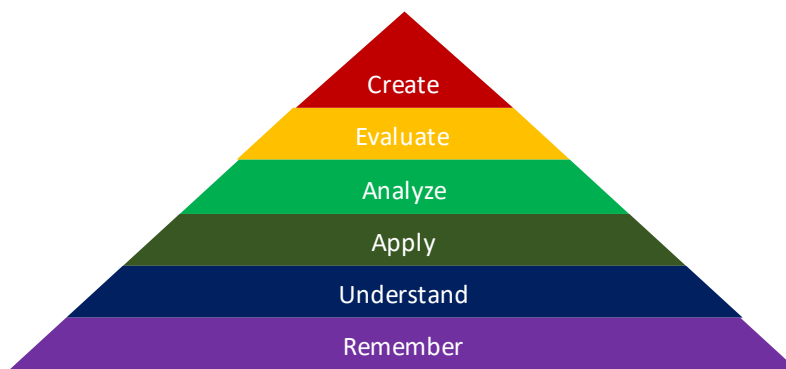
Despite the great success of the *Bioinformatics Algorithms* MOOCs, the space of online bioinformatics education was not yet filled: these courses were excellent for students with extensive backgrounds in programming, discrete mathematics, and algorithms, but for all biologists who wanted to transition into the computational aspects of the field, these courses were incomprehensible due to the students' lack of computational background. This motivated my work in bioinformatics education: the development of beginner-friendly MAITs to embed within MOOCs as well as to integrate into offline classrooms.

## **5.2 Methods**

### **5.2.1 Teaching Philosophy**

Just like running a traditional offline classroom, developing a MAIT requires the implementation of various pedagogical techniques to optimize the learning experience and to enhance student outcomes. As such, the pedagogical design of a MAIT is essential to its success. In this





**Figure 5.1:** Bloom's Taxonomy

section, I discuss the pedagogical techniques I utilize when developing MAITs.

### **Bloom's Taxonomy**

Bloom's Taxonomy is a set of three hierarchical models used to classify educational learning objectives into levels of complexity and specificity [178]. The cognitive (i.e., knowledge-based) domain of the taxonomy is a hierarchy containing the following levels: Remember, Understand, Apply, Analyze, Evaluate, and Create (Fig. 5.1) [179]. I follow the guidelines of Bloom's Taxonomy when developing my materials.

### **Active Learning**

Within my MAITs, I implement the Active Learning approach: students actively engage with the materials as opposed to simply passively reading or viewing them [180]. Specifically, I integrate numerous multiple choice, short answer, numerical, and coding challenges that can be solved directly within the text. By undergoing frequent assessment throughout the learning process, students are able to gauge their mastery of concepts *throughout* a given section, and they will be able to correct their misconceptions precisely when they occur (unlike many existing self-paced learning resources, which typically assess student mastery at the *end* of each section).

## **Adaptive Learning**

A common misconception is that online education lacks the personalized qualities of an offline course. However, on the contrary, in my MAITs, I demonstrate that my tens of thousands of students are able to receive far more personalized feedback than is possible in an offline class of even tens of students. Specifically, in my MAITs, all challenges (including coding) are automatically graded via carefully-designed Intelligent Tutor Systems (ITSs), which attempt to provide students uniquely-tailored feedback based on their specific misconceptions (Fig. 5.2).

## **Inquiry-Based Learning**

In introductory computational courses, the topics that are covered are rarely very interesting when presented out-of-context. When I present new topics in my MAITs, I first motivate them using a real-world problem in the form of a story. By employing Inquiry-Based Learning, an educational strategy in which students perform tasks in a fashion similar to those undertaken by professional scientists in order to construct knowledge [181].

## **Discovery Learning**

Research into Discovery Learning has showed that, when a student finds the solution to an open-ended problem on their own, the student benefits two-fold: the student typically has a stronger fundamental understanding of the solution, and the student has an improved perception of his or her own abilities to solve problems of this nature [182]. In my MAITs, instead of simply presenting the learning goal to the student, I try to *guide* the students and have them discover the solution on their own.

## **Making Learning Fun!**

In my own experiences as a student, I often found it difficult to complete assigned reading assignments and would quickly lose interest during classes. In computational textbooks and

**Code Challenge:** Implement **PatternCount** (reproduced below).

**Input:** Strings *Text* and *Pattern*.

**Output:** *Count(Text, Pattern)*.

```
PatternCount(Text, Pattern)
  count ← 0
  for i ← 0 to |Text| - |Pattern|
    if Text(i, |Pattern|) = Pattern
      count ← count + 1
  return count
```

(a)

**Time Limit:** 5 seconds

**Memory Limit:** 256 MB


```
1 def PatternCount(Text, Pattern):
2     count = 0
3     for i in range(0, len(Text)-len(Pattern)):
4         if Pattern == Text[i:i+len(Pattern)]:
5             count += 1
6     return count
```

Submit

Run code

(b)

Code Challenge – Write a program, test using stdin → stdout

 Didn't work.

Failed test #4. You are failing to account for a pattern occurring at the end of the text.

Get Hint!

You have 1 hint left for today.

(c)

**Figure 5.2:** Example code challenge. (a) Each problem has a clear prompt, and (b) students can solve the problems directly within the text. In this example solution, the student has an off-by-one bug (the student misses the last index), and (c) the carefully-designed ITS is able to provide the student personalized feedback.

learning resources, I often felt as though the learning materials were presented in a manner that was unneededly dry and complex. Instead, I fill my MAITs with stories, jokes, and puns, and I attempt to avoid the use of unnecessarily complex jargon when describing concepts to ensure that students of a wide range of backgrounds are able to follow successfully. I believe the success to learning is in the hands of the *learners*, and it is the responsibility of the teacher as the expert to design the educational journey to be genuinely captivating. Intuitively, it is much easier to teach when students *want* to learn.

### 5.3 Results

I developed *Analyze Your Genome!* (2017), a MOOC designed to teach biologists the best-practice workflows to analyze biological big data [183]. However, instead of discussing the specifics of the algorithms behind the analyses, I focused on how to design, execute, and interpret end-to-end bioinformatics experiments. With this approach, students are able to gain the basic proficiency required to perform relevant analyses to complement their traditional biological experiments. The course covered differential gene expression analysis using RNA-sequencing data, variant calling using Whole Genome Sequencing (WGS) vs. Whole Exome Sequencing (WES) data, rare variant calling and phasing using WGS data obtained from a trio (i.e., mother, father, and child), and bacterial genome assembly.

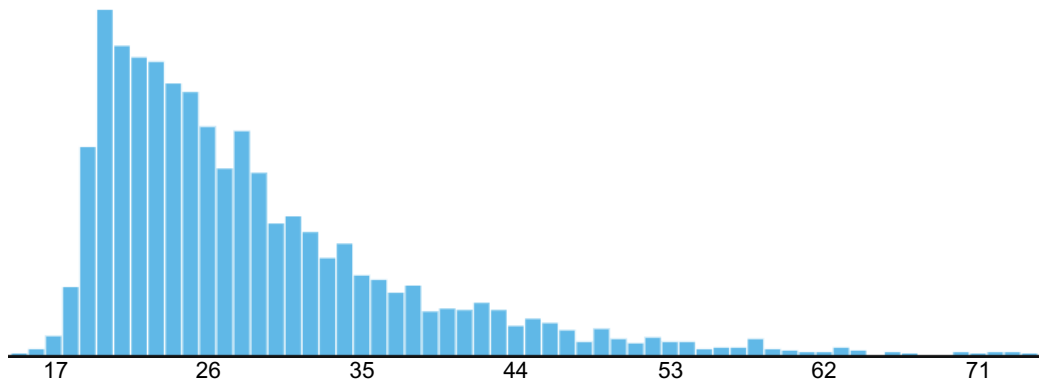
I also developed *Data Structures*, a MAIT to accompany the *Advanced Data Structures* course at the University of California, San Diego. Since its initial development, it has been integrated into data structures courses at the University of San Diego and the University of Puerto Rico. In 2017, the MAIT was integrated into a MOOC on edX: *Data Structures: An Active Learning Approach* (2017) [184]. The goal of the MOOC was to bridge the gap between introductory programming (which exists in many MOOCs) and the *Bioinformatics Algorithms* MOOCs by Compeau and Pevzner. After the large success of the MOOC, the MAIT was adapted

to a physical textbook: *Design and Analysis of Data Structures* (2018) [185]. In total, *Data Structures* has reached a total of nearly 40,000 learners in less than 3 years of existence, and the learners span a wide range of ages, education levels, and countries (Fig. 5.3).

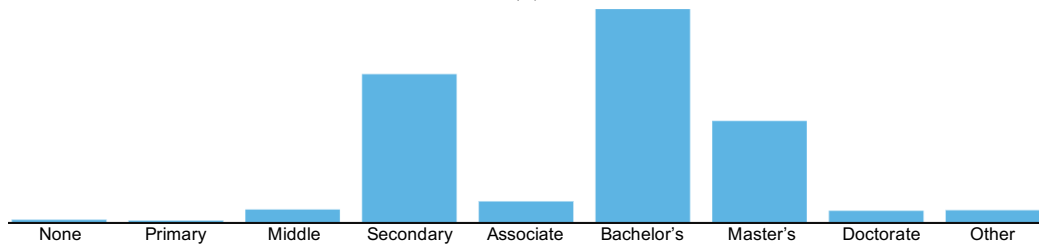
## 5.4 Discussion

Historically, the ability to learn computer science and bioinformatics has been restricted to students in higher education institutions, which can be prohibitive due to financial hardship, time constraints, or other factors. However, due to their self-paced nature, MOOCs reduce these barriers to entry, permitting entrance by previously underrepresented demographics. For example, data structures are typically only taught in undergraduate computer science courses, meaning the distribution of students is predominantly within the range of 17 through early 20s, whereas my MOOC has reached a far wider range of learners who would otherwise not take such a course (Fig. 5.3a). Further, MOOCs serve as a unique opportunity for learners who may have formal education in one field but wish to transition fields, such as biologists with Bachelors, Masters, or even Doctorate levels of education who wish to learn introductory computer science (Fig. 5.3b). Lastly, MOOCs are accessible to curious minds across the globe, thus enabling the education of individuals who physically would not be able to attend a top university (Fig. 5.3c).

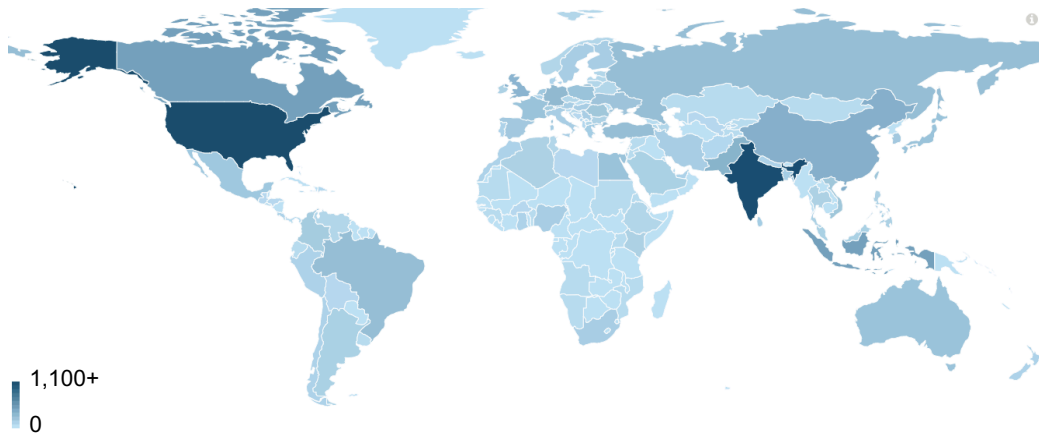
Of course, the success of my MAIT-based MOOCs is largely due to the topics I have chosen, which happen to align well with the automation capabilities of online learning platforms. Specifically, the challenges in my MOOCs are largely coding-focused, and it is typically simple to objectively determine the correctness of a student's code. However, topics in other fields (such as the social sciences) may not experience this simple objectivity in assessing correctness, which could lead to difficulties in developing ITSs to automate grading and provide personalized feedback. Even within Computer Science and Bioinformatics, coding-focused courses may be prime for this form of presentation, but more theoretical or proof-based courses will certainly not



(a)



(b)



(c)

**Figure 5.3:** (a) Age distribution, (b) education level distribution, and (c) geographical locations of learners in *Data Structures: An Active Learning Approach*.

enjoy the same ease of design.

Further, not all students in the same way, and just like any other educational technology or mode of instruction, MAITs may not be the optimal mode of instruction for all students. For example, some students strongly prefer the ability to directly interact with their instructor, and while online education does permit real-time interaction in the form of video meetings, the student may not perceive the interaction to be as meaningful if done remotely. On the other hand, other students who feel lost in large lectures may actually prefer the self-paced and adaptive nature of MAITs, which can provide them a far more personalized learning experience than can an instructor teaching 300+ other students simultaneously.

In short, I believe that, when designed carefully and executed properly, a MAIT can be a powerful tool for improving learning and for allowing instructors to reach a massive number of individuals in a highly-scalable fashion. In the future, I will continue to develop high-quality MAITs to address the learning needs of the bioinformatics community.

# **Appendix A**

## **Supplemental Material for Chapter 1**

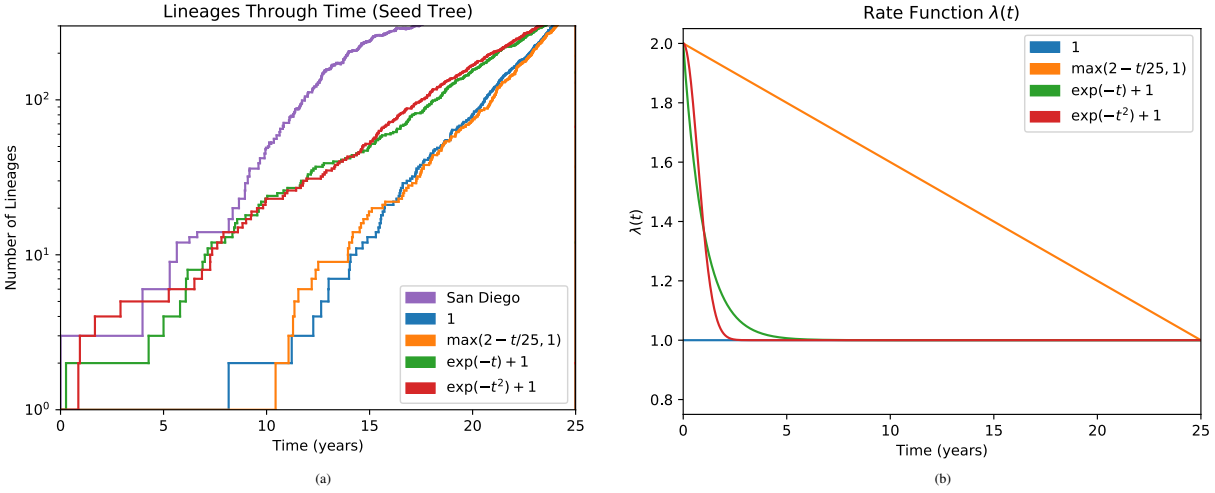


**Table A.1:** Comparison with Existing Simulation Tools

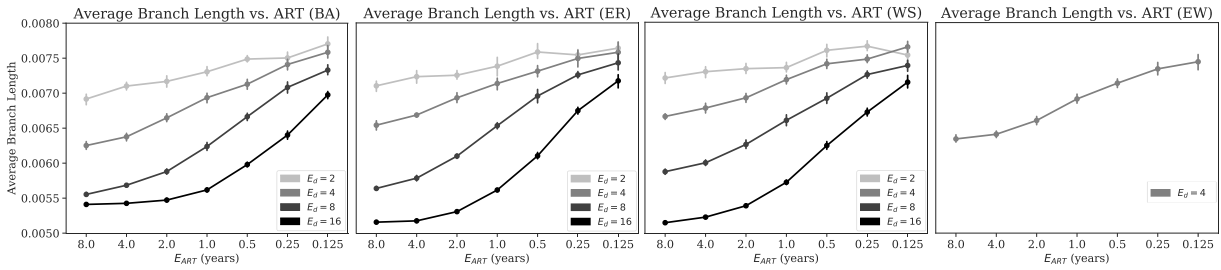
	<b>epinet</b>	<b>TreeSim</b>	<b>outbreaker</b>	<b>seedy</b>	<b>PANGEA</b>	<b>FAVITES</b>
<b>Contact Network</b>	Fixed	Complete	Complete	Complete	Fixed	Any
<b>Trans. Network</b>	Fixed	Fixed	Fixed	Fixed	Fixed	Any
<b>Sampling</b>	N/A	Fixed or Sequential	Fixed	Uniform	Fixed	Any
<b>Phylogeny</b>	None	Fixed	Fixed	Fixed	Coalescent	Any
<b>Mutation Rate</b>	N/A	N/A	Constant	Constant	Fixed	Any
<b>Sequences</b>	None	None	Fixed	Fixed	Fixed	Any
<b>Sequencing</b>	N/A	N/A	No	No	No	Any

**Table A.2:** Post-Validation Tools

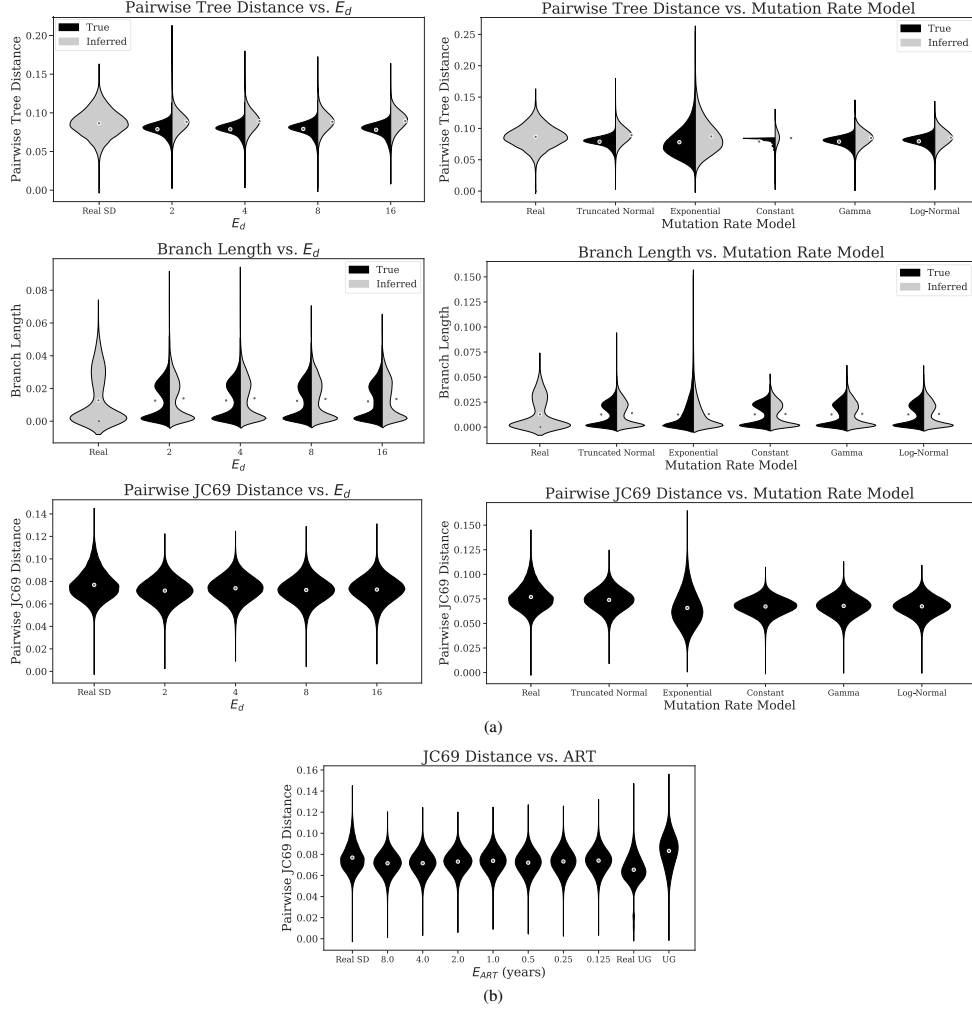
<b>Name</b>	<b>Description</b>
compare_contact_networks.py	Compare the degree distributions of a given simulated contact network against a reference contact network
compare_trees.py	Compare the distributions of all branch lengths, internal branch lengths, terminal branch lengths, and root-to-tip distances between a given simulated tree against a given reference tree
distribution_distance.py	Compute a distance between two distributions given samples from each
lft.py	Create a LTT plot from one or more Newick trees
sequence_score_profile_HMM.py	Score a given sequence dataset against a given profile HMM



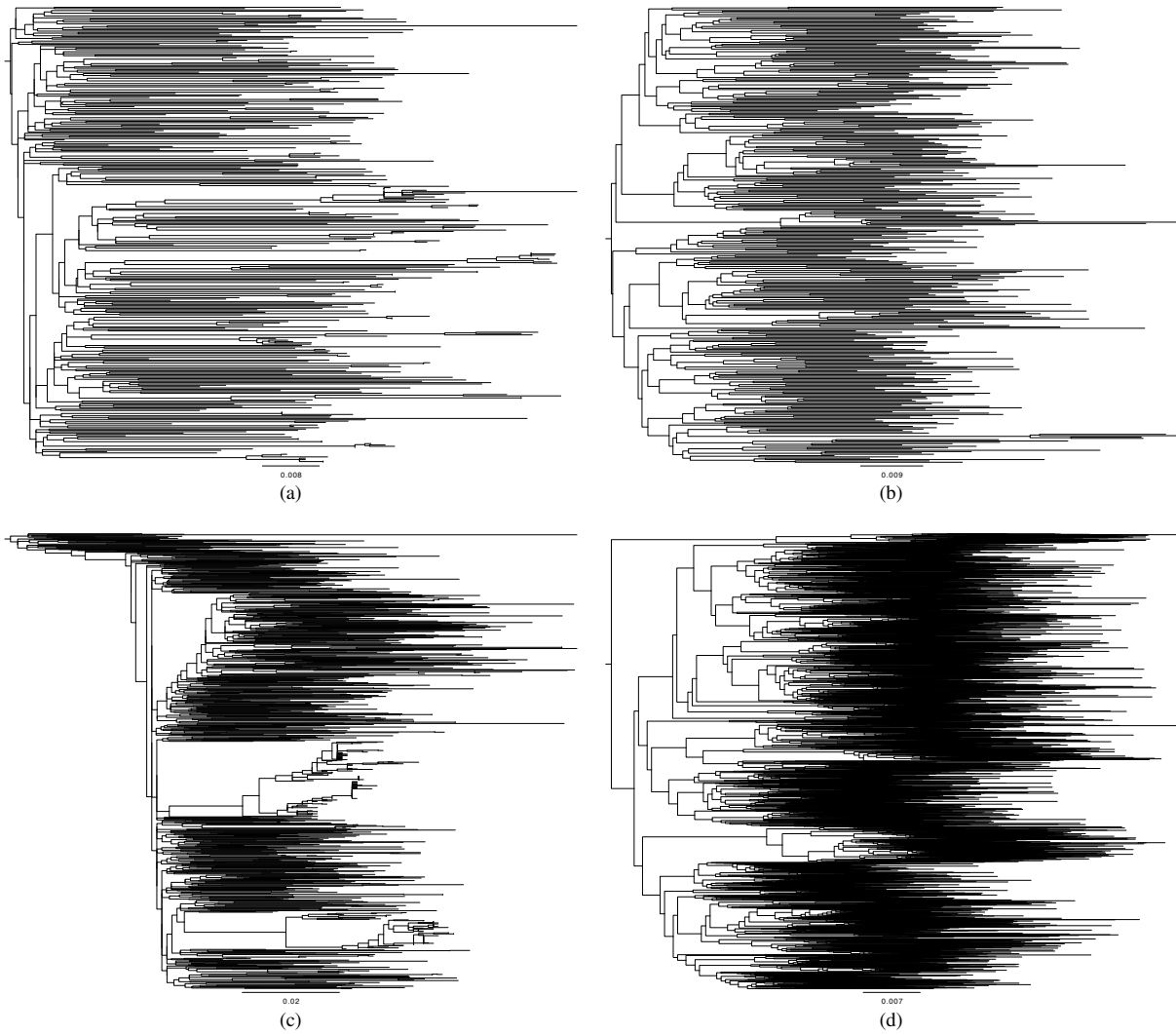
**Figure A.1:** (a) LTT plot of the first 25 years of the dated San Diego tree along with multiple potential rate functions for the non-homogeneous Yule model [99], and (b) plots of the rate functions. Because HIV trees have more short branches than normal Yule models (i.e., rate 1), we looked for functions that lead to increased numbers of lineages close to the root. This can be done by increasing the rate close to the root and then gradually decreasing the rate. As can be seen,  $\lambda(t) = 1$  and  $\lambda(t) = \max(2 - t/25, 1)$  are far lower than the real San Diego curve.  $\lambda(t) = \exp(-t) + 1$  is much closer to the real curve, and  $\lambda(t) = \exp(-t^2) + 1$  is marginally closer than it. We chose to use  $\lambda(t) = \exp(-t^2) + 1$  as a result.



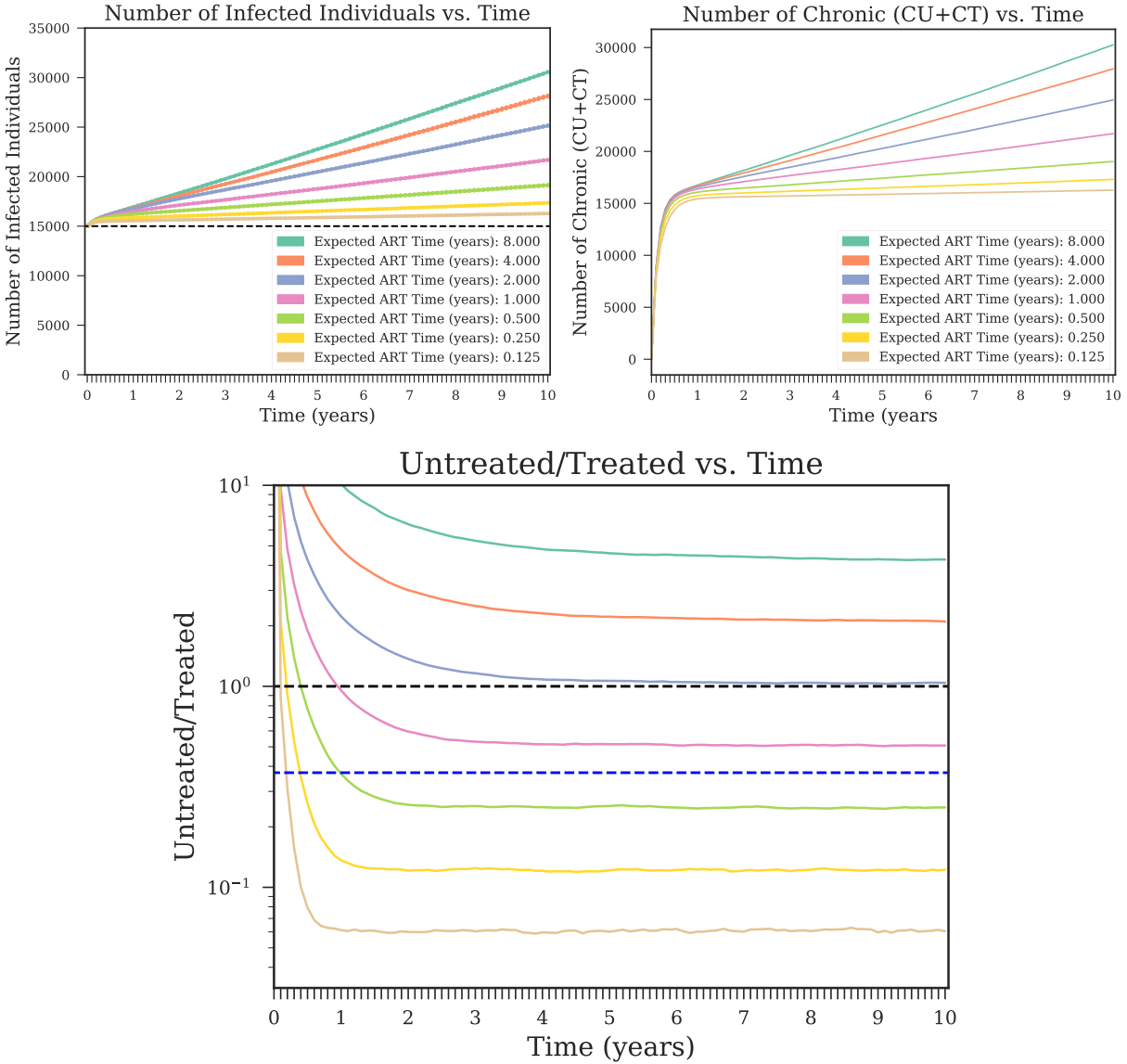
**Figure A.2:** Average true branch length vs.  $\mathbb{E}_{ART}$  for the BA, ER, and WS models with random seed selection as well as for the BA model with edge-weighted seed selection with various expected degrees. The base parameters were chosen for all other parameters.



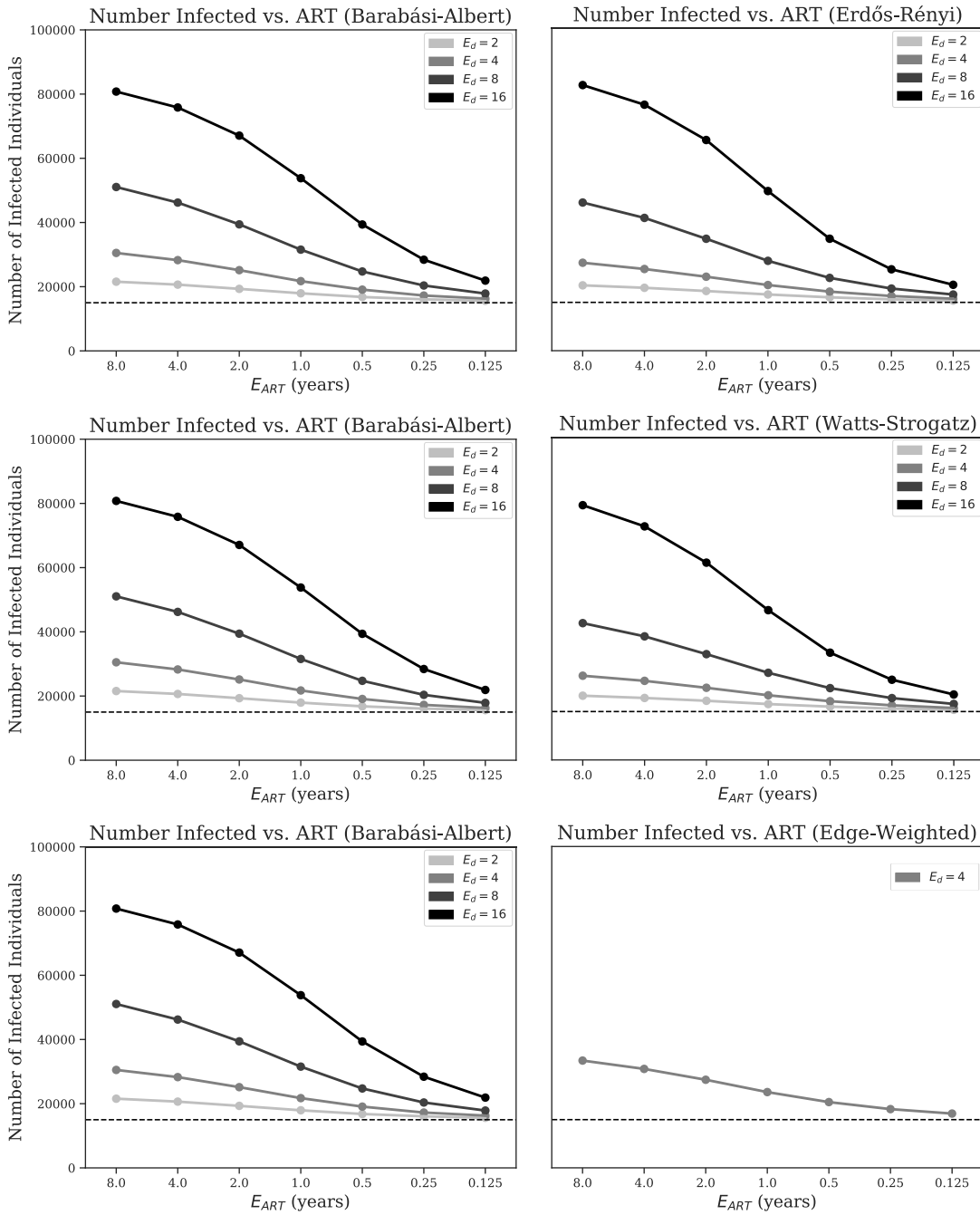
**Figure A.3:** (a) Kernel density estimates of the distributions of (Top) pairwise tree distances, (Middle) branch lengths, and (Bottom) pairwise JC69+ $\Gamma$  distances of real and simulated datasets for San Diego and Uganda using the default value of  $\mathbb{E}_{ART} = 1$  year for (Left) various values of  $\mathbb{E}_d$  and (Right) various mutation rate models. For a pair of sequences with Hamming distance  $d$ , the phylogenetic distance corrected under the JC69+ $\Gamma$  model is  $t = \frac{3\alpha}{4} \left( \left(1 - \frac{4d}{3}\right)^{-\frac{1}{\alpha}} - 1 \right)$ , where  $\alpha$  is the shape parameter of the Gamma distribution and is estimated using IQ-TREE [97] in JC69+ $\Gamma$  mode. The JSD between the distributions of each model and the real dataset distributions are as follows: for inferred pairwise distances, Truncated Normal = 0.023, Exponential = 0.055, Constant = 0.059, Gamma = 0.031, and Log-Normal = 0.024; for inferred branch length, Truncated Normal = 0.044, Exponential = 0.031, Constant = 0.072, Gamma = 0.054, and Log-Normal = 0.059. Overall, truncated normal and log-normal distributions have the best match. The JSD values for the distributions in which  $\mathbb{E}_d$  is varied can be found in Table A.6. (b) Kernel density estimates of distributions of pairwise JC69+ $\Gamma$  distances on San Diego simulations with various  $\mathbb{E}_{ART}$  values and for Uganda.



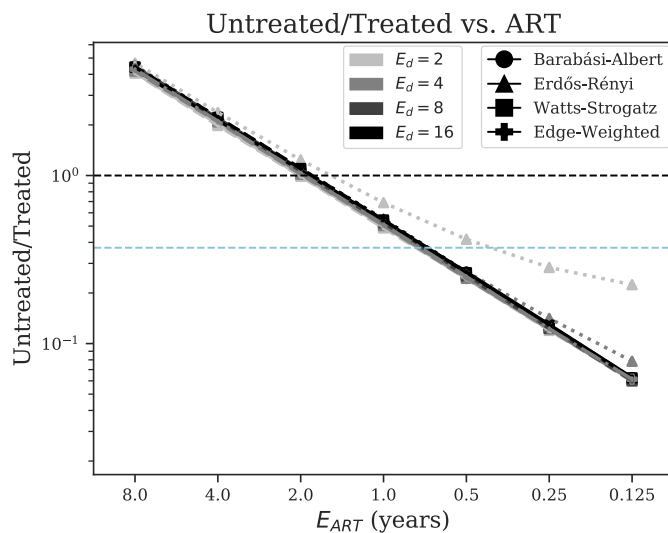
**Figure A.4:** Real versus simulated phylogenetic trees. Phylogenetic trees inferred from a real dataset of *pol* sequences (a) from San Diego [58], (b) from a simulated San Diego dataset, (c) from the set of all *pol* sequences in the LANL HIV database that were sampled in Uganda between 2005 and 2014, and (d) from a simulated Uganda dataset. Trees were inferred using the ModelFinder Plus feature [96] of IQ-TREE [97].



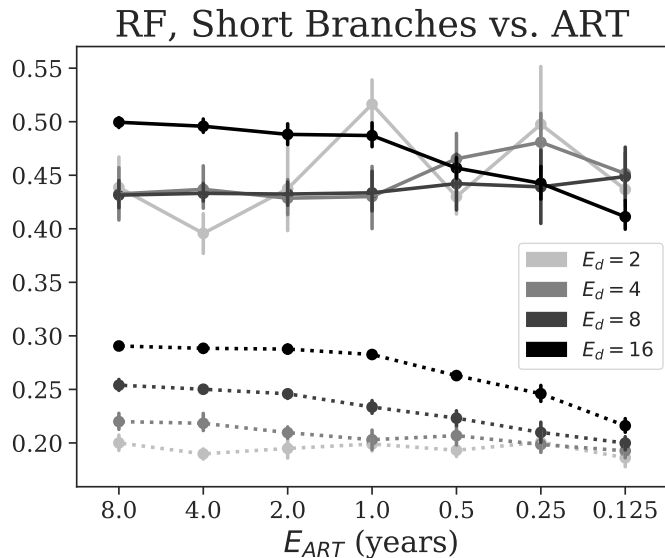
**Figure A.5:** Number of infected individuals vs. time for multiple rates of starting ART (colors). The underlying contact network was simulated using the base parameters listed in Table 1.1 with 100,000 total individuals and 15,000 seed individuals under the epidemiological model shown in Figure 1.2. We show figures for all infected people (Top Left), and those in chronic states (Top Right). We also show (Bottom) the ratio of the number of untreated individuals vs. the number of treated individuals (log-scale) vs. time where untreated/treated = 1 is shown as a dashed black line, and the value of untreated/treated corresponding to the “90-90-90” goal [105] is shown in blue.



**Figure A.6:** Total number of infected individuals vs.  $E_{ART}$  for the BA, ER, and WS models with random seed selection as well as for the BA model with edge-weighted seed selection with various expected degrees. The base parameters were chosen for all other parameters. The number of seed individuals (15,000) is shown by a black dashed line. The BA figure is repeated in each row on the left to facilitate visual comparison to other models.



**Figure A.7:** The ratio of the number of untreated vs. the number of treated individuals (log-scale) vs. expected time to begin Antiretroviral Therapy ( $\mathbb{E}_{ART}$ ) for the BA (solid circles), ER (dotted triangles), and WS models (dashed squares) with random seed selection as well as the BA with edge-weighted seed selection (dot-dashed pluses) with various  $\mathbb{E}_d$  values (colors). Untreated/treated = 1 is shown as a dashed black line, and the value of untreated/treated corresponding to the “90-90-90” goal [105] is shown as a dashed blue line ( $\frac{1-0.9^3}{0.9^3} \approx 0.37$ ).



**Figure A.8:** RF distance (solid lines) and proportion of “extremely short” branches (dotted lines) vs. expected time to begin Antiretroviral Therapy ( $\mathbb{E}_{ART}$ ) for the BA model with various  $\mathbb{E}_d$  values (colors) with all other parameters set to base values. We define branches to be “extremely short” if the expected number of mutations along the branch is less than or equal to 1 (i.e., the branch length is less than or equal to the reciprocal of the sequence length). All the trees are inferred using FastTree 2 and RF is computed with respect to the true tree.

**Table A.3:** Helper Scripts

<b>Name</b>	<b>Description</b>
clean_labels.py	For each read of the given sequence file or each leaf of a given phylogenetic tree, remove everything from the label except for the contact network individual's name
cluster_previous_time.py	Given a clustering from the simulation end time, a FAVITES-format transmission network, and a time, remove individuals who were not infected at the given time and output the resulting clusters
cn_adjacency_matrix_to_favites.py	Convert a given contact network from a binary adjacency matrix to the FAVITES format
degree_stats.py	Given a contact or transmission network, compute various statistics of the node degree distribution
FAVITES2GEXF.py	Convert a FAVITES contact network and transmission network to the GEXF format
PANGEA_transmissions_to_FAVITES.py	Convert a PANGEA transmission network into the FAVITES edge-list format
patristic_distances.py	Given a phylogenetic tree, compute the pairwise distances between leaves and output the resulting distance matrix as a CSV file
scale_tree.py	Given a phylogenetic tree (in the Newick format), scale all branches
score_clusters.py	Score a given query clustering against a given true reference clustering
tn93_to_clusters.py	Convert tn93 output to the Cluster Picker clustering format



**Table A.4:** HIV Simulation Parameters (epidemiological model)

	<b>San Diego</b>	<b>Uganda</b>
<b>CN Model</b>	BA	BA
<b>Expected Degree</b>	4	4
<b>Number of Seeds</b>	1,500	15,000
$\lambda_{AU \rightarrow CU}$ (year <sup>-1</sup> )	8.667	8.667
$\lambda_{AT \rightarrow CT}$ (year <sup>-1</sup> )	4.333	4.333
$\lambda_{U \rightarrow T}$ (year <sup>-1</sup> )	1	1
$\lambda_{T \rightarrow U}$ (year <sup>-1</sup> )	1	0.48
$\lambda_{S,AU}$ (year <sup>-1</sup> )	0.1125	0.1125
$\lambda_{S,AC}$ (year <sup>-1</sup> )	0.0225	0.0225
$\lambda_{S,AT}$ (year <sup>-1</sup> )	0.005625	0.005625
$\lambda_{S,CT}$ (year <sup>-1</sup> )	0	0

**Table A.5:** HIV Simulation Parameters (evolutionary model)

	<b>San Diego</b>	<b>Uganda</b>
<b>Seed Height</b> (years)	25	25
<b>Seed Rate</b>	$1 + e^{-t^2}$	$1 + e^{-t^2}$
<b>Population Growth</b>	2.852	2.852
<b>v.T50</b>	-2	-2
<b>Mutation Rate Location</b>	0.0008	0.001
<b>Mutation Rate Scale</b>	0.0005	0.0005
<b>GTR <math>\pi_A</math></b>	0.392	0.377
<b>GTR <math>\pi_C</math></b>	0.164	0.172
<b>GTR <math>\pi_G</math></b>	0.212	0.210
<b>GTR <math>\pi_T</math></b>	0.232	0.241
<b>GTR <math>\pi_{AC}</math></b>	1.812	1.388
<b>GTR <math>\pi_{AG}</math></b>	9.934	7.017
<b>GTR <math>\pi_{AT}</math></b>	0.718	0.736
<b>GTR <math>\pi_{CG}</math></b>	0.971	0.594
<b>GTR <math>\pi_{CT}</math></b>	9.934	8.618
<b>GTR <math>\pi_{GT}</math></b>	1	1
<b><math>\alpha</math> (<math>\Gamma</math> shape)</b>	0.405	0.449

**Table A.6:** Real vs. Simulated JSD. All columns except the last (UG; for Uganda) correspond to the San Diego simulations. JSD is computed between two distributions, one based on real data and one based on simulated data (either using true trees or trees inferred using IQ-TREE from simulated data). Distributions correspond to pairwise leaf distances on the tree (patristic distance), branch lengths (BL), and pairwise sequence distances corrected using JC69+ $\Gamma$  correction (see Fig. A.3).

	True Patristic	Inferred Patristic	True BL	Inferred BL	JC69+ $\Gamma$
$\mathbb{E}_{ART=8}$	0.195	0.027	0.050	0.100	0.049
$\mathbb{E}_{ART=4}$	0.189	0.025	0.052	0.111	0.045
$\mathbb{E}_{ART=2}$	0.193	0.033	0.045	0.097	0.035
<b>Base</b>	0.202	0.023	0.044	0.102	0.024
$\mathbb{E}_{ART=\frac{1}{2}}$	0.188	0.023	0.057	0.116	0.040
$\mathbb{E}_{ART=\frac{1}{4}}$	0.202	0.018	0.047	0.110	0.027
$\mathbb{E}_{ART=\frac{1}{8}}$	0.163	0.024	0.046	0.103	0.023
$\mathbb{E}_d=2$	0.183	0.024	0.108	0.043	0.042
$\mathbb{E}_d=8$	0.179	0.025	0.103	0.056	0.033
$\mathbb{E}_d=16$	0.196	0.035	0.108	0.053	0.034
<b>UG</b>	0.100	0.082	0.082	0.119	0.243

**Table A.7:** Simulation Result Summary. U:T denotes the ratio of untreated to treated individuals.

Condition	U:T	Prop. Inf. Increase	RF Distance	Prop. Short
Base	$0.507 \pm 0.004$	$1.449 \pm 0.005$	$0.430 \pm 0.052$	$0.203 \pm 0.015$
$\mathbb{E}_{ART=\frac{1}{8}}$	$0.061 \pm 0.001$	$1.086 \pm 0.003$	$0.452 \pm 0.032$	$0.193 \pm 0.012$
$\mathbb{E}_{ART=\frac{1}{4}}$	$0.122 \pm 0.002$	$1.150 \pm 0.004$	$0.481 \pm 0.042$	$0.199 \pm 0.012$
$\mathbb{E}_{ART=\frac{1}{2}}$	$0.248 \pm 0.004$	$1.127 \pm 0.005$	$0.465 \pm 0.040$	$0.207 \pm 0.014$
$\mathbb{E}_{ART=2}$	$1.036 \pm 0.013$	$1.677 \pm 0.019$	$0.429 \pm 0.027$	$0.210 \pm 0.009$
$\mathbb{E}_{ART=4}$	$2.122 \pm 0.019$	$1.885 \pm 0.008$	$0.437 \pm 0.034$	$0.218 \pm 0.013$
$\mathbb{E}_{ART=8}$	$4.289 \pm 0.047$	$2.034 \pm 0.012$	$0.433 \pm 0.041$	$0.220 \pm 0.013$
$\mathbb{E}_d=2$	$0.499 \pm 0.007$	$1.196 \pm 0.006$	$0.516 \pm 0.038$	$0.199 \pm 0.010$
$\mathbb{E}_d=8$	$0.531 \pm 0.009$	$2.103 \pm 0.013$	$0.434 \pm 0.031$	$0.234 \pm 0.010$
$\mathbb{E}_d=16$	$0.537 \pm 0.005$	$3.586 \pm 0.017$	$0.487 \pm 0.019$	$0.283 \pm 0.005$
ER	$0.503 \pm 0.006$	$1.359 \pm 0.007$	$0.384 \pm 0.039$	$0.186 \pm 0.015$
WS	$0.504 \pm 0.007$	$1.337 \pm 0.005$	$0.370 \pm 0.047$	$0.180 \pm 0.011$
Edge-Weighted	$0.511 \pm 0.005$	$1.571 \pm 0.007$	$0.409 \pm 0.025$	$0.209 \pm 0.009$
Uganda	$1.041 \pm 0.046$	$1.639 \pm 0.027$	$0.297 \pm 0.042$	$0.185 \pm 0.021$

# **Appendix B**

## **Supplemental Material for Chapter 2**

## B.1 Theoretical Results

### B.1.1 Proofs

**Theorem 4.** *Let  $X$  be a random variable (r.v.) over ordered ranked tree shapes and distributed according to the dual-birth model with parameter  $r = \lambda_a/\lambda_b$ . Then,*

$$\Pr(X = T_\omega^\Psi; n) = \frac{r^{n_r-1}}{\prod_{i=1}^{n-2} ((r-1)l_i + i + 1)} \quad (\text{B.1})$$

where  $n_r$  is the number of right leaves in  $T_\omega^\Psi$  and  $l_i$  is the number of its left branches before node  $i$ .

*Proof.* Proof (sketch) Consider the intervals between consecutive birth events in  $T_\omega^\Psi$ , and denote each interval by the rank of its end node (e.g. Figure 2.1a). Because  $T_\omega^\Psi$  is ordered, branches in the interval  $i$  can be ordered from left to right (including the order of parents) and assigned an index between 1 and  $i + 1$ . Two ordered ranked tree shapes are equal iff the index of the branch where node  $i$  is born is identical in the two trees for all  $i \in N$ . Seeing that two identical ordered ranked shapes have this property is trivial. The opposite direction becomes clear if the nodes that give birth at point  $i$  in the two trees are mapped together; the shapes become obviously equivalent (edges are the same), but also the ranking becomes the same. Finally, the ordering is the same because of identical left to right ordering. Let  $\xi_i$  denote the event that the index of the branch on which node  $i$  is born in  $X$  is equal to the index of  $i$  in  $T_\omega^\Psi$ . Then,  $\Pr(X = T_\omega^\Psi) = \Pr(\cap_1^{n-2} \xi_i)$ .

Birth on each branch is governed by a Poisson process with rate  $\lambda_a$  and  $\lambda_b$  for left and right branches, respectively. Due to the memoryless property of the exponential distribution, the length of each branch before node  $i - 1$  has no bearing on subsequent birth events.

Thus, given  $l_i$  (the number of left branches in the interval  $i$ ), the probability of  $\xi_i$  does not depend on  $\xi_1 \dots \xi_{i-1}$ . Therefore,  $\Pr(\cap_1^{n-2} \xi_i) = \prod_1^{n-2} \Pr(\xi_i; l_i)$ . Also, the probability that any one of the  $i + 1$  competing independent Poisson processes (present on different branches of the

interval  $i$ ) results in the first event is simply the ratio of its rate to the sum of all rates. Thus,

$$\Pr(\xi_i; l_i) = \frac{\lambda_a(1 - \omega_i) + \lambda_b \omega_i}{l_i \lambda_a + (i + 1 - l_i) \lambda_b} = \frac{r(1 - \omega_i) + \omega_i}{(r - 1)l_i + (i + 1)}.$$

Multiplying  $\Pr(\xi_i; l_i)$ s and manipulations gives results. □

**Theorem 5.** *For a tree shape  $Z$  generated by the dual-birth model with  $r = \lambda_a/\lambda_b$ , let  $C = c(Z)/n$  be an r.v.; then,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(C) = \frac{\sqrt{r}}{1 + r + \sqrt{r}} \tag{B.2}$$

**Corollary 4.** *For an r.v.  $N_r$  capturing the number of right (i.e., active) leaves in tree shape  $T$ ,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(N_r) = \frac{\sqrt{r}}{1 + \sqrt{r}} \tag{B.3}$$

*Proof.* Proof (sketch) Our proof follows the approach of McKenzie and Steel [117]. We categorize the terminal branches (i.e., those incident on leaves) of an ordered tree shape into four types: right branch in a cherry (Right Cherry, or RC), right branch not in a cherry (Right Non-cherry, or RN), left branch in a cherry (Left Cherry, or LC), and left branch not in a cherry (Left Non-cherry, or LN). Note that the number of RC and LC branches must be equal (they could be potentially combined, but the discussions are more clear if we keep them separate). Suppose an urn includes four types of balls RC, RN, LC, and LN, respectively corresponding to these four types of branches. As the tree is evolving, each birth event adds a child to one of existing terminal branches. Moreover, the terminal branch to be used is chosen at random (but not uniformly) from the terminal branches available at that time point. After the birth, two new terminal branches are added and the original branch turns into an internal branch. Because of the memoryless property of our process, each birth event is equivalent to removing a ball from the urn and adding two balls back to the urn. To make matters slightly more complicated, a birth can also change the

type of sibling branches that have not been removed (e.g. a non-cherry can be turned to a cherry). This can be modeled by removing a ball of one type and adding another ball of another type. In total, after each round, the number of added balls of each type can be potentially negative but the total number of new balls is a positive constant of 1; this kind of urn models are referred to as an Extended Polya Urn (EPU).

For EPUs, asymptotic central limit theorems exist for the distribution of balls [186]. Specifically, an EPU with  $k$  types can be described by a matrix  $A_{k \times k}$ , in which  $A_{ij}$  gives the number of balls of type  $j$  added when a ball of type  $i$  was drawn. Under certain conditions [186], the number of balls of type  $i$  out of  $n$  total balls is asymptotically normally distributed with a mean of  $n\lambda_1 v_i$ , where  $\lambda_1$  is the principal eigenvalue of  $A$  and  $v$  is its left eigenvector normalized to add up to one; more precisely, the number of all ball types asymptotically has a joint normal distribution. A birth in the dual-birth model can be described by

$$A' = \begin{matrix} RC : \\ RN : \\ LC : \\ LN : \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix}$$

Let  $p$  be the branch where the birth happens and let  $s$  be the sister to  $p$ . Each birth always adds a new RC and a new LC branch, but depending on the type  $p$  other changes will occur too. If  $p$  is an RC branch, an RC branch ( $p$ ) is removed, an RC and an LC are added, and  $s$  changes from LC to LN. Thus, in total, we gain one LN; hence, the first row of  $A'$ . A similar logic gives the third row. For the second row, note that when  $p$  is an RN type, we simply remove  $p$ , reducing the count of RN by one, and add an RC and an LC. A similar logic gives the last row.

A further complexity is that not all branches will have an equal chance of splitting in the dual-birth model. Each left (or right) branch is selected with a probability proportional to  $\lambda_a$  (or  $\lambda_b$ ). We account for this by replacing each left ball with  $\lambda_a/\lambda = r/(r+1)$  balls and each right

ball with  $\lambda_b/\lambda = 1/(r+1)$  balls. Thus, we get

$$A = \frac{1}{r+1} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & r & 0 & 0 \\ r & 0 & r & -r \end{bmatrix}$$

It can be checked that our EPU satisfies the conditions of the EPU central list theorem.

The principle eigenvalue of  $A$  is  $\lambda_1 = \frac{\sqrt{r}}{r+1}$ , and a left eigenvector is

$$v' = \left[ 1 + \sqrt{r} \quad r \quad 1 + \sqrt{r} \quad \frac{1}{1 + \sqrt{r}} \right]$$

The results immediately follow by computing  $\lambda_1 v'_1 / \sum_1^4 v'_i$  for  $\mathbb{E}(C)$  and  $\lambda_1 (v'_1 + v'_2) / \sum_1^4 v'_i$  for  $\mathbb{E}(N_r)$ . □

**Theorem 6.** *For a weighted tree shape  $t$  generated by the dual-birth model with parameters  $r$  and  $\lambda$  conditioned on having  $n$  leaves, let  $D$  be an r.v. giving the length of a random branch in  $t$ ; i.e.,  $D = \delta_I$  for  $I \sim \mathcal{U}(1, n-2)$ . Then,*

$$\lim_{n \rightarrow \infty} \mathbb{E}(D) \rightarrow \frac{1}{2\lambda} \left( \frac{r+1}{\sqrt{r}} \right) \tag{B.4}$$

*Proof.* Proof (sketch) It is constructive to think about the sampling strategy. First, an *uncensored* tree is created with  $n$  terminal branches but with varying depth for leaves. Half of the branches in this tree are drawn from the exponential distribution with rate  $\lambda_a$  and the other half with rate  $\lambda_b$ . Thus, the expected sum of branch lengths in the uncensored tree is  $\frac{1}{2}(\frac{1}{\lambda_a} + \frac{1}{\lambda_b})n$ . We then cut  $n-1$  branches. Because of the memoryless property of the exponential distribution, the expected length of the branches we cut from a tip is  $1/\lambda_b$  for right branches and  $1/\lambda_a$  for left branches. By the proof of Corollary 3, the number of left and right branches are normally distributed with

known expectation (Eq. 2.6); thus,

$$\mathbb{E}(D) = \frac{1}{n} \left( \frac{n}{2\lambda_a} + \frac{n}{2\lambda_b} - \left( \frac{\sqrt{r}}{1+\sqrt{r}} \frac{1}{\lambda_b} + \frac{1}{1+\sqrt{r}} \frac{1}{\lambda_a} \right) (n-1) \right)$$

which in limit gives the results. □

**Lemma 1.** *For a tree shape  $t$ , the most parsimonious number of activation events is the minimum number of activation events possible, which is equal to the number of cherries in  $t$ .*

*Proof.* Proof First, the most parsimonious number of activation events is the minimum number of activation events possible. As mentioned in the main paper, an activation event is a biological change, so a most parsimonious number of activation events would be one that minimizes the number of activation events. Note that, under the dual-birth model, the root node is considered an activation event.

Next, we prove that the minimum number of activation events possible is equal to the number of cherries in  $t$ . Let  $N_t$  represent the minimum number of activation events possible for tree  $t$ , let  $V_t$  denote the set of nodes in  $t$ , and let  $a_v = 1$  if an activation event occurs on node  $v$ , otherwise  $a_v = 0$ . Under the dual-birth model, activation events only occur on nodes (i.e., not on edges), so  $N_t = \sum_{v \in V} a_v$ .

A node  $v$  can have either 0 or 2 children. Under the dual-birth model, activation events occur when an inactive node gives birth to a new node, meaning activation events can only occur on internal nodes of  $t$ . Leaves that are active cannot undergo an activation event (by definition), and leaves that are inactive have not yet given birth, meaning they have not yet been activated. Therefore, for all leaves  $f$ ,  $a_f = 0$ .

For an internal node  $v$ , there are three possible cases: both children of  $v$  are leaves (i.e.,  $v$  is a cherry), one child is a leaf and the other is an internal node, or both children of  $v$  are internal nodes.

If both children of  $v$  are leaves (i.e.,  $v$  is a cherry), by definition, one child is active (and is



therefore the propagation of  $v$ ), and the other child is inactive. Therefore, either  $v$  or an ancestor of  $v$  must have undergone an activation event. Let  $A_v$  denote the set of ancestors of node  $v$ . Because the dual-birth model does not allow deactivation, there can only be a single activation event in the lineage of an active node, meaning  $a_v + \sum_{u \in A_v} a_u = 1$ .

If one child of  $v$  is an internal node and the other is a leaf, let  $c$  denote the child that is an internal node. One of the descendants of  $c$  must be a cherry. Therefore, there must be a cherry  $u$  such that  $v \in A_u$ .

If both children of  $v$  are internal nodes, by the same logic of the previous paragraph,  $v$  must be the ancestor of at least two cherries: one for each of its internal node children.

Therefore, because each lineage of a cherry must have exactly 1 activation event, and because every internal node that is not a cherry must be in the lineage of a cherry, the minimum number of activation events possible would be a single activation event for each cherry's lineage. Therefore, the minimum number of activation events is equal to the number of cherries in  $t$ .  $\square$

## B.1.2 Set of All Possible Orderings

For  $T^\Psi = (T, \psi)$ , the set  $\Omega(T^\Psi)$  of all possible orderings for  $T^\Psi$  (as shown in Figure 2.1c of the main paper) can be constructed recursively. For an internal node  $u$  and its two children  $u_1$  and  $u_2$ , let

$$\Omega'(\{u\}) = \begin{cases} \{\{(u_1, 0), (u_2, 1)\}, \{(u_2, 0), (u_1, 1)\}\} & \text{if } u_1, u_2 \neq \otimes \\ \{\{(u_2, 0)\}, \{(u_2, 1)\}\} & \text{if } u_1 = \otimes \\ \{\{(u_1, 0)\}, \{(u_1, 1)\}\} & \text{if } u_2 = \otimes \\ \{\{\}\} & \text{if } u_1 = u_2 = \otimes \end{cases}$$

and for two disjoint node sets  $X$  and  $Y$ :

$$\Omega'(X \cup Y) = \{\omega_X \cup \omega_Y \mid (\omega_X, \omega_Y) \in \Omega'(X) \times \Omega'(Y)\}. \quad (\text{B.5})$$

Then,  $\Omega(T^\Psi) = \{\omega_X \cup \{(r, 1)\} \mid \omega_X \in \Omega'(V)\}$  where  $r$  is the root.

## B.2 Supplementary Methods

### B.2.1 Simulation Setup

#### Motivation of Default Parameters

The default value of  $n = 1,000$  (which is used for all trees in all experiments) is chosen because it is large enough to observe changes in tree shape resulting from tweaking the other parameters, yet it is small enough that simulations and subsequent tree inferences remain computationally tractable. The default values for the alignment simulation parameters, including GTR parameters, are chosen based on ML estimates from of the *Alu* tree as computed by FastTree-2 (see Section 2.3.3).

The default value of  $r = 10^{-2}$  is chosen because  $r = 1$  is equivalent to the Yule model and  $r = 10^{-4}$  results in an almost fully ladder-like tree, so  $r = 10^{-2}$  serves as an intermediate. The default value of  $\lambda = 169.328$  is chosen because the best estimate of the average branch length of the *Alu* tree is 0.029824, which can be used with the default value of  $r = 10^{-2}$  to find  $\lambda$  (Eq. 2.6).  $k = 300$  is chosen to match the length of *Alu*.

The default value of ultrametricity deviation gamma distribution rate  $\alpha = 29.518$  is chosen by first rooting the best estimate of the *Alu* tree on the MRCA of 7SLRNA sequences, which we assume is the outgroup of the *Alu* elements [56]. Then, root-to-tip distances are computed and are normalized by the distribution average. A gamma distribution is then fit on the resulting distribution with the constraint that the distribution's rate and shape must be equal.

## Data Generation

To generate “true trees,” we use our implementation of the generative process of the dual-birth model, which takes three parameters:  $\lambda_a$ ,  $\lambda_b$ , and  $n$ . We then deviate each tree from ultrametricity by multiplying each branch of the tree by a multiplier sampled from a gamma distribution with shape and rate both set to some parameter  $\alpha$  (so as to keep the expected value of the distribution equal to 1, and as a result, keep the average branch lengths of the trees constant). We then simulate a multiple sequence alignment with no indels according to the GTR+ $\Gamma$  model using INDELible.

We have a series of “experiments,” where we start with a default set of parameters and then deviate one parameter at a time.

- **INDELible Parameters (Global)**

- GTR Frequencies: 0.2922 0.2319 0.2401 0.2358
- GTR Rates (ac ag at cg ct gt): 0.8896 2.9860 0.8858 1.0657 3.8775 1.0000
- $\alpha = 5.256$

- **Default Parameters (param-00)**

- $n = 1000$  (Global)
- $r = 10^{-2}$
- $\lambda_a = 1.6765100539857060$
- $\lambda_b = 167.65100539857060$
- Ultrametricity Gamma Distribution Parameter  $\alpha = 29.518173529892621$
- Sequence Length = 300

- **Experiment 1 (Changing  $r$ ) (Constant Average Branch Length)**

- param-04:  $r = 10^{-4}$ ,  $\lambda_a = 0.16765100539857060$ ,  $\lambda_b = 1676.5100539857060$
- param-03:  $r = 10^{-3}$ ,  $\lambda_a = 0.53015902907666816$ ,  $\lambda_b = 530.15902907666816$
- param-00:  $r = 10^{-2}$ ,  $\lambda_a = 1.6765100539857060$ ,  $\lambda_b = 167.65100539857060$
- param-02:  $r = 10^{-1}$ ,  $\lambda_a = 5.3015902907666816$ ,  $\lambda_b = 53.015902907666816$
- param-01:  $r = 10^0$ ,  $\lambda_a = 16.765100539857060$ ,  $\lambda_b = 16.765100539857060$

• **Experiment 2 (Changing Model of DNA Evolution)**

- param-00: JC69
- param-00: K80
- param-00: HKY85
- param-00: GTRCAT
- param-00: GTR+ $\Gamma$

• **Experiment 3 (Changing  $\lambda$ )**

- param-05:  $\lambda_a = 0.33530201079714$ ,  $\lambda_b = 33.53020107971412$
- param-06:  $\lambda_a = 0.83825502699285$ ,  $\lambda_b = 83.8255026992853$
- param-00:  $\lambda_a = 1.6765100539857060$ ,  $\lambda_b = 167.65100539857060$
- param-07:  $\lambda_a = 3.35302010797141$ ,  $\lambda_b = 335.3020107971412$
- param-08:  $\lambda_a = 8.38255026992853$ ,  $\lambda_b = 838.255026992853$

• **Experiment 4 (Changing Sequence Length)**

- param-09: Sequence Length = 50
- param-10: Sequence Length = 100
- param-11: Sequence Length = 200

- param-00: Sequence Length = 300
- param-12: Sequence Length = 600
- param-13: Sequence Length = 1200
- param-14: Sequence Length = 2400
- param-15: Sequence Length = 4,800

• **Experiment 5 (Changing Number of Leaves  $n$ )**

- param-25:  $n = 25$
- param-26:  $n = 50$
- param-27:  $n = 250$
- param-28:  $n = 500$
- param-00:  $n = 1000$
- param-29:  $n = 2000$
- param-30:  $n = 4000$

• **Experiment 6 (Changing Ultrametricity Gamma Distribution Parameter  $\alpha$ )**

- param-16:  $\alpha = 2.95181735298926$
- param-17:  $\alpha = 5.90363470597852$
- param-00:  $\alpha = 29.518173529892621$
- param-18:  $\alpha = 147.590867649463$
- param-19:  $\alpha = 295.181735298926$
- param-20:  $\alpha = 9999999999999999$  (i.e.,  $\infty$ )

## Methods

For each alignment created by INDELible (one alignment per “true tree”), we use FastTree-2 and RAxML to infer a tree using the GTR+ $\Gamma$  model. We run RAxML a second time on the trees outputted by RAxML in order to compute branch support values. We then estimate cherries using the method described in Section 2.2.2.

- **FastTree 2:** `fasttree -nt -gtr -gamma < SEQS`
  - `-nt`: Alignment contains nucleotide sequences
  - `-gtr`: Use GTR model
  - `-gamma`: Rescale tree’s branch lengths to optimize Gamma20 likelihood
  - `SEQS`: INDELible multiple sequence alignment (FASTA)
- **Initial RAxML (Tree Inference):** `raxmlHPC -s SEQS -m GTRGAMMA`  
`-n OUT -p $RANDOM`
  - `-s SEQS`: Specify the sequence alignment file to be SEQS (FASTA)
  - `-m GTRGAMMA`: Use the GTR+ $\Gamma$  model
  - `-n OUT`: Specify output project name to be OUT
  - `-p $RANDOM`: Use a random number as the seed
- **Final RAxML (Branch Support):** `raxmlHPC -f J -p $RANDOM`  
`-m GTRGAMMA -s SEQS -t TREE -n OUT`
  - `-f J`: Compute SH-like support values on the given tree
  - `-p $RANDOM`: Use a random number as the seed
  - `-m GTRGAMMA`: Use the GTR+ $\Gamma$  model
  - `-s SEQS`: Specify the sequence alignment file to be SEQS (FASTA)

- -t TREE: Specify the input tree (from “Initial RAxML” step)
- -n OUT: Specify output project name to be OUT
- **Cherry Estimation:** `estimate-cherries.sh TREE THRESHOLD`
  - TREE: Tree for which to estimate cherries
  - THRESHOLD: Branch support threshold to use

## Error Measurement

We measure the accuracy of inferred tree topology using the RF distance as well as the MS metric.

- **RF Computation:** `echo $(echo -n '(' &&  
echo -n `compareTrees.missingBranch TRUE INFERRED | cut -d' ' -f3` &&  
echo -n ' + ' &&  
echo -n `compareTrees.missingBranch INFERRED TRUE | cut -d' ' -f3` &&  
echo -n ') / 2') | bc -l`
  - TRUE: “True tree” simulated by our dual-birth simulation tool
  - INFERRED: Inferred tree (from either FastTree 2 or RAxML)
  - `compareTrees.missingBranch`: Tool to compute missing branch rate (FN) between two trees
  - First compute FN of INFERRED with respect to TRUE
  - Then compute FN of TRUE with respect to INFERRED
  - Average these two FN values to compute the RF metric
- **MS Computation:** `TreeCmp.jar -r TRUE -d ms -i INFERRED`

- TRUE: “True tree” simulated by our dual-birth simulation tool
- INFERRED: Inferred tree (from either FastTree 2 or RAxML)
- TreeCmp.jar: Tool used to compute MS metric [129]
- -r TRUE: Specify TRUE to be the reference tree
- -d ms: Compute the MS distance metric
- -i INFERRED: Specify INFERRED to be the inferred tree

## B.2.2 Human *Alu* Analyses

### Data Acquisition

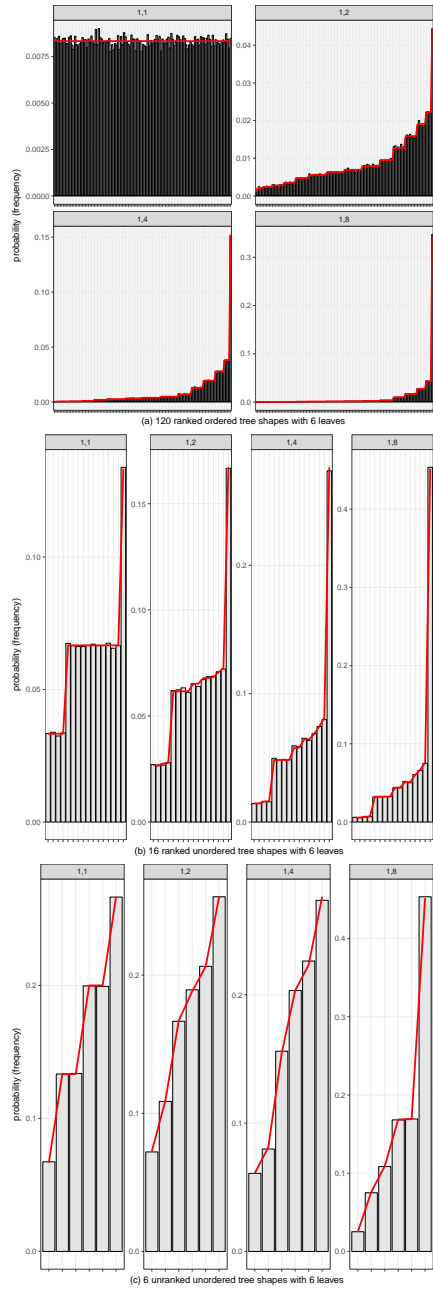
- **DfamScan:** `dfamscan.pl -fastafile hg19.fa -hmmfile Dfam-Alu.hmm -dfam_outfile hg19.out`
  - `-fastafile hg19.fa`: Specify the hg19 reference genome as the input
  - `-hmmfile Dfam-Alu.hmm`: Use the *Alu* Dfam HMM database
  - `-dfam_outfile hg19.out`: Output results to hg19.out

### Alignment and Tree Inference

Our first dataset of the “*Alu*” family based on the Dfam database included non-*Alu* items (e.g. 7SLRNA, which is thought to be a predecessor of *Alu* elements). Our initial analyses included these non-*Alu* members of the Dfam *Alu* family, which resulted in poor placement of these more ancestral repeats on the resulting tree. We filtered out these non-*Alu* elements and recomputed both alignments and trees. Our online data include both filtered and unfiltered datasets.

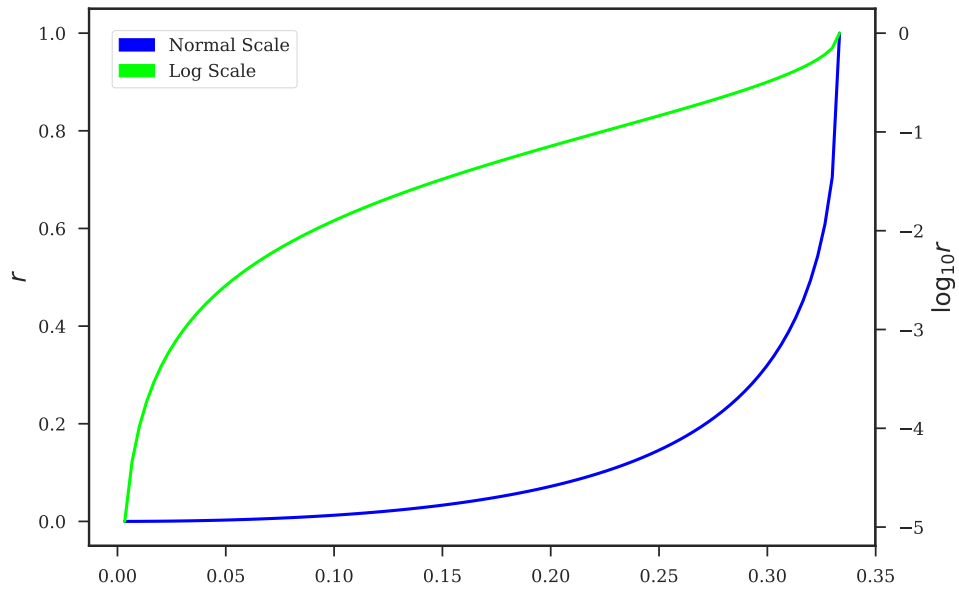


## **B.3 Supplementary Figures**

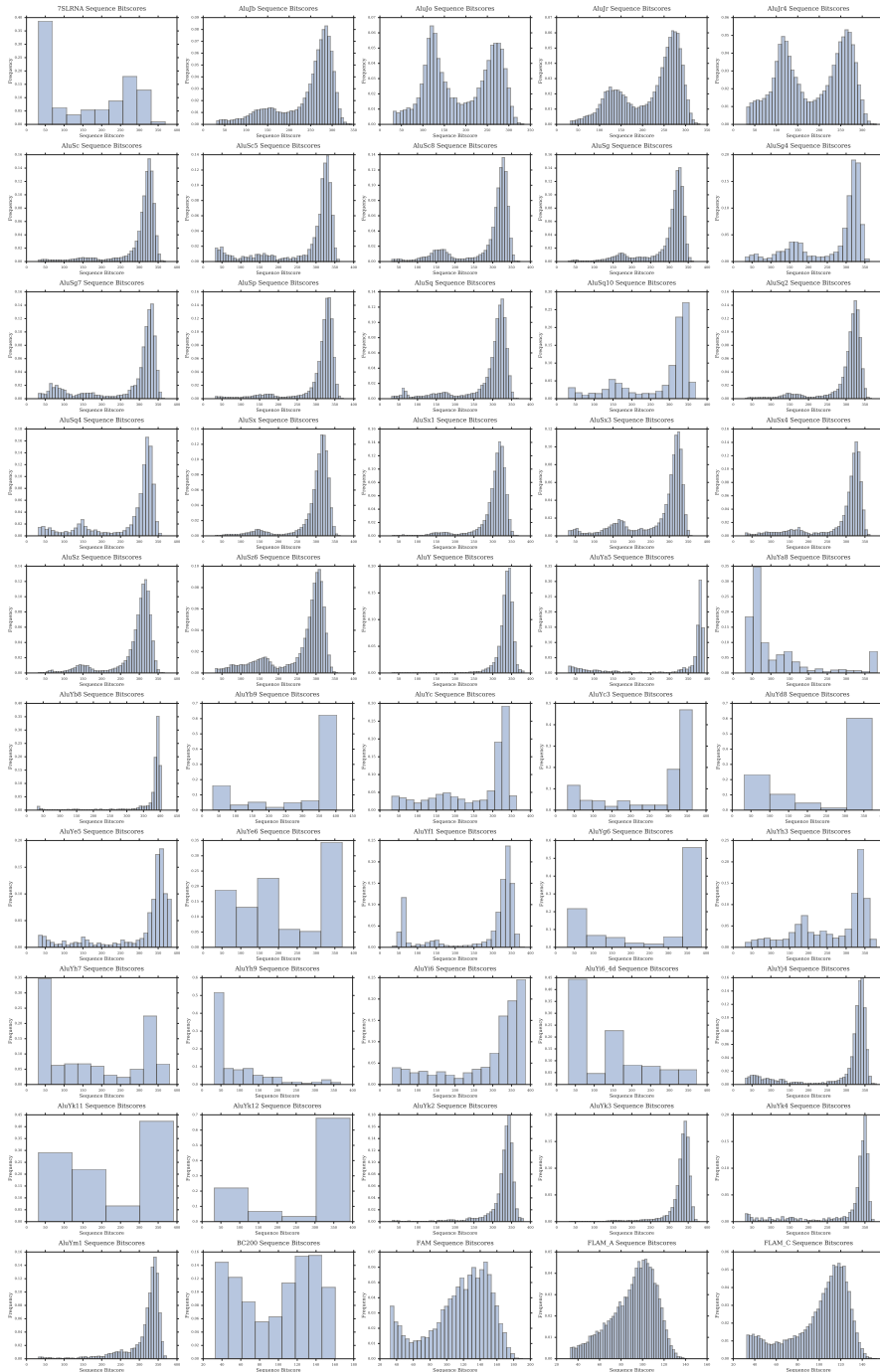


**Figure B.1:** Probability distributions on ranked tree shapes. There are (a) 120 ranked ordered tree shapes, (b) 16 ranked unordered tree shapes, and (c) 6 unranked unordered tree shapes with  $n = 6$ . The distribution according to the dual-birth model is given over these trees for four choices of  $\lambda_a$  and  $\lambda_b$  (box header) corresponding to  $r = 1$  (i.e., Yule),  $r = 1/2$ ,  $r = 1/4$ , and  $r = 1/8$ . Red line gives the theoretical distribution and the grey bars give the frequencies in 100,000 simulations.

*r* vs. Cherry Fraction



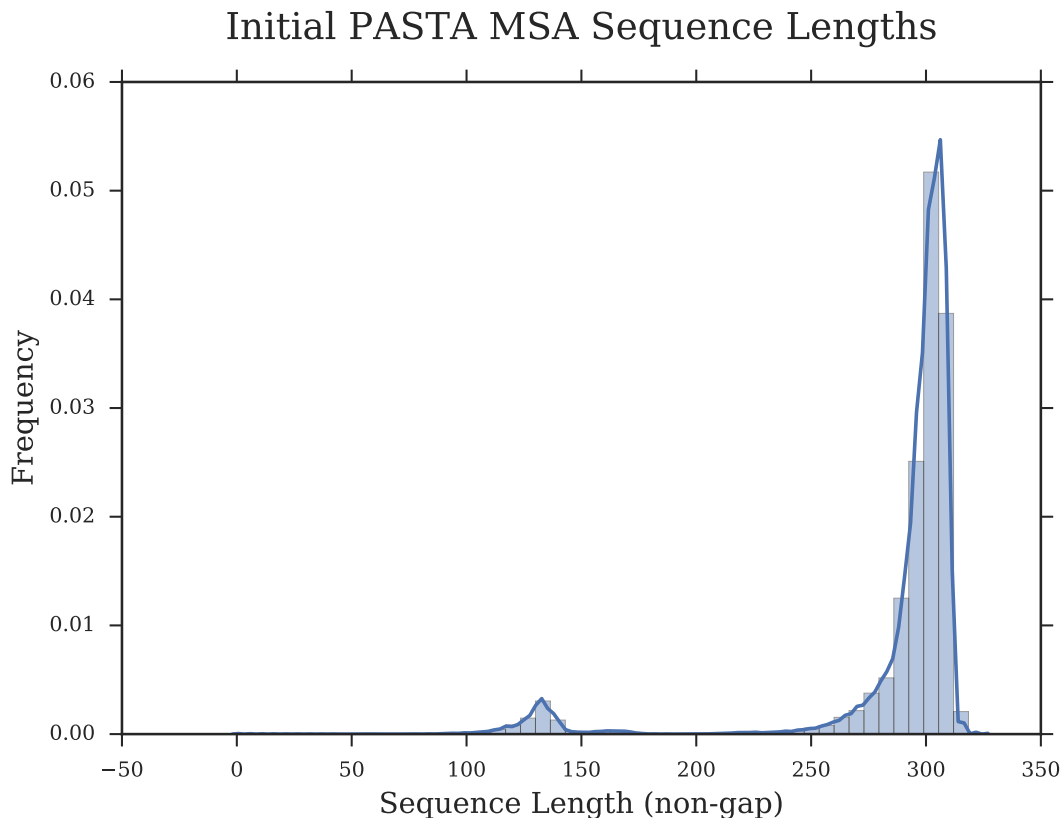
**Figure B.2:** Estimated *r* vs. Cherry Fraction. Estimated *r* (*y*-axis) as a function of the fraction of cherries (*x*-axis). Blue (left axis) shows normal scale and green (right) shows the logarithmic scale.



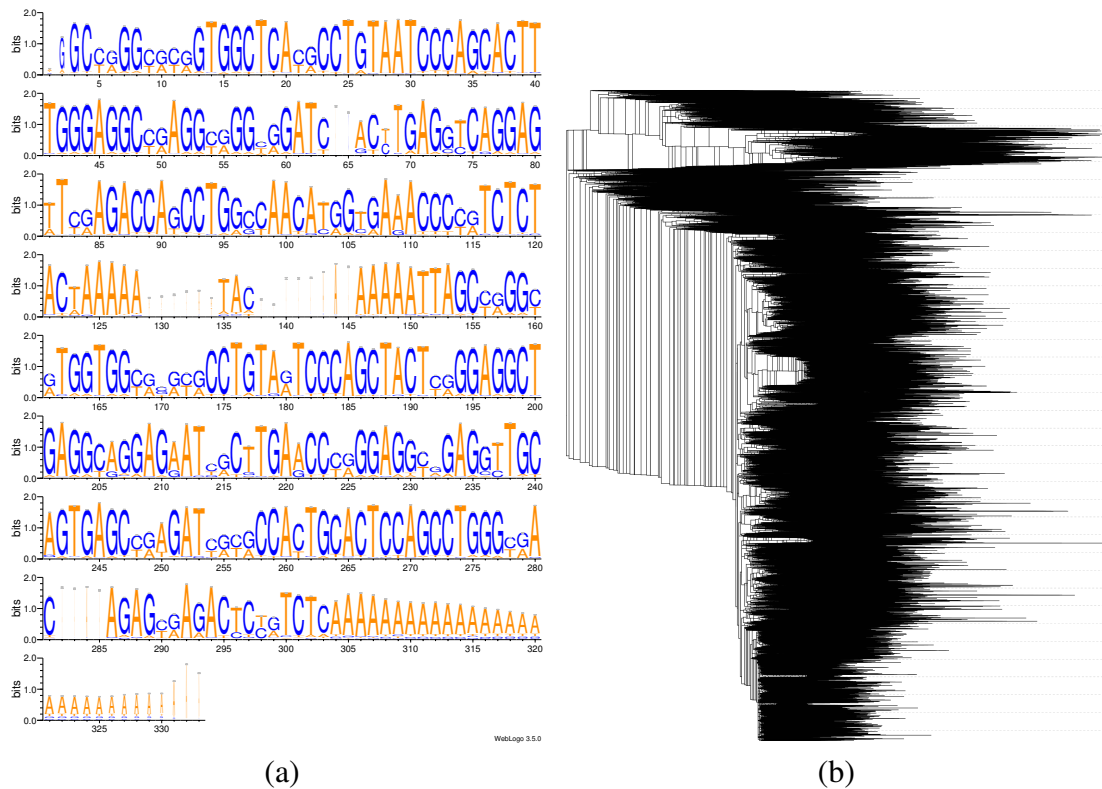
**Figure B.3:** Histograms of bitcores.

7SLRNA	AluJb	AluJo	AluJr	AluJr4	AluSc	AluSc5	AluSc8
200	200	200	225	200	275	275	275
AluSg	AluSg4	AluSg7	AluSp	AluSq	AluSq10	AluSq2	AluSq4
275	275	275	275	275	275	275	250
AluSx	AluSx1	AluSx3	AluSx4	AluSz	AluSz6	AluY	AluYa5
250	250	250	250	250	250	300	325
AluYa8	AluYb8	AluYb9	AluYc	AluYc3	AluYd8	AluYe5	AluYe6
100	250	250	275	300	300	300	300
AluYf1	AluYg6	AluYh3	AluYh7	AluYh9	AluYi6	AluYi6_4d	AluYj4
300	300	300	275	200	225	225	300
AluYk11	AluYk12	AluYk2	AluYk3	AluYk4	AluYm1	BC200	FAM
325	325	250	250	375	300	100	65
FLAM_A	FLAM_C						
80	80						

**Figure B.4:** Bitscore thresholds.

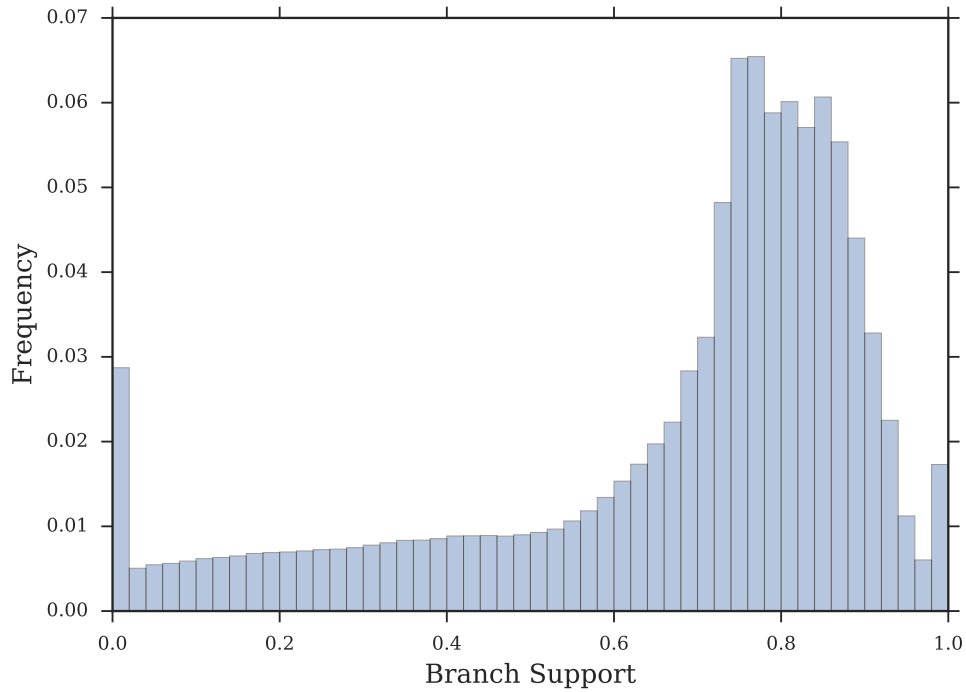


**Figure B.5:** PASTA alignment sequence lengths. Histogram of sequences based on non-gap sequence length. As can be seen, a nontrivial number of sequences in the alignment have non-gap lengths well below 300, which we know *a priori* to be the typical length of *Alu* sequences.

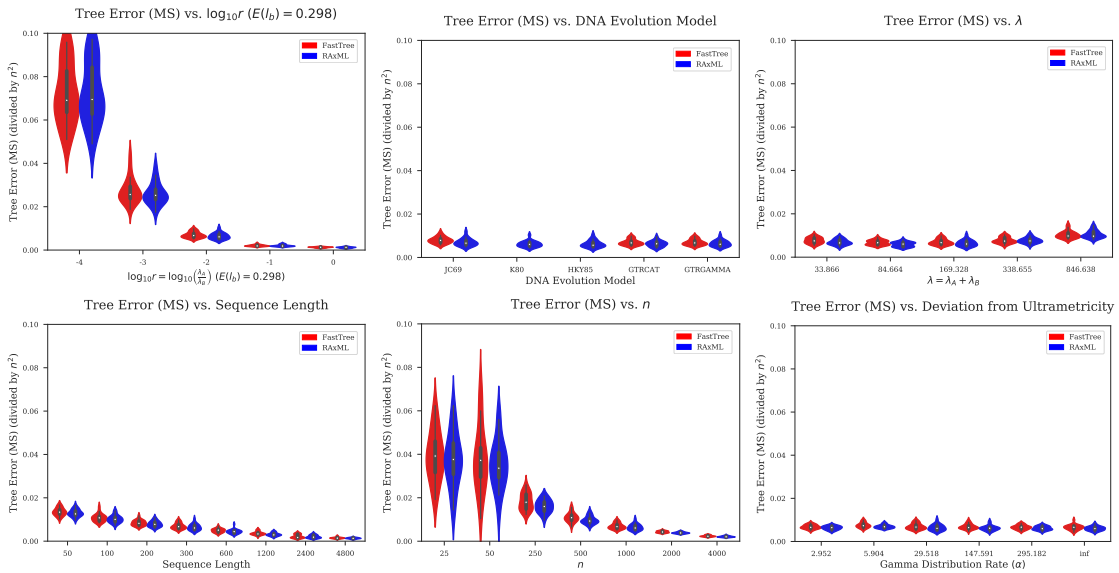


**Figure B.6:** Human *Alu* alignment and tree. (a) Sequence logo constructed from the *Alu* multiple sequence alignment in which sequences with less than 200 non-gap characters were removed and sites with less than 1% non-gap characters were masked, using WebLogo [187]. The logo indicates conserved sequences and a good quality alignment: most sites have a clear high-frequency consensus nucleotide. (b) Midpoint-rooted *Alu* phylogenetic tree inferred from the aforementioned sequence alignment by RAxML under the GTR+CAT model. As expected, portions of the tree are very ladder-like.

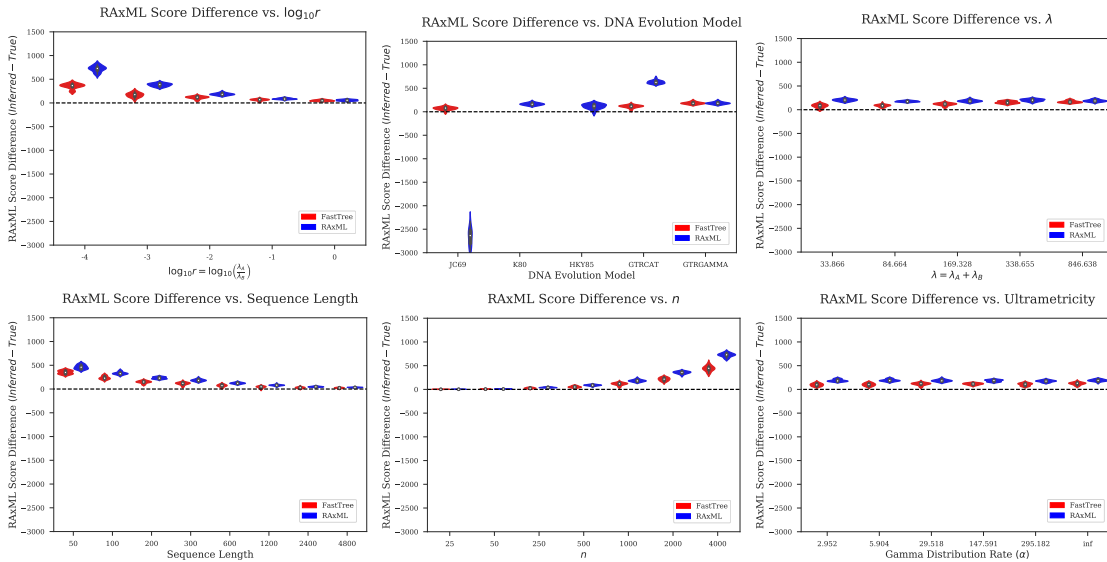
## Branch Support



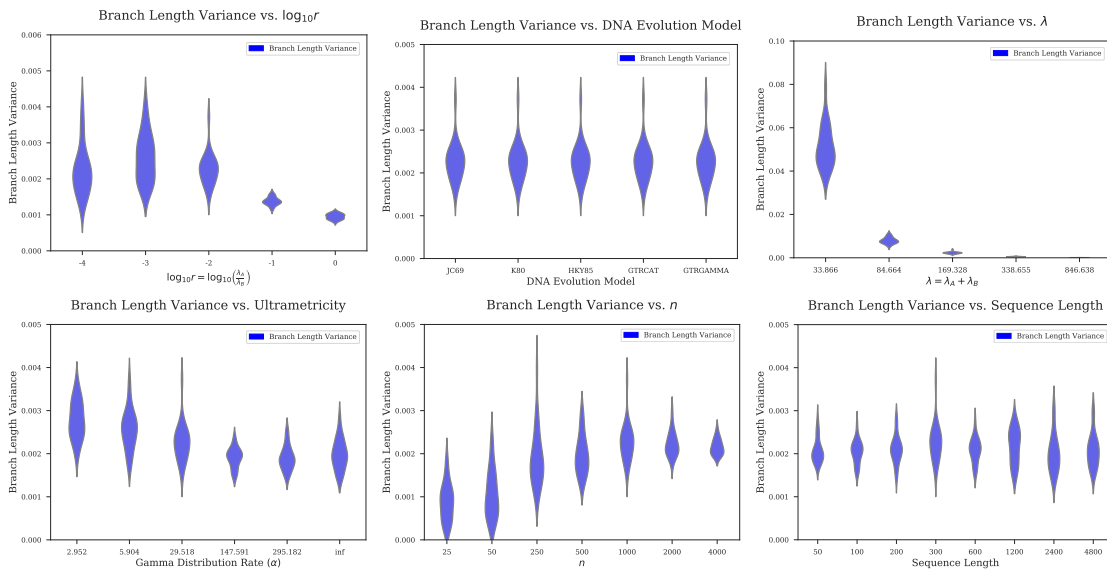
**Figure B.7:** Human *Alu* tree branch support. Histogram of SH-like branch support values in the tree constructed from the masked alignment using FastTree 2. As can be seen, there are many low-support branches. Values below 0.9 are typically considered low SH-like support.



**Figure B.8:** Tree inference error (MS). Violin plots are shown for the MS distance between true and estimated trees.

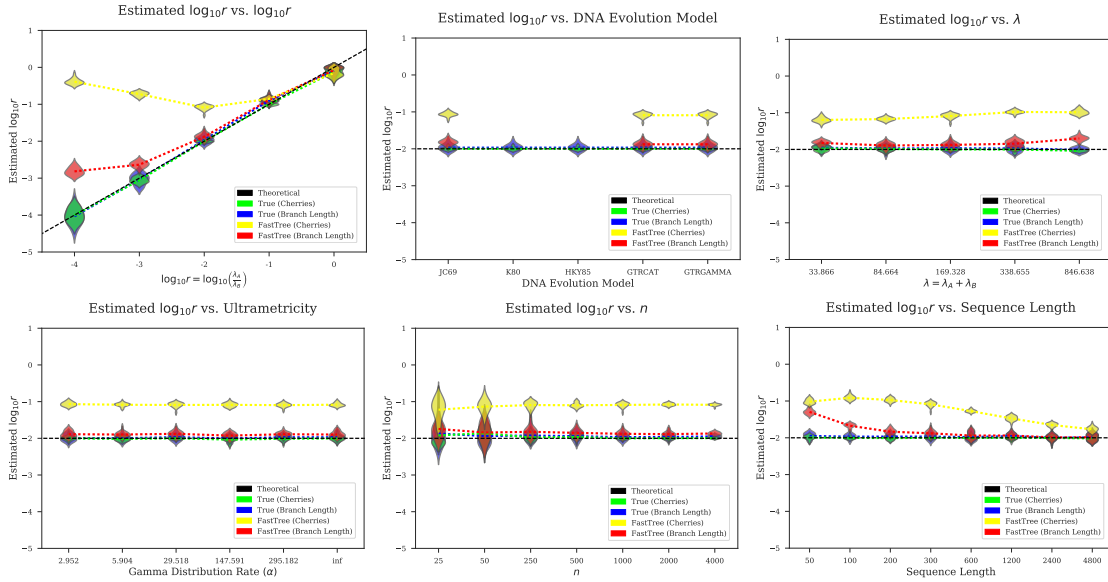


**Figure B.9:** Tree inference error. Violin plots are shown for the log-likelihood score, as computed by RAxML, of the inferred tree minus the true tree; values away from zero indicate that the true tree has low log-likelihood scores.

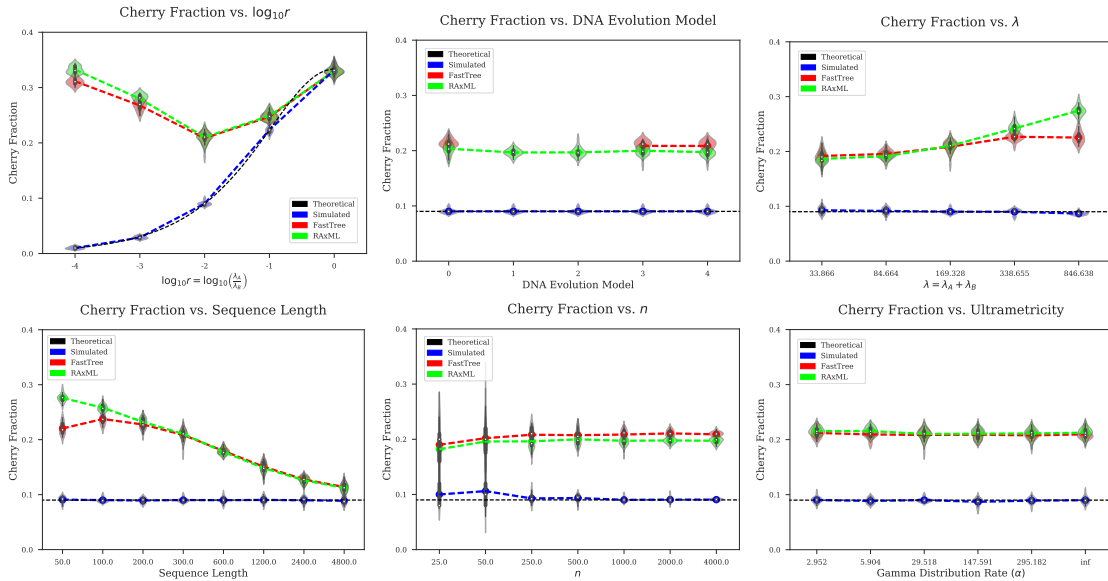


**Figure B.10:** Branch length summary statistics. Violin plots are shown for the branch length variance computed for true trees.

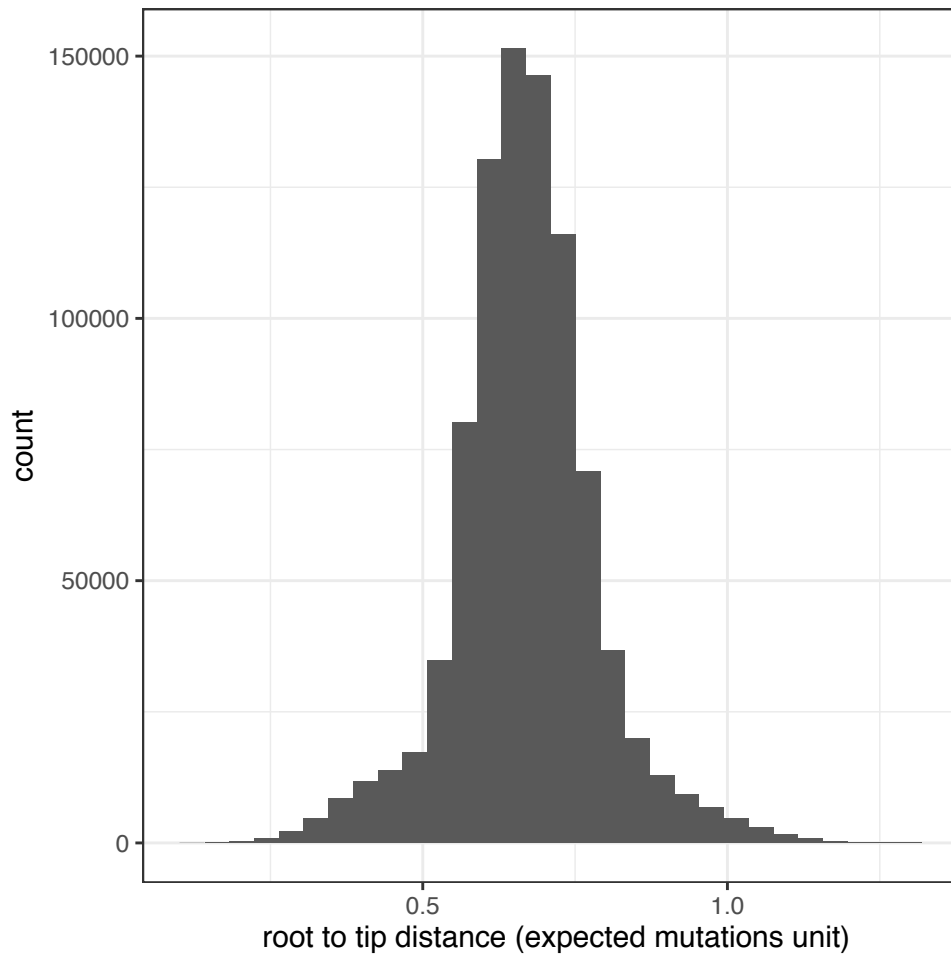




**Figure B.11:** Parameter estimation accuracy. Violin plots are shown for the estimated  $r$ , using the cherry-based estimator and the branch-length-based estimator, for true trees and for inferred FastTree 2 trees for each of the experiments. Note that FastTree 2 does not have K80 or HKY85 models implemented.



**Figure B.12:** Cherry fraction. Violin plots are shown for the cherry fractions of the true trees and inferred RAxML and FastTree 2 trees.



**Figure B.13:** Molecular clock on the *Alu* tree. The distribution of the root-to-tip distances after midpoint rooting are shown for the *Alu* tree with 1% masking. Under the molecular clock, root to tip distances for all leaves are expected to be identical.

# **Appendix C**

## **Supplemental Material for Chapter 3**

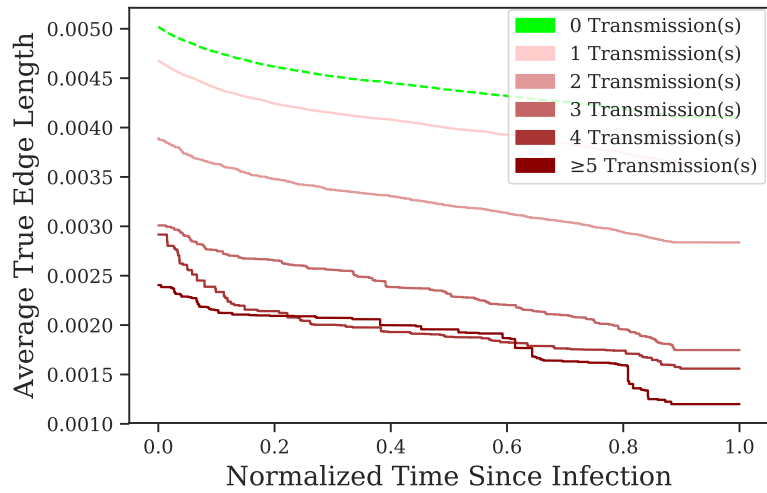
**Table C.1:** Kendall’s tau-b test for a null hypothesis that a given prioritization yields a total outcome measure no better than random. We show  $p$ -values for a real San Diego dataset for the first through ninth deciles. These  $p$ -values do not correct for multiple hypothesis testing. Tests that failed to reject the null hypothesis with (uncorrected)  $\alpha = 0.001$  are marked with †.

Sigmoid Function ( $\lambda = 5$ )

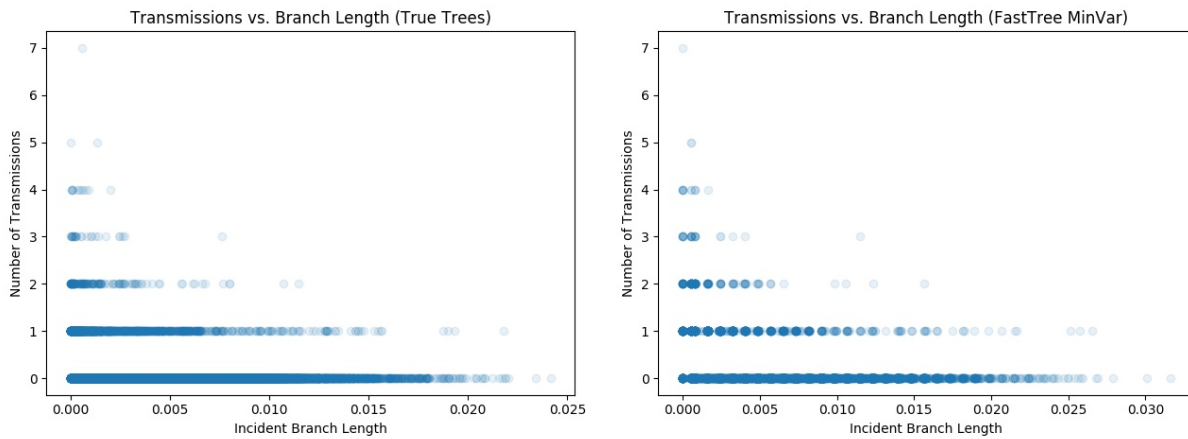
Percentile	GD + Cluster Growth	ProACT (FastTree)
10%	$6 \times 10^{-4}$	$1 \times 10^{-8}$
20%	† $6 \times 10^{-3}$	$8 \times 10^{-5}$
30%	$3 \times 10^{-7}$	$2 \times 10^{-6}$
40%	$5 \times 10^{-5}$	$5 \times 10^{-8}$
50%	$8 \times 10^{-6}$	$1 \times 10^{-8}$
60%	$2 \times 10^{-7}$	$1 \times 10^{-11}$
70%	$8 \times 10^{-8}$	$1 \times 10^{-10}$
80%	$1 \times 10^{-6}$	$3 \times 10^{-11}$
90%	$1 \times 10^{-10}$	$1 \times 10^{-17}$

Sigmoid Function ( $\lambda = 100$ )

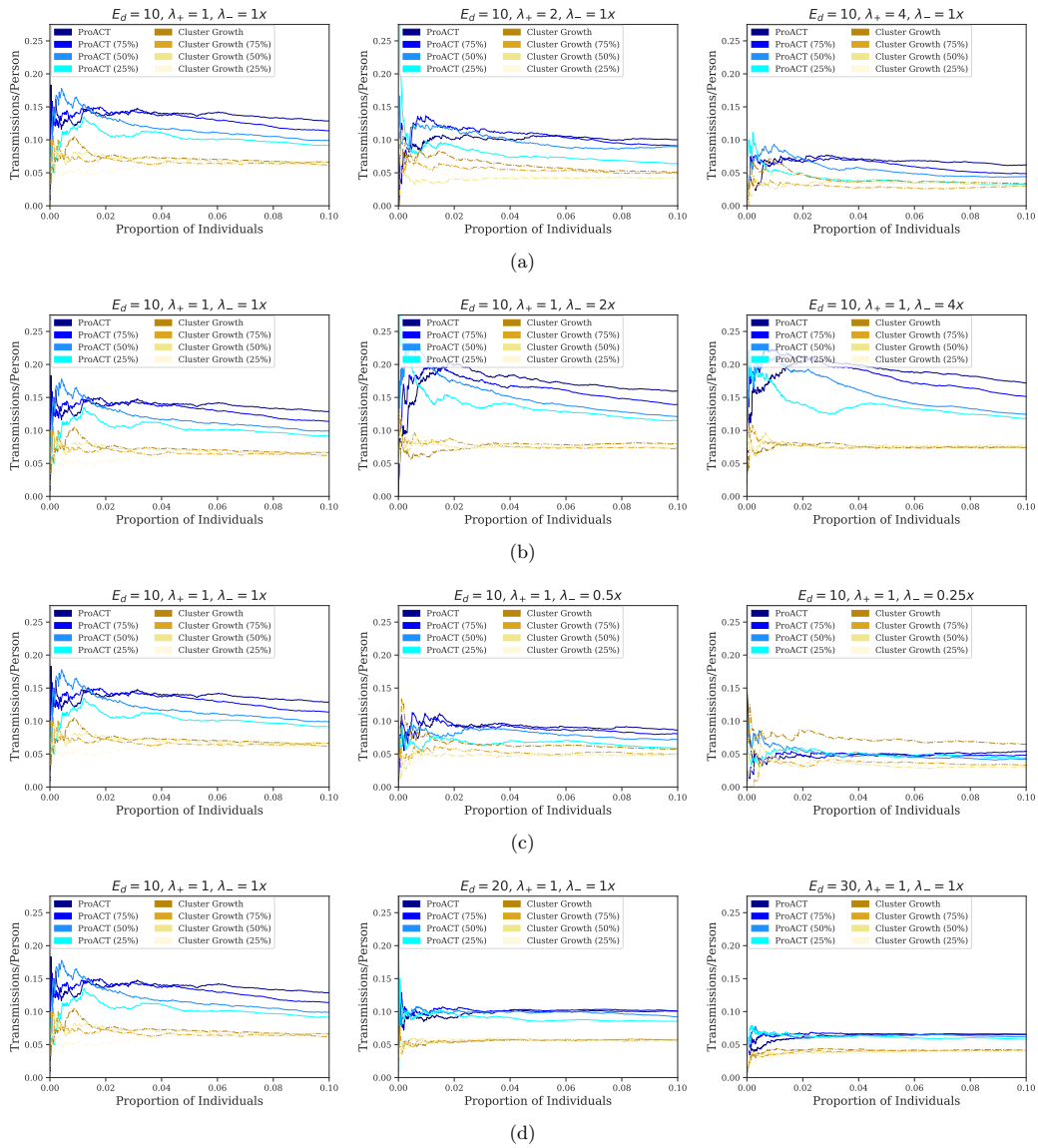
Percentile	GD + Cluster Growth	ProACT (FastTree)
10%	$1 \times 10^{-8}$	$2 \times 10^{-10}$
20%	$2 \times 10^{-11}$	$7 \times 10^{-9}$
30%	$6 \times 10^{-20}$	$3 \times 10^{-11}$
40%	$3 \times 10^{-24}$	$4 \times 10^{-18}$
50%	$2 \times 10^{-23}$	$9 \times 10^{-17}$
60%	$5 \times 10^{-17}$	$4 \times 10^{-20}$
70%	$3 \times 10^{-15}$	$7 \times 10^{-15}$
80%	$6 \times 10^{-11}$	$2 \times 10^{-12}$
90%	$4 \times 10^{-16}$	$1 \times 10^{-20}$



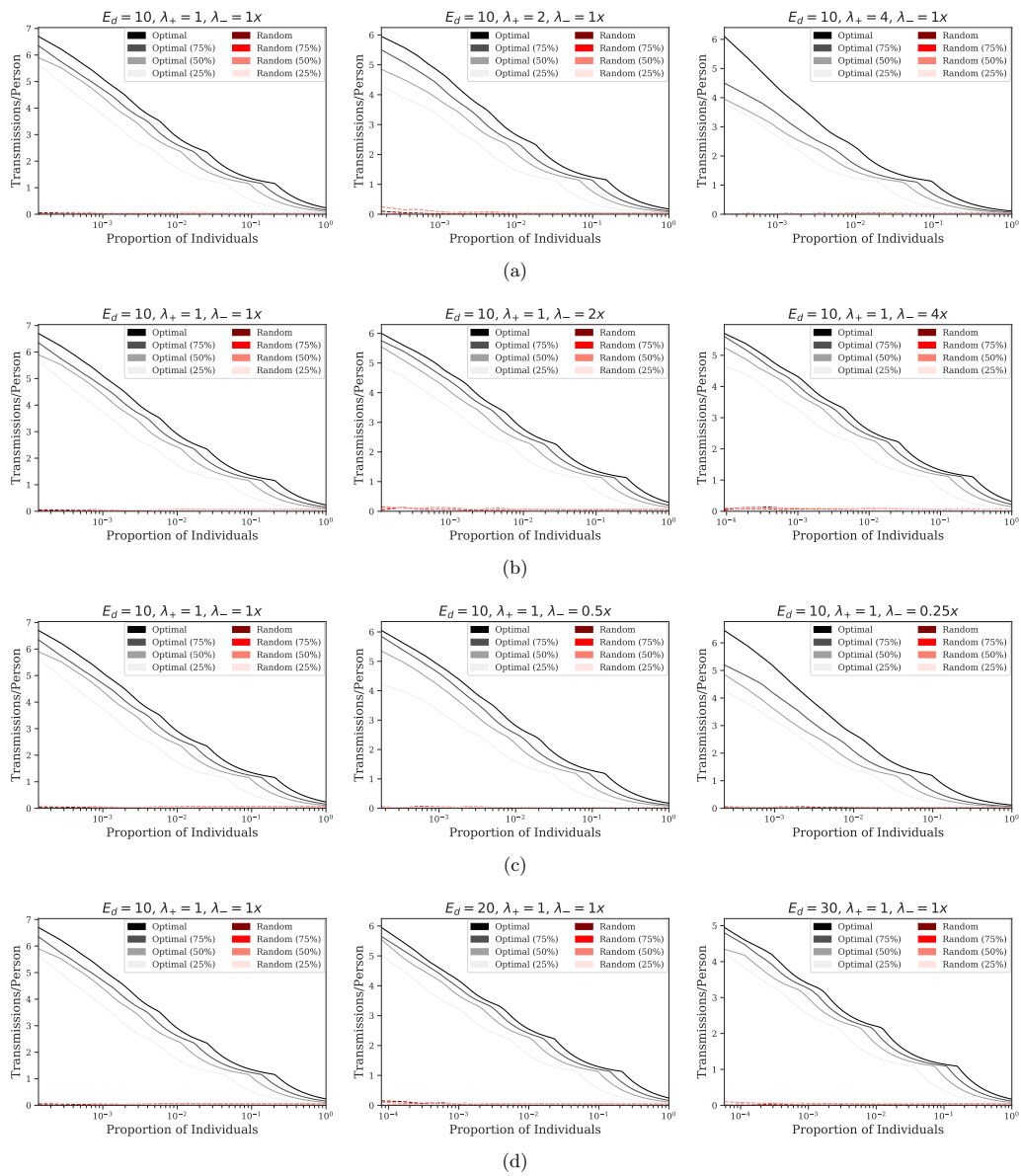
**Figure C.1:** As time progresses, the true incident branch length of each individual tends to decrease. This holds in inferred phylogenies as well (Fig. 3.1d).



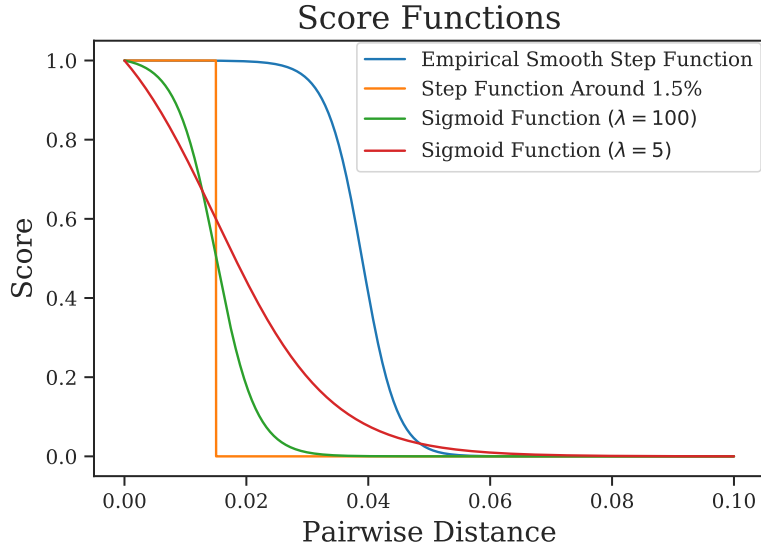
**Figure C.2:** Number of transmissions vs. incident branch lengths for individuals in a simulated epidemic. The epidemic was run for 10 years, samples were obtained at the 9-year mark, and a phylogeny was inferred using FastTree 2 [102] and subsequently MinVar-rooted [134]. Number of transmissions were measured between the 9-year and 10-year mark.



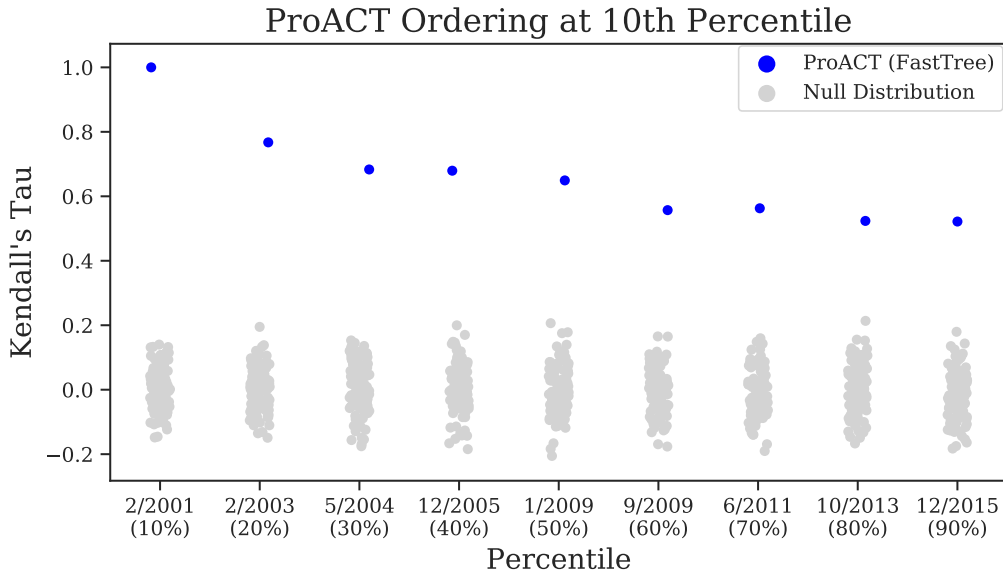
**Figure C.3:** Efficacy on datasets simulated using FAVITES. CMA of number of transmissions per person across all  $SH^+$ Is for each simulation parameter set.



**Figure C.4:** Efficacy of optimal and random selections on datasets simulated using FAVITES. CMA of number of transmissions per person across all SH<sup>+</sup>Is for each simulation parameter set.



**Figure C.5:** Score functions vs. pairwise sequence distance.

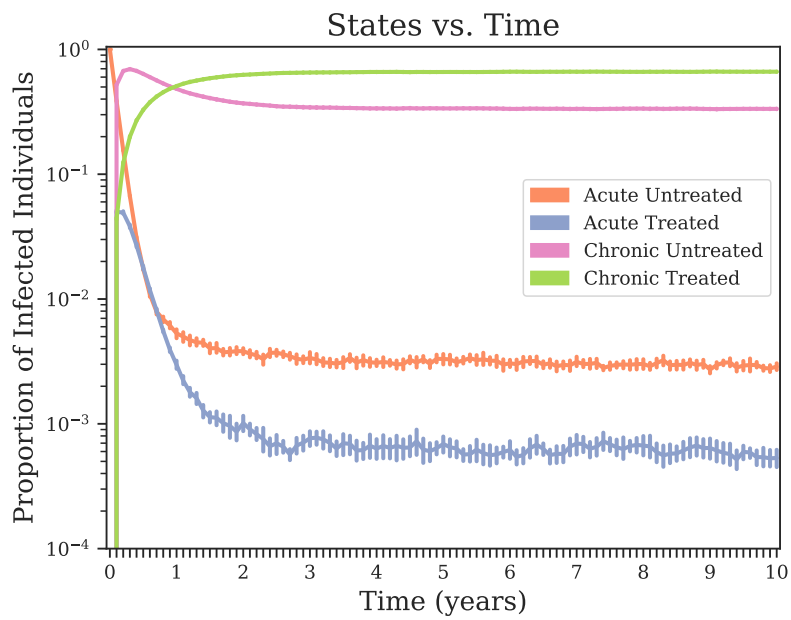


**Figure C.6:** Kendall's tau-b test results for ProACT ordering with respect to the ProACT ordering obtained with only the first decile of the dataset. The full San Diego dataset was split into two sets (*pre* and *post*) at each decile (shown on the horizontal axis). The individuals in *pre* were ordered using ProACT and by cluster growth. Kendall's tau-b correlation coefficient was computed for each ordering with respect to the ProACT ordering at the first decile. The null distribution was visualized by randomly shuffling the individuals in *pre*.



**Table C.2:** Default FAVITES simulation parameters.

<b>Parameter</b>	<b>Default Value</b>
Number of Contact Network Communities	20
Number of Individuals per Community	5,000
Mean Number of Edges Within Community	10
Mean Number of Edges Outside Community	1
Number of Seed Individuals	15,000
Seed Selection Model	Uniformly Random
Seed State Frequencies $\{AU, AT, CU, CT\}$	$\{0.0033, 0.0006, 0.3396, 0.6565\}$
Expected Transition Time AU→CU	6 weeks
Expected Transition Time AT→CT	12 weeks
Expected ART Initiation Time	1 year
Expected ART Termination Time	25 months
Rates of Infectiousness $\{AU, AT, CU, CT\}$	$\{0.1125, 0.005625, 0.0225, 0.000\}$
Seed Sequence Phylogenetic Model	Non-Homogeneous Yule Process
Seed Phylogeny Height	25 years
Seed Phylogeny Speciation Rate Function	$\exp(-t^2) + 1$
Mutation Rate Model	Truncated Normal
Mutation Rate Location	0.0008
Mutation Rate Scale	0.0005
Mutation Rate Minimum	0
Mutation Rate Maximum	$\infty$
Viral Sequence Type	HIV-1 Subtype B <i>pol</i>
Sequence Evolution Model	GTR+ $\Gamma$
GTR Frequencies $\{p_A, p_C, p_G, p_T\}$	$\{0.392, 0.165, 0.212, 0.232\}$
GTR Rates $\{\lambda_{AC}, \lambda_{AG}, \lambda_{AT}, \lambda_{CG}, \lambda_{CT}, \lambda_{GT}\}$	$\{1.766, 9.588, 0.692, 0.863, 10.283, 1.000\}$
GTR Gamma Distribution Shape	0.405
Viral Population Growth Rate Model	Logistic
Viral Population Growth Rate	2.851904
Initial Viral Population Size	1
Viral Population T50	-2
Number of Sampled Lineages per Person	1
Time of Sampling	ART Initiation



**Figure C.7:** Proportion of individuals in each infected state (AU, AT, CU, and CT) vs. time in simulations in which all seed individuals at time 0 were placed in state AU.

# Bibliography

- [1] R. D. Page and M. A. Charleston, “From Gene to Organismal Phylogeny: Reconciled Trees and the Gene Tree/Species Tree Problem,” *Molecular Phylogenetics and Evolution*, vol. 7, no. 2, pp. 231–240, 1997.
- [2] M. El-Kebir, G. Satas, L. Oesper, and B. J. Raphael, “Inferring the Mutational History of a Tumor Using Multi-state Perfect Phylogeny Mixtures,” *Cell Systems*, vol. 3, no. 1, pp. 43–53, 2016.
- [3] P. C. Nowell, “The clonal evolution of tumor cell populations,” *Science*, vol. 194, no. 4260, pp. 23–28, 1976.
- [4] G. Litman, J. Rast, M. Shablott, R. Haire, M. Hulst, W. Roess, R. Litman, K. Hinds-Frey, A. Zilch, and C. Amemiya, “Phylogenetic diversification of immunoglobulin genes and the antibody repertoire,” *Molecular Biology and Evolution*, vol. 10, no. 1, pp. 60–72, 1993.
- [5] W. H. Robinson, “Sequencing the functional antibody repertoire—diagnostic and therapeutic discovery,” *Nature Reviews Rheumatology*, vol. 11, no. 3, pp. 171–182, 2015.
- [6] Y. Safonova, S. Bonissone, E. Kurpilyansky, E. Starostina, A. Lapidus, J. Stinson, L. Depalatis, W. Sandoval, J. Lill, and P. A. Pevzner, “Ig Repertoire Constructor: A novel algorithm for antibody repertoire construction and immunoproteogenomics analysis,” *Bioinformatics*, vol. 31, no. 12, pp. i53–61, 2015.
- [7] J. A. Bailey and E. E. Eichler, “Primate segmental duplications: Crucibles of evolution, diversity and disease,” *Nature Reviews Genetics*, vol. 7, no. 7, pp. 552–564, 2006.
- [8] Z. Jiang, H. Tang, M. Ventura, M. F. Cardone, T. Marques-Bonet, X. She, P. A. Pevzner, and E. E. Eichler, “Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution,” *Nature Genetics*, vol. 39, no. 11, pp. 1361–1368, 2007.
- [9] M. Dewannieux, C. Esnault, and T. Heidmann, “LINE-mediated retrotransposition of marked Alu sequences,” *Nature Genetics*, vol. 35, no. 1, pp. 41–48, 2003.
- [10] N. Moshiri and S. Mirarab, “A two-state model of tree evolution and its applications to Alu retrotransposition,” *Systematic Biology*, vol. 67, no. 3, pp. 475–489, 2018.

- [11] S. D. W. Frost, M.-J. Dumaurier, S. Wain-Hobson, and A. J. L. Brown, “Genetic drift and within-host metapopulation dynamics of HIV-1 infection,” *Proceedings of the National Academy of Sciences*, vol. 98, pp. 6975–6980, jun 2001.
- [12] P. Lemey, A. Rambaut, and O. G. Pybus, “HIV evolutionary dynamics within and among hosts,” *AIDS Reviews*, vol. 8, pp. 125–140, jul 2006.
- [13] B. Vrancken, A. Rambaut, M. A. Suchard, A. Drummond, G. Baele, I. Derdelinckx, E. Van Wijngaerden, A. M. Vandamme, K. Van Laethem, and P. Lemey, “The Genealogical Population Dynamics of HIV-1 in a Large Transmission Chain: Bridging within and among Host Evolutionary Rates,” *PLoS Computational Biology*, vol. 10, no. 4, p. e1003505, 2014.
- [14] J. M. Carlson, M. Schaefer, D. C. Monaco, R. Batorsky, D. T. Claiborne, J. Prince, M. J. Deymier, Z. S. Ende, N. R. Klatt, C. E. Deziel, T.-H. Lin, J. Peng, A. M. Seese, R. Shapiro, J. Frater, T. Ndung’u, J. Tang, P. Goepfert, J. Gilmour, M. A. Price, W. Kilembe, D. Heckerman, P. J. R. Goulder, T. M. Allen, S. Allen, and E. Hunter, “Selection bias at the heterosexual HIV-1 transmission bottleneck,” *Science*, vol. 345, no. 6193, p. 1254031, 2014.
- [15] A. L. Rivas, F. O. Fasina, A. L. Hoogesteyn, S. N. Konah, J. L. Febles, D. J. Perkins, J. M. Hyman, J. M. Fair, J. B. Hittner, and S. D. Smith, “Connecting network properties of rapidly disseminating epizoonotics,” *PLoS ONE*, vol. 7, no. 6, p. e39778, 2012.
- [16] J. O. Wertheim, S. L. Kosakovsky Pond, S. J. Little, and V. De Gruttola, “Using HIV Transmission Networks to Investigate Community Effects in HIV Prevention Trials,” *PLoS ONE*, vol. 6, no. 11, p. e27775, 2011.
- [17] J. L. Aldous, S. K. Pond, A. Poon, S. Jain, H. Qin, J. S. Kahn, M. Kitahata, B. Rodriguez, A. M. Dennis, S. L. Boswell, R. Haubrich, and D. M. Smith, “Characterizing HIV transmission networks across the United States,” *Clinical Infectious Diseases*, vol. 55, no. 8, pp. 1135–1143, 2012.
- [18] B. Brenner, M. A. Wainberg, and M. Roger, “Phylogenetic inferences on HIV-1 transmission: Implications for the design of prevention and treatment interventions,” *Aids*, vol. 27, no. 7, pp. 1045–1057, 2013.
- [19] A. info, “HIV Treatment: The Basics,” 2019.
- [20] J. O. Wertheim, S. L. Kosakovsky Pond, L. A. Forgiione, S. R. Mehta, B. Murrell, S. Shah, D. M. Smith, K. Scheffler, and L. V. Torian, “Social and Genetic Networks of HIV-1 Transmission in New York City,” *PLoS Pathogens*, vol. 13, no. 1, p. e1006000, 2017.
- [21] M. Ragonnet-Cronin, Y. W. Hu, S. R. Morris, Z. Sheng, K. Poortinga, and J. O. Wertheim, “HIV transmission networks among transgender women in Los Angeles County, CA, USA: a phylogenetic analysis of surveillance data,” *The Lancet HIV*, vol. 6, pp. e164–e172, mar 2019.

- [22] R. Rose, S. L. Lamers, J. J. Dollar, M. K. Grabowski, E. B. Hodcroft, M. Ragonnet-Cronin, J. O. Wertheim, A. D. Redd, D. German, and O. Laeyendecker, “Identifying Transmission Clusters with Cluster Picker and HIV-TRACE,” *AIDS Research and Human Retroviruses*, vol. 33, no. 3, pp. 211–218, 2017.
- [23] M. C. Prosperi, M. Ciccozzi, I. Fanti, F. Saladini, M. Pecorari, V. Borghi, S. Di Giambenedetto, B. Bruzzone, A. Capetti, A. Vivarelli, S. Rusconi, M. C. Re, M. R. Gismondo, L. Sighinolfi, R. R. Gray, M. Salemi, M. Zazzi, and A. De Luca, “A novel methodology for large-scale phylogeny partition,” *Nature Communications*, vol. 2, no. 1, p. 321, 2011.
- [24] M. Ragonnet-Cronin, E. Hodcroft, S. Hué, E. Fearnhill, V. Delpech, A. J. Brown, and S. Lycett, “Automated analysis of phylogenetic clusters,” *BMC Bioinformatics*, vol. 14, no. 1, p. 317, 2013.
- [25] M. Balaban, N. Moshiri, U. Mai, and S. Mirarab, “TreeCluster: clustering biological sequences using phylogenetic trees,” *bioRxiv*, 2019.
- [26] S. L. Kosakovsky Pond, S. Weaver, A. J. Leigh Brown, and J. O. Wertheim, “HIV-TRACE (TRANsmiSSion Cluster Engine): a Tool for Large Scale Molecular Epidemiology of HIV-1 and Other Rapidly Evolving Pathogens,” *Molecular Biology and Evolution*, vol. 35, no. 7, pp. 1812–1819, 2018.
- [27] E. M. Campbell, H. Jia, A. Shankar, D. Hanson, W. Luo, S. Masciotra, S. M. Owen, A. M. Oster, R. R. Galang, M. W. Spiller, S. J. Blosser, E. Chapman, J. C. Roseberry, J. Gentry, P. Pontones, J. Duwve, P. Peyrani, R. M. Kagan, J. M. Whitcomb, P. J. Peters, W. Heneine, J. T. Brooks, and W. M. Switzer, “Detailed transmission network analysis of a large opiate-driven outbreak of HIV infection in the United States,” *Journal of Infectious Diseases*, vol. 216, no. 9, pp. 1053–1062, 2017.
- [28] O. Ratmann, E. B. Hodcroft, M. Pickles, A. Cori, M. Hall, S. Lycett, C. Colijn, B. Dearlove, X. Didelot, S. Frost, A. S. Md Mukarram Hossain, J. B. Joy, M. Kendall, D. Kuhnert, G. E. Leventhal, R. Liang, G. Plazzotta, A. F. Poon, D. A. Rasmussen, T. Stadler, E. Volz, C. Weis, A. J. Brown, and C. Fraser, “Phylogenetic tools for generalized HIV-1 epidemics: Findings from the PANGEA-HIV methods comparison,” *Molecular Biology and Evolution*, vol. 34, pp. 185–203, jan 2017.
- [29] G. U. Yule, “A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F. R. S,” *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, vol. 213, pp. 21–87, 1925.
- [30] D. J. Aldous, “Stochastic models and descriptive statistics for phylogenetic trees, from Yule to today,” *Statistical Science*, vol. 16, no. 1, pp. 23–34, 2001.
- [31] A. J. Drummond and A. Rambaut, “BEAST: Bayesian evolutionary analysis by sampling trees,” *BMC Evolutionary Biology*, vol. 7, no. 214, 2007.

- [32] A. Mooers, O. Gascuel, T. Stadler, H. Li, and M. Steel, “Branch lengths on birth-death trees and the expected loss of phylogenetic diversity,” *Systematic Biology*, vol. 61, no. 2, pp. 195–203, 2012.
- [33] E. Sayyari and S. Mirarab, “Fast Coalescent-Based Computation of Local Branch Support from Quartet Frequencies,” *Molecular Biology and Evolution*, vol. 33, no. 7, pp. 1654–1668, 2016.
- [34] C. Guyer and J. B. Slowinski, “Comparisons of Observed Phylogenetic Topologies with Null Expectations Among Three Monophyletic Lineages,” *Evolution*, vol. 45, no. 2, pp. 340–350, 1991.
- [35] M. Kirkpatrick and M. Slatkin, “Searching for Evolutionary Patterns in the Shape of a Phylogenetic Tree,” *Evolution*, vol. 47, no. 4, pp. 1171–1181, 1993.
- [36] P.-M. Agapow and A. Purvis, “Power of Eight Tree Shape Statistics to Detect Nonrandom Diversification: A Comparison by Simulation of Two Models of Cladogenesis,” *Systematic Biology*, vol. 51, no. 6, pp. 866–872, 2002.
- [37] H. Morlon, “Phylogenetic approaches for studying diversification,” *Ecology Letters*, vol. 17, no. 4, pp. 508–525, 2014.
- [38] F. D. Sahneh, C. Scoglio, and P. Van Mieghem, “Generalized epidemic mean-field model for spreading processes over multilayer complex networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1609–1620, 2013.
- [39] F. D. Sahneh, A. Vajdi, H. Shakeri, F. Fan, and C. Scoglio, “GEMFsim: A stochastic simulator for the generalized epidemic modeling framework,” *Journal of Computational Science*, vol. 22, pp. 36–44, 2017.
- [40] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [41] D. J. Watts, “Networks, Dynamics, and the Small World Phenomenon,” *American Journal of Sociology*, vol. 105, no. 2, pp. 493–527, 1999.
- [42] A. L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [43] P. Erdos and A. Rényi, “On Random Graphs I,” *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.
- [44] T. H. Jukes and C. R. Cantor, “Evolution of protein molecules,” *Mammalian Protein Metabolism*, pp. 21–123, 1969.
- [45] M. Kimura, “A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences,” *Journal of Molecular Evolution*, vol. 16, no. 2, pp. 111–120, 1980.

- [46] J. Felsenstein, “Evolutionary trees from DNA sequences: A maximum likelihood approach,” *Journal of Molecular Evolution*, vol. 17, no. 6, pp. 368–376, 1981.
- [47] K. Tamura and M. Nei, “Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees,” *Molecular Biology and Evolution*, vol. 10, no. 3, pp. 512–526, 1993.
- [48] S. Tavaré, “Some probabilistic and statistical problems in the analysis of DNA sequences,” in *American Mathematical Society: Lectures on Mathematics in the Life Sciences*, vol. 17, pp. 57–86, American Mathematical Society, 17 ed., 1986.
- [49] A. Rambaut and N. C. Grass, “Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees,” *Bioinformatics*, vol. 13, no. 3, pp. 235–238, 1997.
- [50] J. Felsenstein, *Inferring Phylogenies*. Sunderland: Sinauer Associates, Inc., 2003.
- [51] J. M. Whitcomb and S. H. Hughes, “Retroviral Reverse Transcription and Integration: Progress and Problems,” *Annual Review of Cell and Developmental Biology*, vol. 8, pp. 275–306, jan 1992.
- [52] N. Wood, T. Bhattacharya, B. F. Keele, E. Giorgi, M. Liu, B. Gaschen, M. Daniels, G. Ferrari, B. F. Haynes, A. McMichael, G. M. Shaw, B. H. Hahn, B. Korber, and C. Seoighe, “HIV evolution in early infection: Selection pressures, patterns of insertion and deletion, and the impact of APOBEC,” *PLoS Pathogens*, vol. 5, p. e1000414, may 2009.
- [53] J. F. Kingman, “On the Genealogy of Large Populations,” *Journal of Applied Probability*, vol. 19, pp. 27–43, 1982.
- [54] M. A. Batzer and P. L. Deininger, “Alu repeats and human genomic diversity,” *Nature Reviews Genetics*, vol. 3, no. 5, pp. 370–379, 2002.
- [55] A. L. Price, E. Eskin, and P. A. Pevzner, “Whole-genome analysis of Alu repeat elements reveals complex evolutionary history,” *Genome Research*, vol. 14, no. 11, pp. 2245–2252, 2004.
- [56] P. Deininger, “Alu elements: know the SINES,” *Genome Biology*, vol. 12, no. 12, p. 236, 2011.
- [57] R. Cordaux, D. J. Hedges, and M. A. Batzer, “Retrotransposition of Alu elements: how many sources?,” *Trends in Genetics*, vol. 20, no. 10, pp. 464–467, 2004.
- [58] S. J. Little, S. L. K. Pond, C. M. Anderson, J. A. Young, J. O. Wertheim, S. R. Mehta, S. May, and D. M. Smith, “Using HIV networks to inform real time prevention interventions,” *PLoS ONE*, vol. 9, no. 6, 2014.

- [59] J. A. Kelly, J. S. St. Lawrence, Y. E. Diaz, L. Y. Stevenson, A. C. Hauth, T. L. Brasfield, S. C. Kalichman, J. E. Smith, and M. E. Andrew, “HIV risk behavior reduction following intervention with key opinion leaders of population: An experimental analysis,” *American Journal of Public Health*, vol. 81, no. 2, pp. 168–171, 1991.
- [60] E. B. Shargie and B. Lindtjørn, “Determinants of Treatment Adherence Among Smear-Positive Pulmonary Tuberculosis Patients in Southern Ethiopia Methods and Findings,” *PLoS Medicine*, vol. 4, no. 2, pp. 0001–0008, 2007.
- [61] T. Leitner, D. Escanilla, C. Franzén, M. Uhlén, and J. Albert, “Accurate reconstruction of a known HIV-1 transmission history by phylogenetic tree analysis.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 20, pp. 10864–9, 1996.
- [62] R. J. Ypma, W. M. van Ballegooijen, and J. Wallinga, “Relating phylogenetic trees to transmission trees of infectious disease outbreaks,” *Genetics*, vol. 195, no. 3, pp. 1055–1062, 2013.
- [63] E. Romero-Severson, H. Skar, I. Bulla, J. Albert, and T. Leitner, “Timing and order of transmission events is not directly reflected in a pathogen phylogeny,” *Molecular Biology and Evolution*, vol. 31, no. 9, pp. 2472–2482, 2014.
- [64] T. Leitner and E. Romero-Severson, “Phylogenetic patterns recover known HIV epidemiological relationships and reveal common transmission of multiple variants,” *Nature Microbiology*, vol. 3, pp. 983–988, 2018.
- [65] M. K. Grabowski and A. D. Redd, “Molecular tools for studying HIV transmission in sexual networks,” *Current Opinion in HIV and AIDS*, vol. 9, no. 2, pp. 126–133, 2014.
- [66] J. O. Wertheim, A. J. Leigh Brown, N. L. Hepler, S. R. Mehta, D. D. Richman, D. M. Smith, and S. L. Kosakovsky Pond, “The global transmission network of HIV-1,” *Journal of Infectious Diseases*, vol. 209, no. 2, pp. 304–313, 2014.
- [67] L. Villandre, D. A. Stephens, A. Labbe, H. F. Günthard, R. Kouyos, T. Stadler, V. Aubert, M. Battegay, E. Bernasconi, J. Böni, H. C. Bucher, C. Burton-Jeangros, A. Calmy, M. Cavassini, G. Dollenmaier, M. Egger, L. Elzi, J. Fehr, J. Fellay, H. Furrer, C. A. Fux, M. Gorgievski, H. Günthard, D. Haerry, B. Hasse, H. H. Hirsch, M. Hoffmann, I. Höfli, C. Kahlert, L. Kaiser, O. Keiser, T. Klimkait, H. Kovari, B. Ledergerber, G. Martinetti, B. Martinez De Tejada, K. Metzner, N. Müller, D. Nadal, D. Nicca, G. Pantaleo, A. Rauch, S. Regenass, M. Rickenbach, C. Rudin, F. Schöni-Affolter, P. Schmid, J. Schüpbach, R. Speck, P. Tarr, A. Telenti, A. Trkola, P. Vernazza, R. Weber, and S. Yerly, “Assessment of overlap of phylogenetic transmission clusters and communities in simple sexual contact networks: Applications to HIV-1,” *PLoS ONE*, vol. 11, no. 2, p. e0148459, 2016.
- [68] C. Groendyke, D. Welch, and D. R. Hunter, “A Network-based Analysis of the 1861 Hagelloch Measles Data,” *Biometrics*, vol. 68, no. 3, pp. 755–765, 2012.



- [69] T. Stadler and S. Bonhoeffer, “Uncovering epidemiological dynamics in heterogeneous host populations using phylogenetic methods,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 368, no. 1614, p. 20120198, 2013.
- [70] T. Jombart, A. Cori, X. Didelot, S. Cauchemez, C. Fraser, and N. Ferguson, “Bayesian Reconstruction of Disease Outbreaks by Combining Epidemiologic and Genomic Data,” *PLoS Computational Biology*, vol. 10, no. 1, p. e1003457, 2014.
- [71] C. J. Worby and T. D. Read, “‘SEEDY’ (Simulation of Evolutionary and Epidemiological Dynamics): An R package to follow accumulation of within-host mutation in pathogens,” *PLoS ONE*, vol. 10, no. 6, p. e0129745, 2015.
- [72] M. Karoński, “A review of random graphs,” *Journal of Graph Theory*, vol. 6, no. 4, pp. 349–389, 1982.
- [73] M. Newman, D. Watts, and S. Strogatz, “Random graph models of social networks.,” *PNAS*, vol. 99 Suppl 1, no. Suppl 1, pp. 2566–72, 2002.
- [74] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [75] B. Bollobas, “The Evolution of Random Graphs,” *Transactions of the American Mathematical Society*, vol. 286, no. 1, p. 257, 1984.
- [76] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference*, no. SciPy, (Pasadena), pp. 11–15, 2008.
- [77] S. Eddy, “Profile hidden Markov models.,” *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.
- [78] J. Sukumaran and M. T. Holder, “DendroPy: A Python library for phylogenetic computing,” *Bioinformatics*, vol. 26, no. 12, pp. 1569–1571, 2010.
- [79] R. M. Granich, C. F. Gilks, C. Dye, K. M. De Cock, and B. G. Williams, “Universal voluntary HIV testing with immediate antiretroviral therapy as a strategy for elimination of HIV transmission: a mathematical model,” *The Lancet*, vol. 373, no. 9657, pp. 48–57, 2009.
- [80] A. Cori, H. Ayles, N. Beyers, A. Schaap, S. Floyd, K. Sabapathy, J. W. Eaton, K. Hauck, P. Smith, S. Griffith, A. Moore, D. Donnell, S. H. Vermund, S. Fidler, R. Hayes, and C. Fraser, “HPTN 071 (PopART): A cluster-randomized trial of the population impact of an HIV combination prevention intervention including universal testing and treatment: Mathematical model,” *PLoS ONE*, vol. 9, no. 1, p. e84511, 2014.
- [81] K. Hartmann, D. Wong, and T. Stadler, “Sampling trees from evolutionary models,” *Systematic Biology*, vol. 59, no. 4, pp. 465–476, 2010.

- [82] Z. Yang, “Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods,” *Journal of Molecular Evolution*, vol. 39, no. 3, pp. 306–314, 1994.
- [83] M. Zaheri, L. Dib, and N. Salamin, “A generalized mechanistic codon model,” *Molecular Biology and Evolution*, vol. 31, no. 9, pp. 2528–2541, 2014.
- [84] C. Kosiol, I. Holmes, and N. Goldman, “An empirical codon model for protein sequence evolution,” *Molecular Biology and Evolution*, vol. 24, no. 7, pp. 1464–1479, 2007.
- [85] S. J. Spielman and C. O. Wilke, “Pyvolve: A flexible python module for simulating sequences along phylogenies,” *PLoS ONE*, vol. 10, no. 9, p. e0139047, 2015.
- [86] W. Huang, L. Li, J. R. Myers, and G. T. Marth, “ART: A next-generation sequencing read simulator,” *Bioinformatics*, vol. 28, no. 4, pp. 593–594, 2012.
- [87] F. E. Angly, D. Willner, F. Rohwer, P. Hugenholtz, and G. W. Tyson, “Grinder: A versatile amplicon and shotgun sequence simulator,” *Nucleic Acids Research*, vol. 40, no. 12, p. e94, 2012.
- [88] E. S. Rosenberg, P. S. Sullivan, E. A. Dinunno, L. F. Salazar, and T. H. Sanchez, “Number of casual male sexual partners and associated factors among men who have sex with men: Results from the National HIV Behavioral Surveillance system,” *BMC Public Health*, vol. 11, no. 189, 2011.
- [89] D. T. Hamilton, M. S. Handcock, and M. Morris, “Degree distributions in sexual networks: A framework for evaluating evidence,” *Sexually Transmitted Diseases*, vol. 35, no. 1, pp. 30–40, 2008.
- [90] N. Macchione, W. J. Wooten, K. Waters-Montijo, E. McDonald, M. Bursaw, L. Freitas, S. Tweeten, E. Awa, F. McGann, M. Johnson, and S. Hunter, “HIV/AIDS Epidemiology Report,” *County of San Diego Health and Human Services Agency Public Health Services*, 2015.
- [91] S. E. Bellan, J. Dushoff, A. P. Galvani, and L. A. Meyers, “Reassessment of HIV-1 Acute Phase Infectivity: Accounting for Heterogeneity and Study Design with Simulated Cohorts,” *PLoS Medicine*, vol. 12, no. 3, p. e1001801, 2015.
- [92] M. S. Cohen, Y. Q. Chen, M. McCauley, T. Gamble, M. C. Hosseinipour, N. Kumarasamy, J. G. Hakim, J. Kumwenda, B. Grinsztejn, J. H. Pilotto, S. V. Godbole, S. Mehendale, S. Chariyalertsak, B. R. Santos, K. H. Mayer, I. F. Hoffman, S. H. Eshleman, E. Piwowar-Manning, L. Wang, J. Makhema, L. A. Mills, G. de Bruyn, I. Sanne, J. Eron, J. Gallant, D. Havlir, S. Swindells, H. Ribaudou, V. Elharrar, D. Burns, T. E. Taha, K. Nielsen-Saines, D. Celentano, M. Essex, and T. R. Fleming, “Prevention of HIV-1 Infection with Early Antiretroviral Therapy,” *New England Journal of Medicine*, vol. 365, no. 6, pp. 493–505, 2011.

- [93] M. O'Brien and M. Markowitz, "Should we treat acute HIV infection?," *Current HIV/AIDS Reports*, vol. 9, no. 2, pp. 101–110, 2012.
- [94] B. Nosyk, L. Lourenço, J. E. Min, D. Shopin, V. D. Lima, and J. S. Montaner, "Characterizing retention in HAART as a recurrent event process: Insights into 'cascade churn'," *Aids*, vol. 29, no. 13, pp. 1681–1689, 2015.
- [95] M. Wawer, R. Gray, N. Sewankambo, D. Serwadda, X. Li, O. Laeyendecker, N. Kiwanuka, G. Kigozi, M. Kiddugavu, T. Lutalo, F. Nalugoda, F. Wabwire-Mangen, M. Meehan, and T. Quinn, "Rates of HIV-1 Transmission per Coital Act, by Stage of HIV-1 Infection, in Rakai, Uganda," *The Journal of Infectious Diseases*, vol. 191, no. 9, pp. 1403–1409, 2005.
- [96] S. Kalyaanamoorthy, B. Q. Minh, T. K. Wong, A. Von Haeseler, and L. S. Jermin, "ModelFinder: Fast model selection for accurate phylogenetic estimates," *Nature Methods*, vol. 14, no. 6, pp. 587–589, 2017.
- [97] O. Chernomor, A. Von Haeseler, and B. Q. Minh, "Terrace Aware Data Structure for Phylogenomic Inference from Supermatrices," *Systematic Biology*, vol. 65, no. 6, pp. 997–1008, 2016.
- [98] T. H. To, M. Jung, S. Lycett, and O. Gascuel, "Fast Dating Using Least-Squares Criteria and Algorithms," *Systematic Biology*, vol. 65, no. 1, pp. 82–97, 2016.
- [99] Y. Le Gat, *Recurrent Event Modeling Based on the Yule Process, Volume 2*. London: ISTE Ltd, 2016.
- [100] N. McCreesh, I. Andrianakis, R. N. Nsubuga, M. Strong, I. Vernon, T. J. McKinley, J. E. Oakley, M. Goldstein, R. Hayes, and R. G. White, "Universal test, treat, and keep: Improving ART retention is key in cost-effective HIV control in Uganda," *BMC Infectious Diseases*, vol. 17, no. 1, p. 322, 2017.
- [101] M. Pérez-Losada, A. D. Castel, B. Lewis, M. Kharfen, C. P. Cartwright, B. Huang, T. Maxwell, A. E. Greenberg, and K. A. Crandall, "Characterization of HIV diversity, phylodynamics and drug resistance in Washington, DC," *PLoS ONE*, vol. 12, no. 9, p. e0185644, 2017.
- [102] M. N. Price, P. S. Dehal, and A. P. Arkin, "FastTree 2 - Approximately maximum-likelihood trees for large alignments," *PLoS ONE*, vol. 5, no. 3, 2010.
- [103] J. O. Wertheim, B. Murrell, S. R. Mehta, L. A. Forgiione, S. L. Kosakovsky Pond, D. M. Smith, and L. V. Torian, "Growth of HIV-1 Molecular Transmission Clusters in New York City," *The Journal of Infectious Diseases*, vol. 218, no. 12, pp. 1943–1953, 2018.
- [104] J. Lin, "Divergence Measures Based on the Shannon Entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

- [105] UNAIDS, “90-90-90 An ambitious treatment target to help end the AIDS epidemic,” tech. rep., UNAIDS, Geneva, Switzerland, 2014.
- [106] D. F. Robinson and L. R. Foulds, “Comparison of phylogenetic trees,” *Mathematical Biosciences*, vol. 53, no. 1-2, pp. 131–147, 1981.
- [107] T. Azarian, A. Ali, J. A. Johnson, D. Mohr, M. Prosperi, N. M. Veras, M. Jubair, S. L. Strickland, M. H. Rashid, M. T. Alam, T. A. Weppelmann, L. S. Katz, C. L. Tarr, R. R. Colwell, J. G. Morris, and M. Salemi, “Phylogenetic analysis of clinical and environmental *Vibrio cholera* isolates from Haiti reveals diversification driven by positive selection,” *mBio*, vol. 5, no. 6, pp. e01824–14, 2014.
- [108] R. D. Finn, A. Bateman, J. Clements, P. Coghill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. Sonnhammer, J. Tate, and M. Punta, “Pfam: The protein families database,” *Nucleic Acids Research*, vol. 42, pp. D222–230, jan 2014.
- [109] C. A. Hinde, R. A. Johnstone, and R. M. Kilner, “Parent-offspring conflict and coadaptation,” *Science*, vol. 327, pp. 1373–1376, mar 2010.
- [110] J. K. M. Brown, “Probabilities of Evolutionary Trees,” *Systematic Biology*, vol. 43, no. 1, pp. 78–91, 1994.
- [111] D. Aldous, “Probability Distributions on Cladograms,” *Random Discrete Structures*, no. 143, pp. 1–18, 1996.
- [112] M. Steel and A. McKenzie, “Properties of phylogenetic trees generated by yule-type speciation models,” *Mathematical Biosciences*, vol. 170, no. 1, pp. 91–112, 2001.
- [113] D. J. Ford, “Probabilities on cladograms: introduction to the alpha model,” *arXiv*, no. math/0511246, pp. 1–75, 2005.
- [114] M. G. B. Blum and O. François, “Which random processes describe the tree of life? A large-scale study of phylogenetic tree imbalance,” *Systematic Biology*, vol. 55, no. 4, pp. 685–691, 2006.
- [115] G. Jones, “Calculations for multi-type age-dependent binary branching processes,” *Journal of Mathematical Biology*, vol. 63, no. 1, pp. 33–56, 2011.
- [116] W. P. Maddison, P. E. Midford, and S. P. Otto, “Estimating a binary character’s effect on speciation and extinction,” *Systematic Biology*, vol. 56, no. 5, pp. 701–710, 2007.
- [117] A. McKenzie and M. Steel, “Distributions of cherries for two models of trees,” *Mathematical Biosciences*, vol. 164, no. 1, pp. 81–92, 2000.
- [118] E. M. Volz, K. Koelle, and T. Bedford, “Viral Phylodynamics,” *PLoS Computational Biology*, vol. 9, no. 3, p. e1002947, 2013.

- [119] A. Lambert and T. Stadler, “Birth-death models and coalescent point processes: The shape and probability of reconstructed phylogenies,” *Theoretical Population Biology*, vol. 90, no. 90, pp. 113–128, 2013.
- [120] C. W. Schmid, “Alu: a parasite’s parasite?,” *Nature Genetics*, vol. 35, no. 1, pp. 15–16, 2003.
- [121] P. L. Deininger and M. A. Batzer, “Alu repeats and human disease,” *Mol Genet Metab*, vol. 67, no. 3, pp. 183–193, 1999.
- [122] M. Stoneking, J. J. Fontius, S. L. Clifford, H. Soodyall, S. S. Arcot, N. Saha, T. Jenkins, M. A. Tahir, P. L. Deininger, and M. A. Batzer, “Alu Insertion Polymorphisms and Human Evolution: Evidence for a Larger Population Size in Africa,” *Genome Research*, vol. 7, no. 11, pp. 1061–1071, 1997.
- [123] S. S. Singer, J. Schmitz, C. Schwiegk, and H. Zischler, “Molecular cladistic markers in New World monkey phylogeny (Platyrrhini, Primates),” *Molecular Phylogenetics and Evolution*, vol. 26, no. 3, pp. 490–501, 2003.
- [124] W. S. Watkins, A. R. Rogers, C. T. Ostler, S. Wooding, M. J. Bamshad, A. M. E. Brassington, M. L. Carroll, S. V. Nguyen, J. A. Walker, B. V. Prasad, P. G. Reddy, P. K. Das, M. A. Batzer, and L. B. Jorde, “Genetic variation among world populations: Inferences from 100 Alu insertion polymorphisms,” 2003.
- [125] G. E. Liu, C. Alkan, L. Jiang, S. Zhao, and E. E. Eichler, “Comparative analysis of Alu repeats in primate genomes,” *Genome Research*, vol. 19, no. 5, pp. 876–885, 2009.
- [126] M. K. Konkel, J. A. Walker, A. B. Hotard, M. C. Ranck, C. C. Fontenot, J. Storer, C. Stewart, G. T. Marth, and M. A. Batzer, “Sequence analysis and characterization of active human alu subfamilies based on the 1000 genomes pilot project,” *Genome Biology and Evolution*, vol. 7, no. 9, pp. 2608–2622, 2015.
- [127] W. Fletcher and Z. Yang, “INDELible: A flexible simulator of biological sequence evolution,” *Molecular Biology and Evolution*, vol. 26, no. 8, pp. 1879–1888, 2009.
- [128] A. Stamatakis, “RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies,” *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.
- [129] D. Bogdanowicz, K. Giaro, and B. Wróbel, “TreeCmp: Comparison of trees in polynomial time,” *Evolutionary Bioinformatics*, pp. 475–487, aug 2012.
- [130] R. Hubley, R. D. Finn, J. Clements, S. R. Eddy, T. A. Jones, W. Bao, A. F. Smit, and T. J. Wheeler, “The Dfam database of repetitive DNA families,” *Nucleic Acids Research*, vol. 44, no. D1, pp. D81–D89, 2016.
- [131] S. Mirarab, N. Nguyen, S. Guo, L.-S. Wang, J. Kim, and T. Warnow, “PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences.,” *Journal of Computational Biology*, vol. 22, no. 5, pp. 377–86, 2015.

- [132] M. K. Kuhner and J. Felsenstein, “A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates,” *Molecular Biology and Evolution*, vol. 11, pp. 459–468, feb 1994.
- [133] B. Kolaczkowski and J. W. Thornton, “Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous,” *Nature*, vol. 431, pp. 980–984, oct 2004.
- [134] U. Mai, E. Sayyari, and S. Mirarab, “Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction,” *PLoS ONE*, vol. 12, no. 8, 2017.
- [135] F. A. Matsen, “A Geometric Approach to Tree Shape Statistics,” *Systematic Biology*, vol. 55, pp. 652–661, aug 2006.
- [136] F. A. Matsen, “Optimization Over a Class of Tree Shape Statistics,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, pp. 506–512, jul 2007.
- [137] J. Wang, L. Song, M. K. Gonder, S. Azrak, D. A. Ray, M. A. Batzer, S. A. Tishkoff, and P. Liang, “Whole genome computational comparative genomics: A fruitful approach for ascertaining Alu insertion polymorphisms,” in *Gene*, vol. 365, pp. 11–20, 2006.
- [138] A. C. Wacholder and D. Pollock, “Strong and Fluctuating Sequence Constraints Drive Alu Evolution,” *bioRxiv*, 2016.
- [139] A. D. R. N. Schneeberger, C. H. Mercer, S. A. Gregson, N. M. Ferguson, C. A. Nyamukapa, R. M. Anderson, A. M. Johnson, and G. P. Garnett, “Scale-free networks and sexually transmitted diseases: a description of observed patterns of sexual contacts in Britain and Zimbabwe,” *Sexually Transmitted Diseases*, vol. 31, no. 6, pp. 380–387, 2004.
- [140] R. Bunnell, J. P. Ekwaru, P. Solberg, N. Wamai, W. Bikaako-Kajura, W. Were, A. Coutinho, C. Liechty, E. Madraa, G. Rutherford, and J. Mermin, “Changes in sexual behavior and risk of HIV transmission after antiretroviral therapy and prevention interventions in rural Uganda,” *AIDS*, vol. 20, no. 1, pp. 85–92, 2006.
- [141] N. Bbosa, D. Ssemwanga, R. N. Nsubuga, J. F. Salazar-Gonzalez, M. G. Salazar, M. Nanyonjo, M. Kuteesa, J. Seeley, N. Kiwanuka, B. S. Bagaya, G. Yebra, A. Leigh-Brown, and P. Kaleebu, “Phylogeography of HIV-1 suggests that Ugandan fishing communities are a sink for, not a source of, virus from general populations,” *Scientific Reports*, vol. 9, p. 1051, dec 2019.
- [142] L. Villandr e, A. Labbe, B. Brenner, R. I. Ibanescu, M. Roger, and D. A. Stephens, “Assessing the role of transmission chains in the spread of HIV-1 among men who have sex with men in Quebec, Canada,” *PLoS ONE*, vol. 14, p. e0213366, mar 2019.
- [143] A. M. Oster, A. M. France, N. Panneer, M. Cheryl Bañez Ocfemia, E. Campbell, S. Dasgupta, W. M. Switzer, J. O. Wertheim, and A. L. Hernandez, “Identifying Clusters of Recent and Rapid HIV Transmission Through Analysis of Molecular Surveillance Data,” *Journal of Acquired Immune Deficiency Syndromes*, vol. 79, no. 5, pp. 543–550, 2018.

- [144] D. M. Smith, S. J. May, S. Tweeten, L. Drumright, M. E. Pacold, S. L. Kosakovsky Pond, R. L. Pesano, Y. S. Lie, D. D. Richman, S. D. Frost, C. H. Woelk, and S. J. Little, “A public health model for the molecular surveillance of HIV transmission in San Diego, California,” *AIDS*, vol. 23, pp. 225–232, jan 2009.
- [145] N. Moshiri, M. Ragonnet-Cronin, J. O. Wertheim, and S. Mirarab, “FAVITES: simultaneous simulation of transmission networks, phylogenetic trees, and sequences,” *Bioinformatics*, p. bt921, 2018.
- [146] M. G. Kendall, “A New Measure of Rank Correlation,” *Biometrika*, vol. 30, no. 1-2, pp. 81–93, 1938.
- [147] E. O. Romero-Severson, I. Bulla, and T. Leitner, “Phylogenetically resolving epidemiologic linkage,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 10, pp. 2690–2695, 2016.
- [148] L. T. Nguyen, H. A. Schmidt, A. Von Haeseler, and B. Q. Minh, “IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies,” *Molecular Biology and Evolution*, vol. 32, no. 1, pp. 268–274, 2015.
- [149] T. I. Vasylyeva, M. Liulchuk, S. R. Friedman, I. Sazonova, N. R. Faria, A. Katzourakis, N. Babii, A. Scherbinska, J. Thézé, O. G. Pybus, P. Smyrnov, J. L. Mbisa, D. Paraskevis, A. Hatzakis, and G. Magiorkinis, “Molecular epidemiology reveals the role of war in the spread of HIV in Ukraine,” *Proceedings of the National Academy of Sciences*, vol. 115, pp. 1051–1056, jan 2018.
- [150] J. W. Mellors, C. R. Rinaldo, P. Gupta, R. M. White, J. A. Todd, and L. A. Kingsley, “Prognosis in HIV-1 infection predicted by the quantity of virus in plasma,” *Science*, vol. 272, pp. 1167–1170, may 1996.
- [151] R. B. Rothenberg, J. J. Potterat, D. E. Woodhouse, S. Q. Muth, W. W. Darrow, and A. S. Klovdahl, “Social network dynamics and HIV transmission,” *AIDS*, vol. 12, no. 12, pp. 1529–1536, 1998.
- [152] N. Moshiri, “TreeN93: a non-parametric distance-based method for inferring viral transmission clusters,” *bioRxiv*, 2018.
- [153] S. W. Kembel, J. A. Eisen, K. S. Pollard, and J. L. Green, “The phylogenetic diversity of metagenomes,” *PLoS ONE*, vol. 6, no. 8, p. e23214, 2011.
- [154] A. E. Darling, G. Jospin, E. Lowe, F. A. Matsen, H. M. Bik, and J. A. Eisen, “PhyloSift: phylogenetic analysis of genomes and metagenomes,” *PeerJ*, vol. 2, p. e243, 2014.
- [155] A. Filipski, K. Tamura, P. Billing-Ross, O. Murillo, and S. Kumar, “Phylogenetic placement of metagenomic reads using the minimum evolution principle,” *BMC Genomics*, vol. 16, no. Supplement 1, p. S13, 2015.

- [156] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. De Hoon, “Biopython: Freely available Python tools for computational molecular biology and bioinformatics,” *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009.
- [157] J. Huerta-Cepas, F. Serra, and P. Bork, “ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data,” *Molecular Biology and Evolution*, vol. 33, no. 6, pp. 1635–1638, 2016.
- [158] M. J. Phillips and D. Penny, “The root of the mammalian tree inferred from whole mitochondrial genomes,” *Molecular Phylogenetics and Evolution*, vol. 28, no. 2, pp. 171–185, 2003.
- [159] O. G. Pybus and P. H. Harvey, “Testing macro-evolutionary models using incomplete molecular phylogenies,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 267, no. 1459, pp. 2267–2272, 2000.
- [160] P. H. Harvey, R. M. May, and S. Nee, “Phylogenies Without Fossils,” *Evolution*, vol. 48, no. 3, pp. 523–529, 1994.
- [161] M. L. Metzker, “Sequencing technologies - the next generation.,” *Nature Reviews Genetics*, vol. 11, pp. 31–46, jan 2010.
- [162] P. Compeau, “Establishing a computational biology flipped classroom,” *PLoS Computational Biology*, vol. 15, p. e1006764, may 2019.
- [163] N. Mulder, R. Schwartz, M. D. Brazas, C. Brooksbank, B. Gaeta, S. L. Morgan, M. A. Pauley, A. Rosenwald, G. Rustici, M. Sierk, T. Warnow, and L. Welch, “The development and application of bioinformatics core competencies to improve bioinformatics training and education,” *PLoS Computational Biology*, vol. 14, p. e1005772, feb 2018.
- [164] A. Madlung, “Assessing an effective undergraduate module teaching applied bioinformatics to biology students,” *PLoS Computational Biology*, vol. 14, p. e1005872, jan 2018.
- [165] J. Wetzel, D. O’Toole, and S. Peterson, “An Analysis of Student Enrollment Demand,” *Economics of Education Review*, vol. 17, no. 1, pp. 47–54, 1998.
- [166] T. Camp, W. R. Adrion, B. Bizot, S. Davidson, M. Hall, S. Hambrusch, E. Walker, and S. Zweben, “Generation CS: The Growth of Computer Science,” *ACM Inroads*, vol. 8, no. 2, pp. 44–50, 2017.
- [167] A. Ng, “Machine Learning,” 2012.
- [168] L. Pappano, “The Year of the MOOC,” nov 2012.
- [169] L. Guàrdia, M. Maina, and A. Sangrà, “MOOC design principles: A pedagogical approach from the learner’s perspective,” *eLearning Papers*, vol. 33, no. 4, pp. 1–6, 2013.



- [170] D. O. Bruff, D. H. Fisher, K. E. Mcewen, and B. E. Smith, “Wrapping a MOOC: Student Perceptions of an Experiment in Blended Learning,” *MERLOT Journal of Online Learning and Teaching*, vol. 9, no. 2, pp. 187–199, 2013.
- [171] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton, “Studying Learning in the Worldwide Classroom Research into edX’s First MOOC,” *Research & Practice in Assessment*, vol. 8, no. 1, pp. 13–25, 2013.
- [172] P. J. Guo, J. Kim, and R. Rubin, “How video production affects student engagement: an empirical study of MOOC videos,” in *Proceedings of the first ACM conference on Learning @ scale conference - L@S ’14*, (New York, New York, USA), pp. 41–50, ACM Press, 2014.
- [173] R. Y. Bayeck, “Exploratory study of MOOC learners’ demographics and motivation: The case of students involved in groups,” *Open Praxis*, vol. 8, pp. 223–233, aug 2016.
- [174] C. Milligan and A. Littlejohn, “Why Study on a MOOC? The Motives of Students and Professionals,” *International Review of Research in Open and Distributed Learning*, vol. 18, no. 2, 2017.
- [175] M. Y. Vardi, “Will MOOCs Destroy Academia?,” *Communications of the ACM*, vol. 55, no. 11, p. 5, 2012.
- [176] P. Compeau and P. A. Pevzner, *Bioinformatics Algorithms: An Active Learning Approach*. Active Learning Publishers, 2014.
- [177] P. Compeau and P. A. Pevzner, “Life after MOOCs,” *Communications of the ACM*, vol. 58, no. 10, pp. 41–44, 2015.
- [178] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, *Taxonomy of educational objectives: The classification of educational goals*. New York: Addison-Wesley Longman Ltd, 1956.
- [179] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Rath, and M. C. Wittrock, *A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives*. Pearson, 2001.
- [180] C. C. Bonwell and J. A. Eison, “Active Learning: Creating Excitement in the Classroom,” *ASHE-ERIC Higher Education Reports*, p. 121, 1991.
- [181] M. Pedaste, M. Mäeots, L. A. Siiman, T. de Jong, S. A. van Riesen, E. T. Kamp, C. C. Manoli, Z. C. Zacharia, and E. Tsourlidaki, “Phases of inquiry-based learning: Definitions and the inquiry cycle,” *Educational Research Review*, vol. 14, pp. 47–61, feb 2015.
- [182] J. S. Bruner, “The Act of Discovery,” *Harvard Educational Review*, vol. 31, pp. 21–32, 1961.

- [183] N. Moshiri, P. Compeau, and P. Pevzner, “Analyze Your Genome!,” 2017.
- [184] N. Moshiri, L. Izhikevich, and C. Alvarado, “Data Structures: An Active Learning Approach,” 2017.
- [185] N. Moshiri and L. Izhikevich, *Design and Analysis of Data Structures*. Amazon Kindle Direct Publishing, 2018.
- [186] R. T. Smythe, “Central limit theorems for urn models,” *Stochastic Processes and their Applications*, vol. 65, no. 1, pp. 115–137, 1996.
- [187] G. E. Crooks, G. Hon, J. M. Chandonia, and S. E. Brenner, “WebLogo: A sequence logo generator,” *Genome Research*, vol. 14, no. 6, pp. 1188–1190, 2004.