

© 2024 Leah Leslie Weber

ALGORITHMS FOR SINGLE-CELL PHYLOGENETICS TO STUDY TUMOR
EVOLUTION AND ADAPTIVE IMMUNE RESPONSE.

BY

LEAH LESLIE WEBER

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Doctoral Committee:

Assistant Professor Mohammed El-Kebir, Chair and Director of Research
Professor Tandy Warnow
Professor Olga Milenkovic
Professor Nicholas Navin, MD Anderson Cancer Center

ABSTRACT

Molecular phylogenetics, a field at the intersection of biology and computer science, utilizes molecular data such as DNA, RNA, and protein sequences to infer evolutionary relationships among various biological entities. This thesis focuses on advancing the study of microevolutionary processes in cellular populations by leveraging the capabilities of single-cell sequencing technologies. Microevolutionary processes are characterized by high mutation rates, rapid population dynamics, and strong selective pressures, distinguishing them from macroevolutionary processes that typically involve gradual changes over longer timescales. Two notable examples of cellular populations undergoing microevolutionary processes relevant to human health are (i) populations of tumor cells undergoing clonal evolution and (ii) B cells during the adaptive immune response. Single-cell sequencing technologies have transformed our ability to analyze these cellular populations at a granular level, enabling the sequencing of individual cells within a population. This high-resolution data provides a rich source of information for phylogenetic inference, allowing for the direct but noisy observation of genetic, epigenetic, proteomic, or transcriptomic profiles.

This thesis develops novel algorithms that (i) account for the intricacies of the microevolutionary processes of interest and (ii) account for sequencing errors that may distort the signal from the noise. The algorithms proposed in this thesis are specifically designed to improve the accuracy of phylogenetic reconstruction in order to facilitate a deeper understanding of intra-tumor heterogeneity and B cell evolution. By integrating biological constraints with the intricacies of sequencing technologies, these algorithms aim to provide more accurate reconstructions of evolutionary histories, which are crucial for understanding cancer progression and immune response.

One key contribution is the development of a method for optimizing single-cell DNA sequencing experimental design with the goal of accurate downstream phylogeny inference. This thesis presents a strategy to use a less expensive, prior bulk DNA sequencing sample to recommend the minimum number of single cells to sequence needed to resolve phylogenetic uncertainty. Another contribution is the proposal of two novel algorithms for identifying and correcting sequencing errors, such as doublets and allelic dropout, which are prevalent in single-cell DNA sequencing data.

This thesis also proposes two new phylogenetic inference methods for reconstructing the evolution of a tumor from the latest generation of high-throughput but ultra-low coverage single-cell DNA sequencing technologies. The first algorithm scales well with the size of

the input and requires minimal preprocessing steps but focuses solely on reconstructing the evolutionary history of point mutations. The second algorithm more directly utilizes the strengths of these technologies, i.e., the uniformity of sequencing coverage, to infer a comprehensive tumor phylogeny that captures both point mutations and large amplifications and deletions of genomic regions.

Finally, this thesis proposes a new method of B cell phylogenetic inference for single-cell RNA sequencing data. Single-cell RNA sequencing enables the complete characterization of the microevolutionary processes of B cells during an adaptive immune response, yielding features for somatic hypermutation (SHM) and class switch recombination (CSR). Both of these microevolutionary processes play a critical role in yielding an effective short and long-term immune response. To date, existing phylogenetic approaches to reconstruct B cell evolution have primarily focused only on reconstructing the SHM process. Here, we develop an algorithm that optimally reconstructs the evolutionary history of B cell clonal lineages undergoing both SHM and CSR from single-cell RNA sequencing data. This proposed algorithm holds the potential to further advance our understanding of vaccine responses, disease progression, and the identification of therapeutic antibodies.

In summary, this thesis advances the field of molecular phylogenetics by developing specialized algorithms for studying microevolutionary processes in cellular populations. By leveraging the capabilities of single-cell sequencing technologies, these algorithms provide a deeper understanding of cancer progression and immune responses, ultimately advancing our biological knowledge of these highly complex systems and facilitating the design of more effective cancer treatments and immunization strategies.

*To my brothers and sisters who died serving in Iraq and Afghanistan.
You will never be forgotten.*

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude to my thesis advisor, Professor Mohammed El-Kebir, for his invaluable guidance, unwavering support, and insightful advice throughout this journey. I feel extremely fortunate to have begun my scientific career with such a remarkable role model. Your unending passion and joy for scientific research have been an immense source of inspiration and will stay with me in the next phases of my career. Thank you for believing in me and for always motivating me to reach my potential.

I would also like to extend my heartfelt thanks to the members of my doctoral committee, including Professor Tandy Warnow, Professor Olga Milenkovic, and Professor Nicholas Navin, for their helpful feedback, suggestions, and guidance. I especially thank Professor Tandy Warnow for her mentorship and encouragement throughout the entirety of the program. Additionally, I am exceedingly grateful to my many research collaborators. Thank you to Professor Aly A. Khan for giving me an opportunity to pursue my passion for computational immunology and for teaching me so much about the field. Thank you also to Professor Idoia Ochoa for being so generous with your time and providing positive support during some of the most challenging periods during the course of my research. I feel very fortunate to have had the opportunity to work with and learn from both of you.

Additionally, I would like to acknowledge my co-authors Nuraini Aguse, Palash Sashittal, Chuanyi Zhang, Yuanyuan Qi, and Mrinmoy S. Roddur. From each one of you, I have learned so much and grew even more into the researcher I hope to become. Not only were you generous with your technical knowledge and your time, but you made the more tedious parts of the work fun and made these last five years memorable.

I wish to express my gratitude to the Siebel School of Computing and Data Science for taking a chance on someone without a formal background in computer science. I am so grateful that you saw something in me and provided me with the resources and opportunities to pursue my research interests.

I would also like to thank the fellowship providers who have generously supported my academic journey. In particular, the Debra and Ira Cohen Fellowship played an instrumental role in helping me complete a large portion of this research. The SURGE Fellowship was critical in providing me the resources and time to kickstart my research career. Furthermore, I would like to acknowledge the National Science Foundation funds from Professor Mohammed El-Kebir, as well as the computational support from Amazon Web Services provided through the Greg Gulick Honorary Research Award. This research could not have

been completed without their support.

To my family and friends, I cannot thank you enough for your love and support over the years. Thank you especially to the members of the fraternity $\rho\alpha\tau$, past and present, whose daily emotional support and companionship have meant everything to me. I am also exceedingly grateful to RHAP for the (para)social support of the entire community. The insanity has truly helped to keep me sane. I wish to express my deepest gratitude to RLC for her kindness, support, and encouragement, which were instrumental in helping me embark on a research career. This research would not have been possible without her.

To my brother Brian, and to my sister Cara, thank you for your constant support, love, and understanding. To my parents, thank you for always believing in me and empowering me to make my own decisions and stand behind them with confidence. Thus far in life, my journey has been unconventional, but you have always stood by me and supported me every step of the way.

Finally, I wish to offer my deepest appreciation to the many patients who have so generously participated in the studies that generated the data for this research. Their invaluable contributions and willingness to devote their time have been critical in expanding our knowledge in this area. It is a great privilege to be a steward of their data with the hopes of advancing cancer and immunological research in their honor.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	Cancer Phylogenetics for Single-Cell DNA Sequencing Data	2
1.2	B Cell Phylogenetics for Single-Cell RNA Sequencing Data	8
CHAPTER 2	PHYDOSE: DESIGN OF FOLLOW-UP SINGLE-CELL SEQUENC- ING EXPERIMENTS OF TUMORS	11
2.1	Introduction	11
2.2	Materials and Methods	13
2.3	Results	23
2.4	Discussion	33
CHAPTER 3	DOUBLETD: DETECTING DOUBLET IN SINGLE-CELL DNA SEQUENCING DATA	36
3.1	Introduction	36
3.2	Materials and Methods	39
3.3	Results	44
3.4	Discussion	52
CHAPTER 4	PHYOLIN: DISTINGUISHING LINEAR AND BRANCHED EVO- LUTION GIVEN SINGLE-CELL DNA SEQUENCING DATA OF TUMORS . .	55
4.1	Introduction	55
4.2	Materials and Methods	57
4.3	Results	63
4.4	Discussion	71
CHAPTER 5	PERTILIZER: GROWING A CLONAL TREE FROM ULTRA- LOW COVERAGE SINGLE-CELL DNA SEQUENCING OF TUMORS	73
5.1	Introduction	73
5.2	Materials and Methods	75
5.3	Results	79
5.4	Discussion	88
CHAPTER 6	PHARMING: INFERRING A JOINT COPY NUMBER AND SNV CLONAL TREE FROM ULTRA-LOW COVERAGE SINGLE-CELL DNA SE- QUENCING OF TUMORS	90
6.1	Introduction	90
6.2	Materials and Methods	92
6.3	Results	100
6.4	Discussion	106

CHAPTER 7	TRIBAL: CLONAL TREE INFERENCE OF B CELL CLONAL	
	LINEAGES	107
7.1	Introduction	107
7.2	Materials and Methods	109
7.3	Results	116
7.4	Discussion	123
CHAPTER 8	CONCLUSIONS	124
APPENDIX A	SUPPLEMENTARY METHODS AND RESULTS FOR PHYDOSE	126
A.1	Supplementary Text	126
A.2	Supplementary Methods	129
A.3	Supplementary Results	137
APPENDIX B	SUPPLEMENTARY METHODS AND RESULTS FOR DOUBLET D	147
B.1	Supplementary Methods	147
B.2	Supplementary Results	149
APPENDIX C	SUPPLEMENTARY METHODS AND RESULTS FOR PHER-	
	TILIZER	161
C.1	Supplementary Methods	161
C.2	Supplementary Results	179
APPENDIX D	SUPPLEMENTARY METHODS AND RESULTS FOR PHARMING	200
D.1	Supplementary Methods	200
D.2	Supplementary Results	203
APPENDIX E	SUPPLEMENTARY METHODS AND RESULTS FOR TRIBAL . .	208
E.1	Evolutionary Model for Class Switch Recombination	208
E.2	Combinatorial Characterization and Complexity Results	210
E.3	Supplementary Methods	212
E.4	Simulation Details	218
E.5	Supplementary Results	223
REFERENCES		228

CHAPTER 1: INTRODUCTION

Molecular phylogenetics is a field that lies at the intersection of biology and computer science. It makes use of molecular data, such as DNA, RNA, or protein sequences, to infer the evolutionary relationships among various entities [1, 2, 3]. Entities include species, genes, or populations, which may range from groups of species to sets of cells/microorganisms within an individual. As these entities undergo evolutionary processes, genetic modifications accumulate over time. By sequencing samples taken at single or multiple time points, researchers construct phylogenetic trees that illustrate the shared evolutionary history of these entities. On a macro level, molecular phylogenetics has revolutionized our understanding of the tree of life, providing insights into the origins and diversification of life on Earth, the evolutionary history of specific genes, and the mechanisms of evolutionary change. These advancements have been made possible by the development of efficient and effective algorithms that tackle the complex problem of reconstructing our biological past [2, 4, 5].

Microevolutionary processes are evolutionary processes that produce genetic modifications that are observable on a human timescale. The entities of interest in these processes are typically cells, microorganism or viruses. For instance, molecular phylogenetics enabled researchers to track the microevolution of the SARS-CoV-2 virus in near real-time during the pandemic [6, 7, 8]. These insights were crucial for the development of effective vaccines and treatments, as well as for the strategic allocation of public health resources [9].

While species phylogenetic algorithms may sometimes yield sufficient results for analyzing microevolutionary processes, it is critical to distinguish the differences between micro and macroevolutionary processes. First, microevolutionary processes exhibit genetic mutation rates that are orders of magnitude higher than those in macro ones. Second, they involve dynamic populations that undergo rapid changes over short time scales, whereas macroevolutionary processes assume relatively stable populations with gradual evolutionary changes over longer periods. Third, macroevolutionary processes primarily model evolutionary changes driven by genetic drift with occasional selection, whereas microevolutionary processes explicitly account for strong selective pressures driving rapid genetic adaptation. Lastly, microevolutionary processes often deal with smaller population sizes and have more detailed and frequent sampling data compared to macro processes. Therefore, the accurate inference of microevolutionary processes is much more closely intertwined with experimental design and the minutiae of sequencing technology.

As a direct consequence of these differences, phylogenetic algorithms for microevolutionary processes must (i) account for the intricacies of the ongoing dynamic evolutionary pro-

cess of interest and (ii) account for any sequencing errors that may distort the signal from the noise. An exciting technological advancement of the 21st century has been single-cell sequencing, which permits the isolation and sequencing of individual cells within a population [10, 11, 12, 13, 14, 15, 16, 17]. Through single-cell sequencing, noisy genetic, epigenetic, proteomic, or transcriptomic cellular profiles can be directly observed. This facilitates the quantification of population heterogeneity and provides a high-resolution data source for conducting phylogenetic inference.

Two notable examples of cellular populations that are highly relevant to human health and well-suited for single-cell sequencing are (i) tumor cells undergoing clonal evolution and (ii) B cells undergoing affinity maturation during the adaptive immune response. This thesis focuses on developing algorithms specifically designed for microevolutionary processes affecting these cellular populations, leveraging the high-resolution data provided by single-cell sequencing.

1.1 CANCER PHYLOGENETICS FOR SINGLE-CELL DNA SEQUENCING DATA

According to the clonal theory of cancer [18], cancer is a microevolutionary process. This evolutionary process is driven by the stepwise acquisitions of somatic mutations. A somatic mutation is mutation that is not present in the germline but is passed on via cell division. These somatic mutations give tumor cellular populations, also known as clones, a selective advantage and yields subsequent clonal expansions and further genetic diversification. The end result of this process is a heterogeneous tumor comprised of distinct genetic populations of cells or clones. Higher levels of intra-tumor heterogeneity, or a more diverse set of tumor clones, is associated with poor prognosis, treatment resistance and higher rates of relapse [19, 20, 21, 22]. Thus, it is critical to be able to describe and quantify intra-tumor heterogeneity from genetic sequencing data of a tumor. One especially informative way to achieve this is to reconstruct this evolutionary process from sequencing data, thereby revealing the ordering of somatic mutation acquisition and the diversification that gave rise to the heterogeneous extant tumor clones. This reconstructed evolutionary history can be modeled by a tumor phylogeny or clonal tree, where the nodes of the tree represent clones and the directed incoming edges are labeled by the set of somatic mutations newly acquired to that clone (node). These tumor phylogenies are useful in practice because (i) they increase our basic understanding of cancer drivers and disease progression [23, 24, 25, 26], (ii) they facilitate comparison of evolutionary trajectories and patterns across different patient subgroups [27, 28, 29, 30, 31] and (iii) knowledge of the subclonal architecture of a tumor may be useful in guiding treatment decision-making [32, 33].

The growing quantity of sequencing data has paved the way for the field of cancer phylogenetics, whose goal is to develop algorithms and methods to interrogate tumor sequencing data from an evolutionary perspective. Initially, the majority of methods were designed for bulk DNA sequencing data due to its wide availability and decreasing costs [34, 35, 36]. While these algorithms have yielded many insights and increased our collective understanding of tumor evolution, they also suffer from a critical limitation. In particular, bulk DNA sequencing results in a sample that contains a mixture of cells from a heterogeneous tumor comprised of distinct clones. These bulk samples require deconvolution of these mixtures, making it impossible to uniquely identify a tumor phylogeny from a single or even multiple bulk DNA samples [37].

However, innovations in sequencing technology now permit the DNA sequencing of an individual tumor cell [10, 11, 12, 13]. This single-cell sequencing resolution overcomes the mixture problem associated with bulk DNA sequencing as the genotypes of individual cells are now comparable. But on the other hand, single-cell sequencing also introduces new technical errors and challenges associated with sequencing individual cells each containing microscopic amounts of DNA (≈ 6 picograms), at high enough throughput to yield accurate phylogenetic inference. There now exists a diverse array of single-cell DNA technologies (SCDNA-SEQ), each with its own distinct advantages and disadvantages.

The process of SCDNA-SEQ involves a number of sequential steps, such as cell isolation, library preparation, barcoding and the sequencing of reads. However, the particular details of how these steps are accomplished varies by assay. For example, microfluidic droplets or nanowell arrays may be used to isolate individual cells. During this isolation process, more than one cell may be captured leading to technical errors referred to as doublets. In particular, library preparation is a critical step in SCDNA-SEQ and is highly dependent on the downstream goal of the sequencing study. Techniques, such as Mission Bio Tapestry [11], multiple-displacement amplification (MDA), multiple annealing- and looping-based amplification cycles (MALBAC) and degenerate oligonucleotide-primed PCR (DOP-PCR) use an amplification stage to increase the amount of DNA prior to sequencing. However, this may result in biased coverage and/or technical errors, such as allelic dropout. In contrast, direct library preparation (DLP+) [24, 25] and acoustic cell tagmentation (ACT) [23] use direct tagmentation to simultaneously *tag* sequences with adaptors and *fragment* the DNA without amplification. This reduces the sequencing bias associated with amplification based methods but results in much lower sequencing coverage. The throughput, or the number of cells sequenced in an experiment, also varies by assay. MDA, MALBAC, and DOP-PCR were among the first generation SCDNA-SEQ technologies and have much lower throughput compared with Mission Bio Tapestry, DLP+ and ACT, which facilitate high throughput

single-cell sequencing.

Accounting for the advantages and disadvantages of each of these assays is critical when designing a SCDNA-SEQ experimental study and conducting downstream analyses. For example, to study the heterogeneity of copy number variation in a tumor, SCDNA-SEQ technologies that produce more uniform but lower coverage, such as DOP-PCR, DLP+ and ACT, are ideally suited for this task [23, 24]. In contrast, to identify single-nucleotide variants (SNV) and reconstruct their evolutionary history, targeted sequencing and other whole-genome amplification based methods have been historically preferred due to the increased sequencing depth.

Since cancer phylogenetic algorithms from bulk DNA sequencing are not optimized for single-cell DNA sequencing data, new inference algorithms have been developed that maximize the signal of single-cell DNA sequencing while accounting for the various technical limitations [38, 39, 40, 41, 42, 43, 44, 45, 46]. However, there still exists a need for algorithms that guide single-cell sequencing experiment design (Sec. 1.1.1), identify, remove or account for the presence of technical sequencing errors that may distort downstream analysis (Sec. 1.1.2) and new tumor phylogeny inference algorithms that keep pace with rapidly evolving single-cell sequencing technologies (Sec. 1.1.3).

1.1.1 Challenges in experimental design for cancer phylogenetics

A fundamental question in experimental design for SCDNA-SEQ is how many cells to sequence. One goal of SCDNA-SEQ studies is to reconstruct the evolutionary history of tumor with many population of cells, or clones, with distinct genotypes. But, a major challenge is that the number of clones and their corresponding abundances within the tumor is not known *a priori*. To increase reconstruction accuracy, it is critical to sequence a large enough number of cells to sufficiently sample all tumor clones. But on the other hand, increasing the number of sampled cells increases the cost of the experiment, especially when lower throughput SCDNA-SEQ methods are utilized. In addition to uncertainty regarding the level of heterogeneity of the tumor, additional uncertainty is introduced via sequencing errors (Sec. 1.1.2). In particular, allelic dropout (ADO) where one allele is not amplified resulting in no corresponding sequencing reads for that allele, may lead to clonal genotyping inaccuracies. Therefore, when determining the number of cells to sequence, both the goal of the downstream analysis and the error profile of the selected SCDNA-SEQ technology must factor into the experimental design.

Although there are a number of approaches for experimental design of scRNA-seq studies [47, 48, 49], there exists little guidance for the experimental design of SCDNA-SEQ studies

with downstream goal of tumor phylogeny inference. The one such existing method is SCOPIT [50], which utilizes the multinomial distribution to conduct a power calculation when given the frequency of the smallest cell population in the sample. However, it is impossible to discern *a priori* the number of clones and their corresponding frequencies. One way to overcome this challenge is to leverage information from a prior, less expensive bulk DNA sequencing experiment. Although it is not possible to uniquely infer a tumor phylogeny from bulk DNA sequencing data [37], considering the solution space of candidate phylogenies provides valuable information that can be exploited to design of a cost-effective follow up SCDNA-SEQ experiment. In particular, the unique distinguishing features of each candidate tumor phylogeny along with estimations of clonal abundances of these features can be derived and used to guide a power calculation that determines the number of cells to sequence in a follow-up SCDNA-SEQ experiment. More details are provided in Chapter 2.

1.1.2 Challenges in identifying and accounting for technical errors in SCDNA-SEQ

Although SCDNA-SEQ has proven to be a powerful tool for studying intra-tumor heterogeneity, there are a number of sequencing errors which are unique to SCDNA-SEQ when compared to bulk DNA sequencing. One such error is doublets. Doublets occur when two or more cells are isolated together but are given the same barcode, or unique identifier, thus making all subsequent sequencing reads appear as if they came from the same cell. Although cell isolation techniques aim to control this error, the rate of doublets remains a challenge for high throughput technology, such as Mission Bio Tapestry. There are three categories of doublets: (i) selflets, (ii) nested and (iii) neotypic. A *selflet* consists of two cells with identical genotypes. A *nested* doublet is where set of mutations present in one cell is a proper subset of the set of mutations in the other. Finally, a *neotypic* doublet is a doublet that is not a selflet or neotypic and implies the existence of a novel genotype not present in the tumor. Doublets are problematic for downstream analyses of intra-tumor heterogeneity and cancer phylogenetics because they may result in clonal genotyping errors, estimating clonal abundances or falsely imply stepwise evolution (linear) under divergent evolution. Thus, identifying and removed doublets is an important preprocessing step for downstream evolutionary analysis from SCDNA-SEQ data.

Since single-cell RNA sequencing (scRNA-seq) requires a similar cell isolation step, doublets are also pose problems for downstream transcriptomic analyses and gene expression studies. In contrast to SCDNA-SEQ, the problem of doublet detection in scRNA-seq has been well studied [51, 52]. Typical approaches, such as Scrublet [51], involve the simulation of artificial doublets, constructed from mixing the gene expression profiles of sequenced cells,

and training a machine learning model to predict doublets. However, one drawback of this approach is that it is only able to identify neotypic doublets. Furthermore, applying doublet detection methods designed for scRNA-seq data to SCDNA-SEQ would not account for the differences in error profiles between the two technologies. Doublet detection for SCDNA-SEQ has been historically integrated into downstream inference methods as is the case with SCG [40], SiCloneFit [43] and ∞ SCITE [16]. All of these methods introduce a more complex version of their original model to account for the presence of doublets. However, the increased model complexity results in longer running times and potentially reduced inference accuracy. To address the need for standalone methods for doublet detection, we proposed a probabilistic model to classify cell barcodes as either a single cell (singlet) or doublet. Unlike Scrublet, our approach is able to detect nested, selflet and neotypic doublets. The key insight behind this method is that set of possible latent variant allele profiles, i.e., the ratio of mutated to total alleles at a given locus, for doublets will be distinct and distinguishable from singlets. Additional details are provided in Chapter 2.

As mentioned above, another common technical challenge for amplification-based SCDNA-SEQ methods is allelic dropout (ADO). ADO is especially problematic when a mutated allele drops out, resulting in false negative genotyping at the locus for an individual cell. Although phylogeny inference methods exist that account for ADO, such as SCITE [38], it still increases the difficulty associated with tumor phylogeny inference from SCDNA-SEQ data. Moreover, when comparing evolutionary patterns within and across patient groups, researchers may not require a detailed tumor phylogeny for each patient. Instead, they may simply be interested in determining the mode of evolution i.e., linear, branched, punctuated or neutral [53]. We propose a variation of the phylogeny inference problem, that seeks to categorize the evolutionary pattern as either linear or branched from a set of cells sequenced with SCDNA-SEQ. The underlying idea is to compute the lowest ADO rate such that a linear phylogeny can be found for the observed SCDNA-SEQ data. This inferred ADO rate can be compared against the estimated or known ADO rate for the SCDNA-SEQ technology to assess the plausibility of a linear evolution. Chapter 4 contains further details on this work.

1.1.3 Challenges in tumor phylogeny inference

Somatic variants occur at many different genomic scales. The largest are whole-genome duplications, where the entire genome of the cell is duplicated. The smallest are single-nucleotide variants (SNVs), which results in the substitution of a single DNA nucleotide. In between are copy number aberrations (CNAs), which occur when segments of the genome

are amplified or deleted. A comprehensive tumor phylogeny should ideally capture simultaneously model the entire somatic evolutionary history of a tumor, i.e., it captures are types of somatic variants. However, there are inherent tradeoffs associated with SCDNA-SEQ technologies that makes this challenging.

One tradeoff in SCDNA-SEQ is uniformity versus depth of coverage. Targeting sequencing and whole-genome amplification based methods, such as Mission Bio Tapestry and MDA, results in high sequencing coverage ($\gg 1\times$) and is ideally suited for SNV analyses due to a large number of mapped reads at each locus of interest. However, these technologies suffer from biases introduced during amplification, which results in non-uniform coverage of the genome. These biases make it difficult to accurately perform copy number calling, i.e., determine the number of copies at each segment in the genome. In contrast, increased uniformity in sequencing coverage as is achieved by DOP-PCR, DLP+ and ACT, favors CNA analysis because the number of reads in each genomic region is directly proportional to its copy number. But on the other hand, they yield extremely sparse sequencing coverage, sometimes as low as $0.01\times$, making it very challenging to perform SNV analysis from this data. Furthermore, it is not possible to sequence the same set of cells using different techniques as the cells are destroyed during sequencing. Another current challenge of tumor phylogeny inference is the need for algorithms that scale to the number of cells and SNVs that are obtained from current and emerging SCDNA-SEQ technologies. In particular, a triple negative breast cancer tumor sequenced with ACT yield thousands of cells and tens of thousands of SNVs [23]. SNV tumor phylogeny inference methods designed for first generation SCDNA-SEQ, such as SCITE [38], SiCloneFit [43], PhISCS [39], SPhyR [42] rely on Markov chain Monte Carlo (MCMC) or combinatorial optimization and do not scale to the size of existing datasets. Thus, there is a need for new algorithms designed to handle thousands of cells and SNVs produced by ultra-low coverage SCDNA-SEQ technologies ($\ll 1\times$) and provide a more comprehensive view of the evolution of a tumor.

A first step towards performing both SNV and CNA analysis from the same set of cells is to leverage the strengths of a sequencing technology to improve analysis for the somatic variant type where the technology is disadvantaged. To this end, we design a heuristic algorithm for inferring an SNV tumor phylogeny from high throughput ultra-low coverage sequencing data. This algorithm explicitly leverages the large number of cells and SNVs within the data along with easily obtainable, but compressed copy number signal to overcome the sparsity of the sequencing coverage at individual loci within the genome. Chapter 6 provides more details for this algorithm.

But while this approach is able to scale efficiently to large datasets and provides an accurate reconstruction of the evolutionary history of SNVs, compressing the copy number

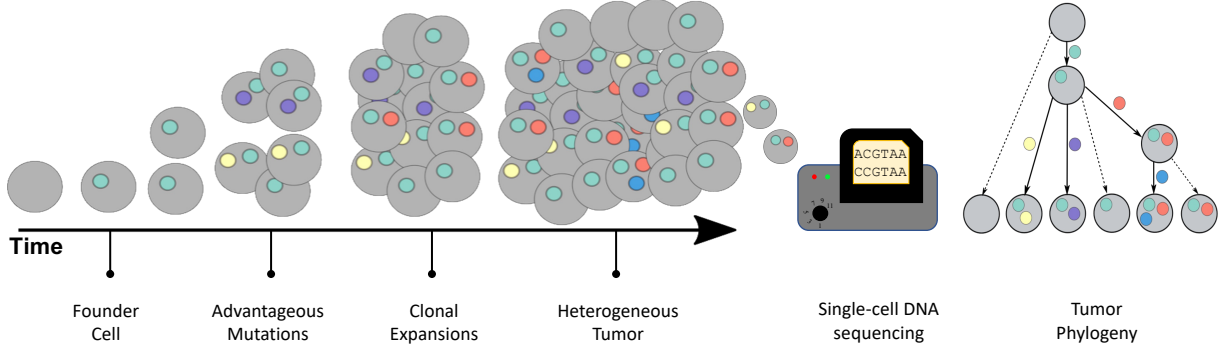


Figure 1.1: **Illustrative example of the clonal theory of cancer [18].** Depicted is the evolutionary process of cancer from a single founder cell to a heterogeneous tumor. Single-cell DNA sequencing is used to interrogate this process and the resulting data is used to reconstruct this process in the form of a tumor phylogeny. The nodes represent the tumor clones, with the root node being the normal clone. Acquired somatic mutations between clones are depicted on the edges.

signal does not permit joint inference of a CNA and SNV tumor phylogeny. Critically, SNVs may co-occur in regions affected by CNAs and a joint tumor phylogeny is key to illuminating dependencies of these somatic variants during tumor evolution. Therefore, we improve further upon the previously described method by developing a new divide and conquer algorithm to jointly infer a joint SNV and CNA tumor phylogeny from ultra-low coverage SCDNA-SEQ. This algorithm proceeds by first inferring a set of candidate tumor phylogenies for each specified segment, or genomic region, then leverages the cellular profiles to find an integration of these inferred ‘segment’ trees that best fits the observed sequencing data. We discuss this algorithm in Chapter 6.

1.2 B CELL PHYLOGENETICS FOR SINGLE-CELL RNA SEQUENCING DATA

B cells play a pivotal role in the adaptive immune response, producing antibodies that neutralize foreign substances and infections [54, 55]. These antibodies, consisting of a heavy chain and a light chain, are initially formed as sequence-specific B cell receptors (BCRs) (Fig. 1.2a). To enhance their effectiveness during immune response, B cells undergo two microevolutionary processes. The first is called affinity maturation [56], a micro-evolutionary process involving repeated cycles of *somatic hypermutation* (SHM) and cellular divisions.

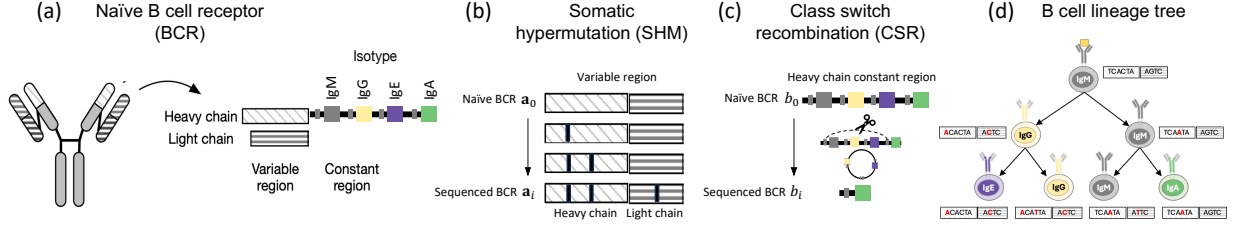


Figure 1.2: **B cells undergo microevolutionary processes during adaptive immune response.** (a) A BCR consists of paired heavy and light immunoglobulin chains, each consisting of a variable and constant region. The isotype is the heavy chain constant locus that is transcribed. (b) The BCR of B cells undergo somatic hypermutation/affinity maturation, where point mutations are introduced into the variable region of the heavy and light chains. (c) B cells also undergo class switch recombination, where the heavy chain constant locus undergoes recombination and begins transcribing a different isotype. (d) These two evolutionary processes can be modeled with a B cell lineage tree that captures the evolutionary relationships between B cells as well as the sequences and isotypes of ancestral B cells.

SHM introduces mutations in the BCR genes, selecting for B cells expressing high-affinity BCRs, while eliminating those with low affinity (Fig. 1.2b). The second is called *class switch recombination* (CSR) [57], which diversifies their response by altering the antibody’s functional class or *isotype*. When a B cell undergoes class switching from its current isotype to a new isotype, any heavy chain constant region locus between the current isotype and the new isotype in the genome is cut out or removed via a recombination process (Fig. 1.2c). Consequently, CSR is an irreversible process and the isotype of a B cell offers a distinct milestone in its evolutionary history. The evolution of a set of B cells that all descend from the same naive BCR can be modeled by a phylogeny referred to as a B cell lineage tree (Fig. 1.2d). Reconstructing the evolutionary history of B cells during these adaptive processes is integral to better understanding B cell response to infection and vaccination.

The selection pressures applied to B cells during the affinity maturation process necessitate more specialized analytical approaches than those utilized for species phylogeny inference [5, 58, 59, 60]. Specifically, Hoehn et al. developed HLP17 [61] and HLP19 [62], which are specialized codon substitution models for use with maximum likelihood inference via IgPhyML. While maximum likelihood methods, such as IgPhyML, provide a single point estimate of the lineage tree that maximizes the likelihood of the observed BCR sequences, they do not provide a complete picture of the uncertainty surrounding this estimate. Bootstrapping or the application of computationally intensive Bayesian methods is therefore needed to properly assess the extent of phylogenetic uncertainty.

Another important characteristic of B cells is the relatively short length of the BCR sequence (≈ 600 bp) and the small number of mutations introduced within a single B cell

lineage, i.e., root to tip path of a B cell lineage tree. These properties imply that maximum parsimony inference methods are viable [63, 64, 65] but typically result in a large solution space of many plausible phylogenies for the same data. In addition, these solutions exhibit additional tree uncertainty in the form of polytomies, or nodes with outdegree greater than two. This is counter to the underlying cell lineage tree, which is bifurcating as cell division results in exactly two daughter cells. Consequently, the inference of a B cell lineage tree from the BCR heavy and/or light chain sequences alone yields a high degree of phylogenetic uncertainty.

Single-cell RNA sequencing (scRNA-seq) technologies, such as 5' 10x Genomics platform and Parse Biosciences Evercode BCR, have emerged as a powerful tool for understanding and modeling cellular evolution and cell fates [14, 15, 16, 17]. These technologies offer a precise method to trace cell lineage through the observation of genomic and somatic changes, allowing for an in-depth study of the role genetic variation plays in determining cell fitness across various environments, including immune response [38, 61, 63, 66, 67, 68, 69, 70]. With scRNA-seq, it is now possible to efficiently assemble BCR sequences that includes both the heavy and light chain from a population of B cells [71]. As a result, the evolution of BCRs during affinity maturation can now be tracked with scRNA-seq with higher fidelity [17].

Previously, sequence abundance has been utilized by both GCTree [63] and ClonalTree [70] on bulk RNA sequencing data to help resolve phylogenetic uncertainty. However, sampling limitations associated scRNA-seq may dilute the sequence abundance signal by yielding very few identical BCR sequences. But on the other hand, a critical advantage of scRNA-seq is that we can simultaneously measure both the BCR sequence and expressed antibody class of an individual B cell. One common approach to resolve phylogenetic uncertainty is the inclusion of additional data or constraints to the system in question such as the use of physical location for studying cancer migration and metastasis [72], or geographical location for the inference of gene flow [73]. We propose a novel B cell lineage inference method that integrates existing biological constraints of antibody class switching, as well as measurements of the antibody class of each sequenced cell to construct B cell lineage trees that accurately reflect the evolutionary trajectory of B cells in an scRNA-seq experiment. Therefore, the inclusion of isotype measurements from scRNA-seq as well as incorporating the biological constraints of CSR evolution into the problem of B cell lineage inference has the potential to help minimize phylogenetic uncertainty in terms of both reducing the size of the solution space and yielding more refined B cell lineage trees with fewer polytomies. Our method has the potential to improve understanding of vaccine responses, track disease progression, and identify therapeutic antibodies. Details of this method are given in Chapter 7.

CHAPTER 2: PHYDOSE: DESIGN OF FOLLOW-UP SINGLE-CELL SEQUENCING EXPERIMENTS OF TUMORS

2.1 INTRODUCTION

The majority of current cancer genomics data consist of pairs of matched normal and tumor samples that have undergone bulk DNA sequencing. Bulk data is composed of sequences from cells with distinct genomes. More specifically, we observe frequencies $\mathbf{f} = [f_i]$ for the set of somatic mutations in the tumor (Fig. 2.1A). Many deconvolution methods have been proposed for tumor phylogeny inference from such data [34, 35, 75, 76, 77, 78], typically inferring a set $\mathcal{T}$ of equally plausible trees (Fig. 2.1B). These approaches are unsatisfactory, as candidate trees with different topologies may alter conclusions in downstream analyses. Single-cell DNA sequencing (SCDNA-SEQ), as opposed to bulk sequencing, enables us to observe specific clones present within the tumor. These clones correspond to the leaves of the true phylogeny, allowing phylogeny inference methods to reconstruct the tree itself once we observe all clones in the tumor [38, 42, 79, 80]. However, the elevated error rates of SCDNA-SEQ, as well as its high cost [81], make it prohibitive as a standalone method for phylogeny inference. As such, hybrid methods have been recently proposed to infer high-fidelity phylogenies from combined bulk and SCDNA-SEQ data obtained from the same tumor [39, 82]. Furthermore, without a rigorous framework to determine the number of single-cells to sequence, this decision is currently guided by budget constraints or arbitrarily determined by exogenous factors, such as thresholds for a sequencing run. This could result in excessive costs by sequencing too many cells or sunk costs associated with an unsuccessful experiment when an insufficient number of cells are sequenced.

Several hybrid datasets have been obtained by performing bulk and single-cell DNA sequencing simultaneously [83, 84]. However, there is merit in first performing bulk sequencing to guide follow-up SCDNA-SEQ experiments. For instance, several studies first identified a subset of single-nucleotide variants from the bulk data to target in subsequent SCDNA-SEQ experiments, thereby reducing costs compared to conventional whole-genome SCDNA-SEQ approaches [85, 86, 87]. A recently introduced method, SCOPIT, computes how many cells

¹This chapter contains identical material with an adapted introduction previously published in “PhyDOSE: Design of follow-up single-cell sequencing experiments of tumors” [74], which is an extended version of the paper from RECOMB-CCB 2020 by the same title. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with N. Aguse and M. El-Kebir and N. Chia, MEK conceived the project; LLW, MEK and NA established theory; N. Chia provided biological and sequencing knowledge. LLW, NA performed experiments; LLW, MEK and NA contributed to the writing of this chapter.

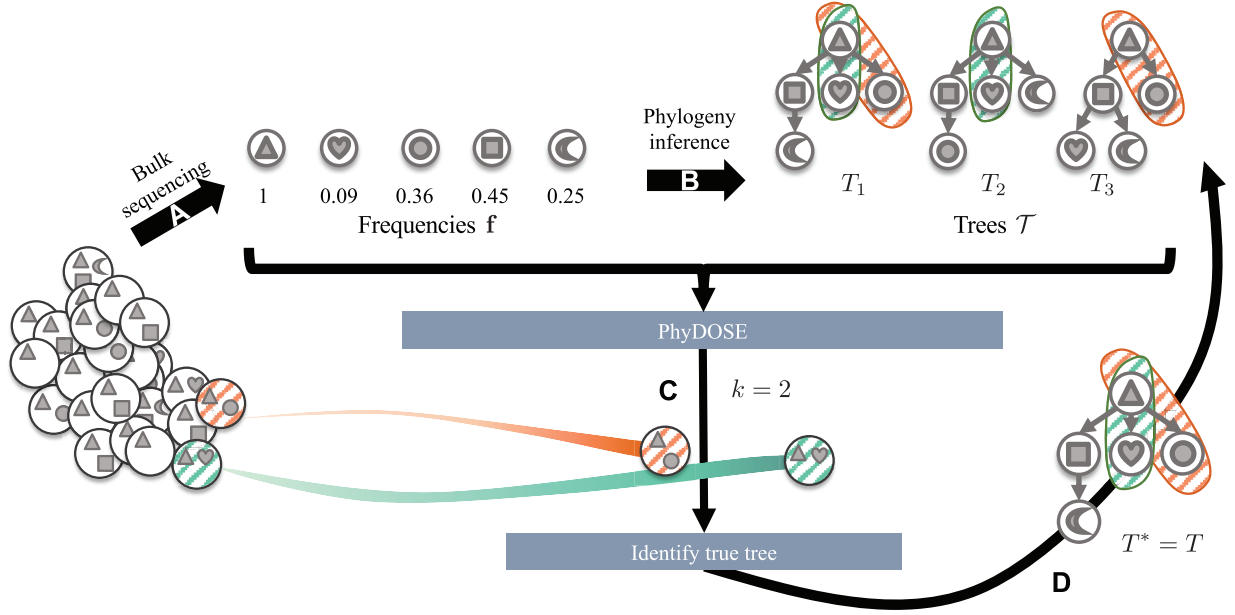


Figure 2.1: **PhyDOSE computes the number of single cells to sequence to identify the true phylogeny.** (A) Mutation frequencies $\mathbf{f}$ obtained from bulk DNA sequencing data. (B) The solution space $\mathcal{T}$ of trees inferred from $\mathbf{f}$. We show a distinguishing feature of T_1 (orange and green). (C) For tree T_1 , PhyDOSE suggests that $k = 2$ single cells suffice to observe clones that are unique to T_1 . (D) In a follow-up SCDNA-SEQ experiment we observe $k = 2$ cells, one from the orange clone and one from the green clone. As such, we eliminate trees T_2 and T_3 , concluding that phylogeny T_1 is the true phylogeny T^* .

are needed to observe all clones of a tumor, given estimates on the smallest prevalence of a clone as well as the number of clones with that smallest prevalence to detect [50]. The authors provide no guidance on how to obtain these two quantities. Here, we build upon this work by directly incorporating knowledge encoded by the trees $\mathcal{T}$ inferred from the initial bulk sequencing data. Indeed, by using data from an SCDNA-SEQ experiment we may eliminate trees from $\mathcal{T}$ that do not align with the observed clones (Fig. 2.1). In other words, if we observe all clones in a tumor, it is possible to determine the phylogeny of the tumor. However, is it possible to achieve the same goal by observing fewer clones? If so, how many cells are necessary for us to observe the required clones?

We introduce Phylogenetic Design Of Single-cell sequencing Experiments (PhyDOSE), a method to strategically design a follow-up SCDNA-SEQ experiment aimed at inferring the true phylogeny (Fig. 2.1). Given a set $\mathcal{T}$ of candidate trees inferred from initial bulk data, we describe how to distinguish a single tree T among the rest using features unique to T . In particular, if our SCDNA-SEQ experiment results in observing cells corresponding to a distinguishing feature of T , we may conclude that T is in fact the true tree. This means that

we can typically identify T using only a subset of the clones. To determine the number of cells to sequence, we introduce a probabilistic model that incorporates SCDNA-SEQ errors and models successful SCDNA-SEQ experiments as a tail probability of a multinomial distribution (Fig. 2.1D). Finally, we reconcile the sampled cells utilizing these distinguishing features to infer the true phylogeny (Fig. 2.1D) and provide heuristics for considering uncertainty in frequency estimates and determining the number of cells to sequence across multiple available biopsies. We validate PhyDOSE using both simulated data and a retrospective analysis of a leukemia patient that has undergone both bulk and SCDNA-SEQ sequencing. We also demonstrate the utility of PhyDOSE by prospectively computing how many cells are needed to resolve the uncertainty in phylogenies of a recent acute myeloid leukemia cohort [88] and lung cancer cohort [89]. The cost-efficient SCDNA-SEQ experiments enabled by PhyDOSE will yield high-fidelity phylogenies, improving downstream analyses aimed at understanding tumorigenesis and developing treatment plans.

2.2 MATERIALS AND METHODS

We introduce Phylogenetic Design Of Single-cell sequencing Experiments (PhyDOSE), a method to determine the number of single cells to sequence to identify the true phylogeny given initial bulk sequencing data. PhyDOSE is implemented in C++/R and is available as an R package at <https://github.com/elkebir-group/phydoser>. This section describes the various methodological components of PhyDOSE.

2.2.1 Problem statement

Let n be the number of single-nucleotide variants, or simply *mutations*, identified from initial bulk sequencing data of a matched normal and tumor biopsy sample. For each mutation i , we observe the *variant allele frequency* (VAF), i.e. the fraction of aligned reads that harbor the tumor allele at the locus of mutation i . Specialized methods exist that combine copy number information and VAFs to infer a *cancer cell fraction* f_i for each mutation i , which is the proportion of cells in the tumor biopsy that contain at least one copy of the mutation [89, 90, 91, 92]. Here, we refer to cancer cell fractions as *frequencies*. Typically, phylogenies $\mathcal{T}$ inferred by current methods from frequencies $\mathbf{f} = [f_i]$ adhere to the infinite sites assumption. That is, each mutation i is introduced exactly once at vertex v_i and never subsequently lost.

When we sequence a single cell from the same tumor biopsy, assuming no errors, we identify a clone of the tumor. In other words, we observe a set of mutations that must form

a connected path in the unknown true phylogeny T^* . By repeatedly sequencing single cells until we observe all clones in the tumor, we will have observed all root-to-vertex paths of T^* , thus identifying tree T^* itself. We assume that (i) the true unknown phylogeny T^* is among the trees in $\mathcal{T}$ and that (ii) mutations among single cells that we sample from the tumor biopsy follow the same distribution as $\mathbf{f}$. These assumptions are important for the mathematical derivation of PhyDOSE but it is typical for violations to occur in practice. Through simulations, we explore the impact of violating these assumptions and show that our approach is robust to many realistic scenarios.

This leads to the following question and problem statement with respect to these two assumptions. How many single cells do we need to identify T^* with confidence level γ ?

Problem 2.1 (SCS POWER CALCULATION (SCS-PC)). Given a set $\mathcal{T}$ of candidate phylogenies, frequencies $\mathbf{f}$ and confidence level γ , find the minimum number k^* of single cells needed to determine the true phylogeny T^* among $\mathcal{T}$ with probability at least γ .

Clearly, we do not know which phylogeny in $\mathcal{T}$ is the true underlying phylogeny T^* of the tumor. Thus, we consider a slightly different problem: In the T -SCS-PC problem (defined formally at the end of the section), we are given an arbitrary phylogeny $T \in \mathcal{T}$ and want to perform a similar power calculation when conditioning on T being the true phylogeny. By solving the T -SCS-PC problem for all trees $T_1, \dots, T_{|\mathcal{T}|}$, we obtain the numbers $k(T_1), \dots, k(T_{|\mathcal{T}|})$ of single cells needed for each tree. As T^* is in $\mathcal{T}$, the maximum number among $k(T_1), \dots, k(T_{|\mathcal{T}|})$ is an upper bound on the number of required SCDNA-SEQ experiments to identify T^* with probability at least γ . To solve the T -SCS-PC problem, we need to reason for which SCDNA-SEQ experiments we can conclude that T is the true phylogeny.

Observe that each tree T in $\mathcal{T}$ describes a unique set of clones, corresponding to the sets of mutations encountered in all root-to-vertex paths of T (Fig. 2.1). Thus, if we observe all clones of a phylogeny T in our SCDNA-SEQ experiments, we may conclude that T is the true phylogeny. What is the probability of doing so? To answer this question, we must compute the prevalence of each clone in the tumor biopsy.

For phylogenies that adhere to the infinite sites assumption, the *prevalence* $\mathbf{u}(T, \mathbf{f}) = [u_i]$ of the clones in the tumor biopsy are uniquely determined by the phylogeny T and frequencies $\mathbf{f}$ as

$$u_i = f_i - \sum_{j \in \delta_T(i)} f_j \quad \forall i \in [n]. \quad (2.1)$$

where $\delta_T(i)$ is the set of children of the node where mutation i was introduced [75].

Tumor phylogeny inference methods guarantee that the inferred phylogenies $\mathcal{T}$ from frequencies $\mathbf{f}$ have clonal prevalence $\mathbf{u}(T, \mathbf{f}) = [u_i]$ that are nonnegative and that $\sum_{i=1}^n u_i \leq 1$, where the remainder $u_0 = 1 - \sum_{i=1}^n u_i$ is the prevalence of the normal clone. Thus, conditioning on a phylogeny T and frequencies $\mathbf{f}$, sequencing one cell from the tumor will lead us to observe one of the $n + 1$ clones of T with probabilities $(u_0, \dots, u_n)$. In other words, the outcome of this SCDNA-SEQ experiment with one cell is a draw from the categorical distribution $\text{Cat}(u_0, \dots, u_n)$. The possible outcomes of an SCDNA-SEQ experiment composed of k cells thus follow a multinomial distribution $\text{Mult}(u_0, \dots, u_n)$. Thus, the probability of observing all tumor clones of T in such an SCDNA-SEQ experiment with k cells corresponds to the tail probability of the multinomial where each of the n tumor clones is observed at least once.

The corresponding *power calculation* is to determine the smallest number for k where the tail probability is greater or equal to the confidence level γ . Note that this power calculation for observing all clones has been previously introduced [50].

Importantly, in many cases we need not observe all clones of T to distinguish T from the remaining phylogenies $\mathcal{T} \setminus \{T\}$ (Fig. 2.2). This means that we may conclude that T is the true phylogeny with an SCDNA-SEQ experiment with fewer cells. To formalize this notion, we start by defining a featurette.

Definition 2.1. A *featurette* τ is a subset of mutations.

We say that a featurette τ is *present* in a phylogeny T if the nodes/mutations of τ form a connected path of T starting at the root node, otherwise we say that τ is *absent* in T . The same featurette, however, may be present in more than one phylogeny. Thus, multiple featurettes may be required to distinguish a phylogeny T from the remaining phylogenies $\mathcal{T} \setminus \{T\}$.

Definition 2.2. A set Π of featurettes is a *distinguishing feature* of T if (i) for all featurettes $\tau \in \Pi$ it holds that τ is present in T , and (ii) for each remaining phylogeny $T' \in \mathcal{T} \setminus \{T\}$ there exists a featurette $\tau' \in \Pi$ where τ' is absent in T' .

Thus, an SCDNA-SEQ experiment where we observe one cell from each clone of a distinguishing feature Π of T enables us to conclude that phylogeny T is the true phylogeny. As discussed, every phylogeny T has a *trivial distinguishing feature*, which is composed of all featurettes present in T . Moreover, T may have multiple distinguishing features. Therefore, we must consider the complete set of all distinguishing features, which we call the distinguishing feature family.

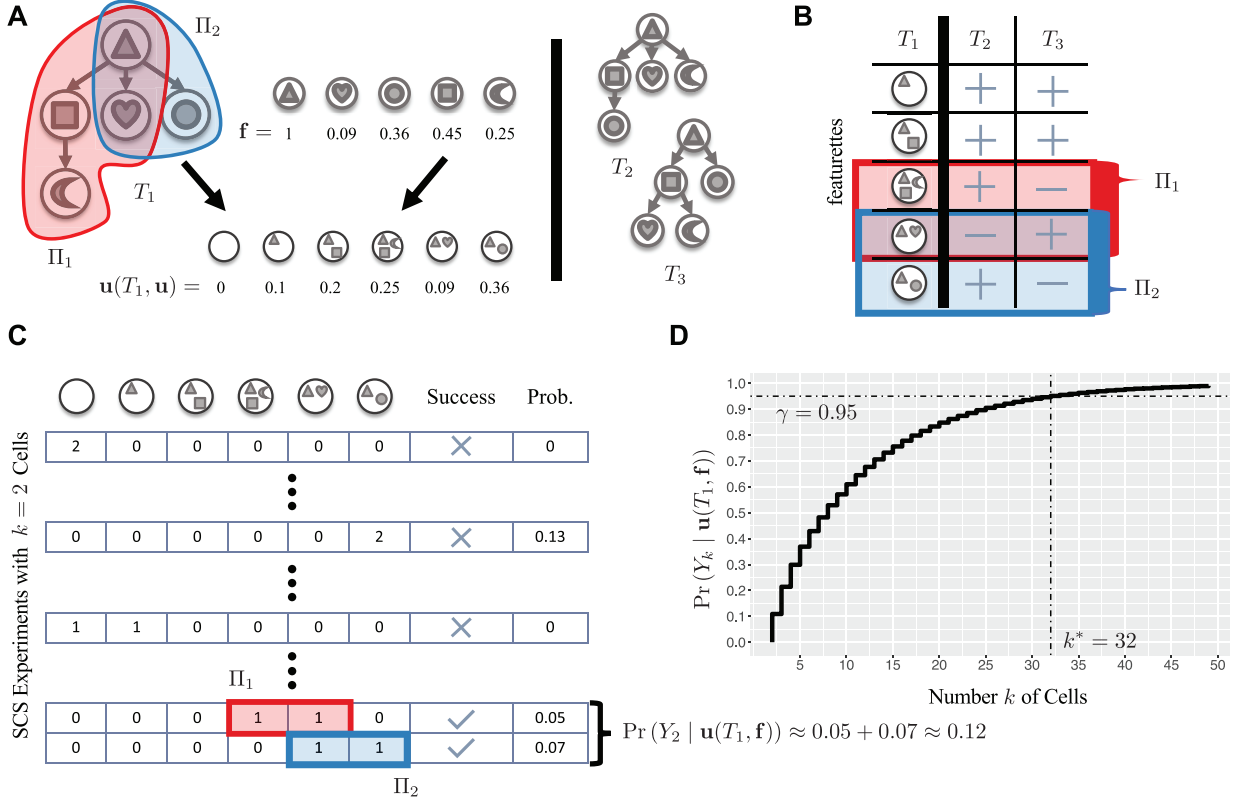


Figure 2.2: **The SCS POWER CALCULATION FOR PHYLOGENY T (T -SCS-PC) problem.** (A) We are given frequencies $\mathbf{f}$ and a tree T_1 that we want to distinguish from the other trees $\{T_2, T_3\}$. The pair $(T_1, \mathbf{f})$ uniquely determine clonal prevalence $\mathbf{u}(T_1, \mathbf{f})$. (B) Featurettes of T_1 correspond to root-to-vertex paths, yielding distinguishing features Π_1 and Π_2 , each with one featurette absent in T_2 and another absent in T_3 . (C) With $k = 2$ cells, we must observe clones from either Π_1 or Π_2 for a successful outcome, resulting in probability $\Pr(Y_2 | \mathbf{u}(T_1, \mathbf{f})) \approx 0.12$. (d) To increase this probability to $\gamma = 0.95$, we need $k^* = 32$ cells.

Definition 2.3. The set $\Phi(T, \mathcal{T} \setminus \{T\})$ composed of all distinguishing features of T with respect to $\mathcal{T} \setminus \{T\}$ is a *distinguishing feature family* of T .

Let $(c_0, \dots, c_n)$ be the outcome of an SCDNA-SEQ experiment of k cells, where $c_i \geq 0$ is the number of cells observed of clone i and $\sum_{i=0}^n c_i = k$. This experiment is *successful* if, among the k sequenced cells, we observe the clones of at least one distinguishing feature $\Pi \in \Phi(T, \mathcal{T} \setminus \{T\})$ — i.e. $c_i > 0$ for all clones i in some distinguishing feature $\Pi \in \Phi(T, \mathcal{T} \setminus \{T\})$. As discussed, conditioning on frequencies $\mathbf{f}$ and T being the true phylogeny, outcomes $(c_0, \dots, c_n)$ of SCDNA-SEQ experiments of k cells follow a multinomial distribution $\text{Mult}(k, u_0, \dots, u_n)$ where $\mathbf{u}(T, \mathbf{f}) = [u_i]$ is defined as in (2.1). Let Y_k denote the event of a successful outcome. We are interested in computing the probability $\Pr(Y_k | \mathbf{u}(T, \mathbf{f}))$, which equals the sum of the probabilities of all successful outcomes. More specifically, we want to

determine the smallest number k^* of single cells to sequence such that $\Pr(Y_{k^*} \mid \mathbf{u}(T, \mathbf{f}))$ is at least the prescribed confidence level γ (Fig. 2.2).

Problem 2.2 (SCS POWER CALCULATION FOR PHYLOGENY T (T -SCS-PC)). Given a set $\mathcal{T}$ of candidate phylogenies and a phylogeny $T \in \mathcal{T}$, frequencies $\mathbf{f}$ and confidence level γ , find the minimum number k^* of single cells needed such that $\Pr(Y_{k^*} \mid \mathbf{u}(T, \mathbf{f})) \geq \gamma$.

In Appendix A.1.1, we prove that the above problem is NP-hard.

Theorem 2.1. T -SCS-PC is NP-hard.

Multiple biopsies. The SCS-PC problem is only applicable to bulk sequencing data obtained from a single biopsy, i.e. the number of cells calculated is only for an SCDNA-SEQ experiment on one sample. However, bulk samples from tumors are often obtained from multiple biopsies, each with different mutation frequencies and consequently different clonal prevalences. One approach to support such data is to solve the SCS-PC problem for each biopsy in isolation and select the biopsy that requires the smallest number of cells. However, a more cost-effective approach that also better captures intra-tumor heterogeneity is to perform a follow-up SCDNA-SEQ experiment with cells from multiple biopsies. In particular, the naive selection approach might not yield a solution if the clones of a distinguishing feature do not co-occur in a single biopsy.

With $b \geq 1$ biopsies, the input changes from a frequency vector $\mathbf{f} \in [0, 1]^n$ to a frequency matrix $F \in [0, 1]^{b \times n}$, whose entries f_{pi} indicate the frequency of mutation i in biopsy p . Similarly, the output changes from an integer $k^* \in \mathbb{N}$ to a count vector $\mathbf{k}^* \in \mathbb{N}^b$, such that each entry k_p^* indicates the number of single cells in biopsy p . We have the following two generalizations of the SCS-PC problem and the T -SCS-PC problem for the case of $b \geq 1$ biopsies.

Problem 2.3 (MULTI-SAMPLE SCS POWER CALCULATION (MUL-SCS-PC)). Given a set $\mathcal{T}$ of candidate phylogenies, frequencies F from b biopsies and confidence level γ , find the numbers $\mathbf{k}^* = [k_1^*, \dots, k_b^*]$ of single cells needed from each biopsy to determine the true phylogeny T^* among $\mathcal{T}$ with probability at least γ and the total number $\|\mathbf{k}^*\|_1 = \sum_{p=1}^b k_p^*$ of cells is minimum.

Problem 2.4 (MULTI-SAMPLE SCS POWER CALCULATION FOR PHYLOGENY T (T -MUL-SCS-PC)). Given a set $\mathcal{T}$ of candidate phylogenies and a phylogeny $T \in \mathcal{T}$, frequencies F from b biopsies and confidence level γ , find the numbers $\mathbf{k}^* = [k_1^*, \dots, k_b^*]$ of single cells needed from each biopsy such that $\Pr(Y_{\mathbf{k}^*} \mid U(T, F)) \geq \gamma$ and the total number $\|\mathbf{k}^*\|_1 = \sum_{p=1}^b k_p^*$ of cells is minimum.

For the case where $b = 1$, the T -SCS-PC and the T -MUL-SCS-PC problems are identical, amounting to following hardness result.

Corollary 2.1. T -MUL-SCS-PC is NP-hard.

2.2.2 Multinomial Power Calculation

To solve the T -SCS-PC problem, it suffices to have an algorithm that computes $\Pr(Y_k \mid \mathbf{u}(T, \mathbf{f}))$, which is the probability of concluding that T is the true phylogeny. Using this algorithm we identify k^* by starting from $k = 0$ and simply incrementing k until the corresponding probability $\Pr(Y_k \mid \mathbf{u}(T, \mathbf{f}))$ exceeds the prescribed confidence level γ . In the following, we describe how to efficiently compute $\Pr(Y_k \mid \mathbf{u}(T, \mathbf{f}))$.

Recall that the outcome of an sCDNA-SEQ experiment composed of k cells corresponds to a vector $\mathbf{c} = [c_i]$, where $c_i \geq 0$ is the number of cells that we observe from clone i and $\sum_{i=0}^n c_i = k$. In a successful outcome $\mathbf{c}$ we observe at least one cell for each featurette in at least one distinguishing feature $\Pi \in \Phi(T, \mathcal{T} \setminus \{T\})$, where $\Phi(T, \mathcal{T} \setminus \{T\})$ is the distinguishing feature family. For brevity, we will write Φ rather than $\Phi(T, \mathcal{T} \setminus \{T\})$.

Let $\mathbf{c}(\Pi, k)$ denote the set of all outcomes where we observe at least one cell for each featurette in a distinguishing feature Π — i.e. $\sum_{i=0}^n c_i = k$, and for all $i \in \{0, \dots, n\}$ it holds that $c_i > 0$ if clone i is a featurette in Π and $c_i \geq 0$ otherwise. The set $\mathbf{c}(\Phi, k)$ of successful outcomes is defined as the union $\bigcup_{\Pi \in \Phi} \mathbf{c}(\Pi, k)$. The probability of any sCDNA-SEQ outcome $\mathbf{c} = (c_0, \dots, c_n)$ is distributed according to $\text{Mult}(k, \mathbf{u}(T, \mathbf{f}))$. Since successful outcomes enable us to conclude that T is the true phylogeny, we have

$$\Pr(Y_k \mid \mathbf{u}(T, \mathbf{f})) = \sum_{\ell \in \mathbf{c}(\Phi, k)} \text{Mult}(\ell \mid k, \mathbf{u}(T, \mathbf{f})) = \sum_{\ell \in \mathbf{c}(\Phi, k)} \frac{k!}{\prod_{i=0}^n \ell_i!} \prod_{i=0}^n u_i^{\ell_i}. \quad (2.2)$$

If there is only one distinguishing feature Π , i.e. $\Phi = \{\Pi\}$, then the desired probability is a standard tail probability of the multinomial where we sum up the probabilities of outcomes $\mathbf{c}(\Pi, k) = [c_i]$ such that $\sum_{i=0}^n c_i = k$, $c_i > 0$ if clone i is a featurette of Π and $c_i \geq 0$ otherwise. A fast calculation of this tail probability was developed using a connection to the conditional probability of independent Poisson random variables [50, 93]. If there are multiple distinguishing features but they are pairwise disjoint — i.e. no two distinct distinguishing features share the same featurette — then we simply have

$$\Pr(Y_k \mid \mathbf{u}(T, \mathbf{f})) = \sum_{\Pi \in \Phi} \sum_{\ell \in \mathbf{c}(\Pi, k)} \text{Mult}(\ell \mid k, \mathbf{u}(T, \mathbf{f})), \quad (2.3)$$

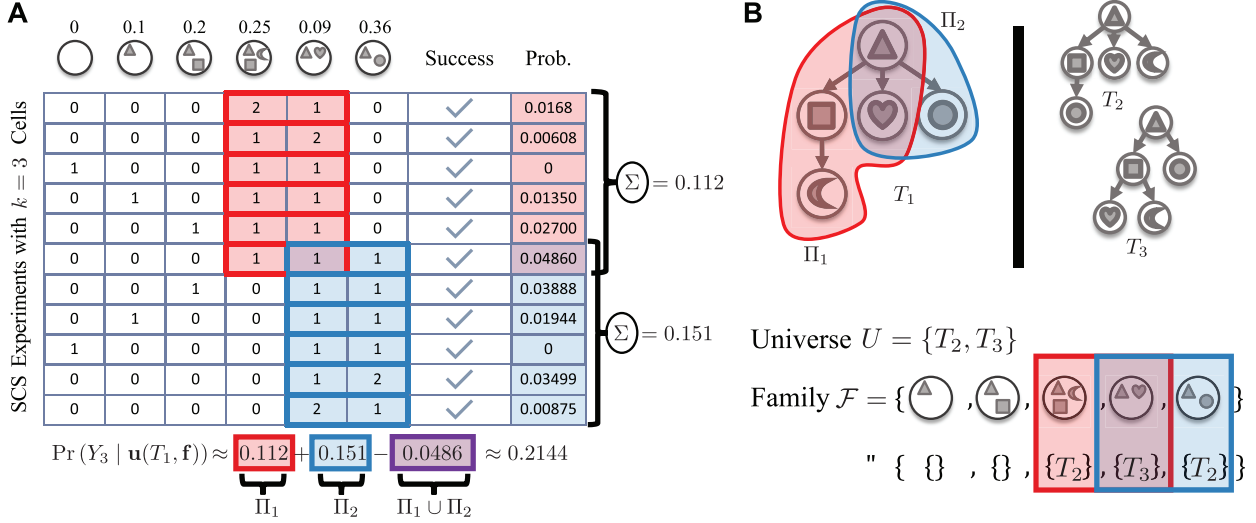


Figure 2.3: **PhyDOSE implementation details.** (A) To account for minimal distinguishing features that share featurettes, we use the inclusion-exclusion principle to compute $\Pr(Y_k | \mathbf{u}(T, \mathbf{f}))$. Here, Π_1 (red) and Π_2 (blue) share a featurette (with ‘triangle’ and ‘heart’ mutations). (B) To enumerate the set Φ^* of minimal distinguishing features of T_1 , we reduce the problem to SET COVER and repeatedly identify minimum covers. Here, the universe U is composed of trees $\{T_2, T_3\}$ and there is a subset in $\mathcal{F}$ for each featurette τ of T_1 composed of the trees where τ is absent.

and we can apply the fast computation [50] to obtain each independent tail probability. However, the equality in the above equation does not hold if the family Φ is composed of distinguishing features with overlapping featurettes. Incorrectly applying this equation will lead us to overestimate the value of k^* . Since single-cell sequencing is expensive, overestimating the number of cells to sequence in an SCDNA-SEQ experiment can be costly and unnecessary. One naive way would be to simply brute force all $(n + 1)^k$ SCDNA-SEQ outcomes, but this will not scale. Instead, to calculate $\Pr(Y_k | \mathbf{u}(T, \mathbf{f}))$ exactly, we propose to use the inclusion-exclusion principle as follows.

$$\Pr(Y_k | \mathbf{u}(T, \mathbf{f})) = \sum_{\emptyset \subsetneq \Phi' \subseteq \Phi} (-1)^{|\Phi'|+1} \sum_{\ell \in \mathbf{c}(I(\Phi'), k)} \text{Mult}(\ell | k, \mathbf{u}(T, \mathbf{f})), \quad (2.4)$$

where $I(\Phi')$ is the set of all featurettes in Φ' , i.e. $I(\Phi') = \bigcup_{\Pi \in \Phi'} \Pi$.

Thus, we need to compute $2^{|\Phi|} - 1$ tail probabilities, which each can be done using the fast calculation in SCOPIT [50].

In the worst case, Φ has $O(2^n)$ distinguishing features resulting in $O(2^n)$ tail probabilities. We now describe one final optimization that will significantly reduce the number of required computations. This is based on the following observation.

Observation 2.1. If Π is a distinguishing feature of T then for all featurettes τ present in T it holds that $\Pi \cup \{\tau\}$ is a distinguishing feature of T .

This means that distinguishing features in Φ form a partially ordered set under the set inclusion relation. We call a distinguishing feature Π *minimal* if there does not exist another distinguishing feature $\Pi' \in \Phi$ that is a proper subset of Π , i.e. $\Pi' \subsetneq \Pi$.

A direct consequence of Observation 2.1 is that the outcome of an SCDNA-SEQ experiment is successful when we observe all featurettes of a distinguishing feature Π , and remains so even if we observe additional featurettes $\tau' \notin \Pi$.

As such, successful outcomes w.r.t. Φ equal those w.r.t. the set Φ^* of all minimal distinguishing features of T .

Observation 2.2. It holds that $\mathbf{c}(\Phi^*, k) = \mathbf{c}(\Phi, k)$.

Therefore, it suffices to restrict our attention to only Φ^* rather than the complete family Φ when computing $\Pr(Y_k \mid \mathbf{u}(T, \mathbf{f}))$ using (2.4). Appendix A.2 describes how to find Φ^* by reducing the problem to that of finding all minimal set covers, which we solve in an iterative fashion using integer linear programming.

2.2.3 Power calculation for multiple biopsies

We now discuss the T -MUL-SCS-PC, which is the generalization of the T -SCS-PC problem to $b \geq 1$ biopsies. The key probability is $\Pr(Y_{\mathbf{k}} \mid U(T, F))$, i.e. the probability of concluding that T is the true tree when sequencing $\mathbf{k} = [k_1, \dots, k_b]^T$ cells from each biopsy. In the following, we discuss an exact (but computationally intensive) approach to compute this probability as well as a fast heuristic.

Given the numbers $\mathbf{k} = [k_1, \dots, k_b]^T$ of cells to sequence from each biopsy, an *outcome* of the corresponding SCDNA-SEQ experiment across b biopsies is defined as a matrix $C = [c_{pi}]$ such that $c_{pi} \in \mathbb{N}$ is the number of cells that we observe from clone i in biopsy p , and $\sum_{i=0} c_{pi} = k_p$ for all biopsies p . A *successful outcome* for a distinguishing feature Π is an outcome where we observe at least one cell for each featurette in Π . Let $C(\Pi, \mathbf{k})$ denote the set of all successful outcomes for distinguishing feature Π . The set $C(\Phi^*, \mathbf{k})$ of successful outcomes for the minimal distinguishing feature family Φ^* is defined as the union $\bigcup_{\Pi \in \Phi^*} C(\Pi, \mathbf{k})$. Since sequencing of each biopsies proceeds independently, the probability of a observing an outcome $C = [\mathbf{c}_1, \dots, \mathbf{c}_b]^T$ across b biopsies equals

$$\Pr(\mathbf{c}_1, \dots, \mathbf{c}_b \mid \mathbf{k}, U(T, F)) = \prod_{p=1}^b \Pr(\mathbf{c}_p \mid k_p, \mathbf{u}(T, \mathbf{f}_p)) = \prod_{p=1}^b \text{Mult}(\mathbf{c}_p \mid k_p, \mathbf{u}(T, \mathbf{f}_p)). \quad (2.5)$$

Hence, the desired tail probability of successful outcomes for the minimal distinguishing feature family Φ^* equals

$$\Pr(Y_{\mathbf{k}} \mid U(T, F)) = \sum_{[\mathbf{c}_1, \dots, \mathbf{c}_b] \in C(\Phi^*, \mathbf{k})} \prod_{p=1}^b \text{Mult}(\mathbf{c}_p \mid k_p, u(T, \mathbf{f}_p)). \quad (2.6)$$

For a single biopsy ($b = 1$), the probability $\Pr(Y_{\mathbf{k}} \mid U(T, F))$ corresponds to sums of multinomial tail probabilities, enabling fast calculation using [50] as discussed in the previous section. This is no longer the case for $b > 1$ biopsies where the tail probability is over b independent multinomial distributions (see product in Equation (2.6)). A naive way to compute $\Pr(Y_{\mathbf{k}} \mid U(T, F))$ would be to exhaustively enumerate all SCDNA-SEQ outcomes with $\mathbf{k}$ cells, which scales exponentially in $\mathbf{k}$. As such, we develop a heuristic approach, which selects a subset of required featurettes/clones in each biopsy that together form a distinguishing feature in Φ^* and achieve the smallest total number of cells with confidence level γ . Appendix A.2.3 provides more details and Fig. A.2 provides an example where the heuristic returns a suboptimal solution.

2.2.4 Consideration of SCDNA-SEQ error rates

One current challenge with SCDNA-SEQ is that the false negative rate per site is quite high with typical rates up to 0.4 for the commonly used multiple displacement amplification (MDA) method [94]. On the other hand, current false positive rates are low and are typically less than 0.0005 for MDA-based whole-genome amplification [94]. A *false negative* is defined as not observing a mutation that is present in the cell. A *false positive* occurs when we observe the presence of a mutation that did not occur in that cell.

With PhyDOSE, we propose one possible method for incorporating the false negative rate β when it is known. Specifically, sampled cells follow a categorical distribution $\mathbf{u} = [u_0, \dots, u_n]$ when conditioned on tree T . Hence, the probability of sampling a cell from clone i equals u_i . True positives, i.e. correctly observing a mutation in a clone, follow a Bernoulli distribution with parameter $1 - \beta$. To observe a featurette/clone i that has n_i mutations and a prevalence of u_i , we thus need to have n_i true positives. In other words, assuming independence among mutations, we require n_i successful draws from a Bernoulli distribution parameterized by $1 - \beta$. As such, we derive new clonal prevalence $\mathbf{u}'(T, \mathbf{f}, \beta) = [u'_i]$ from $\mathbf{u}(T, \mathbf{f}) = [u_i]$. Additionally assuming independence between the events of a cell being sampled from clone i and the absence of false negatives, we set $u'_i = u_i(1 - \beta)^{n_i}$ where n_i is

the number of mutations in featurette/clone i . We set u'_0 to be equal to $1 - \sum_{i=1}^n u'_i$. This adjustment results in a reduction of the clonal prevalence and ultimately increases the value of k^* . The issue of false positives is less serious as error rates are low enough to be negligible.

2.2.5 Prioritizing candidate trees post SCDNA-SEQ experiment

The final step is to prioritize candidate trees after performing an SCDNA-SEQ experiment with the number k^* of cells computed by PhyDOSE. To this end, we compute the *support* of each tree $T \in \mathcal{T}$. Intuitively, $\text{support}(T)$ is the number of cells that support the conclusion that T is the actual phylogeny. Formally, we say that a distinguishing feature Π of a tree T is *observed* if each featurette of Π is observed in at least one cell. Using this, we define $\text{support}(T)$ as the number of cells that correspond to featurettes of an observed distinguishing feature Π of T . Per Observation 2.1, it suffices to restrict our attention to the set Π^* of minimal distinguishing features.

There are two outcomes of an SCDNA-SEQ experiment with k^* cells. Either there is no tree $T \in \mathcal{T}$ with non-zero support or there are one or more trees with non-zero support. In the former case, the SCDNA-SEQ experiment has failed, which is expected to occur with probability $1 - \gamma$. In the latter case, which may occur in the presence of false negatives and false positives, we return the set of trees with maximum support.

Alternatively, we may use existing methods that infer tumor phylogeny from SCDNA-SEQ data [38, 42, 43] or a combination of SCDNA-SEQ and bulk data [39, 82].

2.2.6 k^* confidence interval

A common challenge in bulk sequencing is uncertainty in the cancer cell fractions $\mathbf{f}$ due to sampling of reads as well as sequencing and mapping errors. Following standard practice [34, 72, 75], we account for this uncertainty by taking confidence intervals $[\mathbf{f}^-, \mathbf{f}^+]$ as input. Typically, such confidence intervals are obtained by viewing variant read counts as draws from a binomial or beta-binomial distribution.

Importantly, uncertainty in the cancer cell fraction leads to uncertainty in clonal prevalences. Therefore, for each tree $T \in \mathcal{T}$, we utilize $[\mathbf{f}^-, \mathbf{f}^+]$ to construct an interval $[k_-^*(T), k_+^*(T)]$ of the number of single cells reflecting the extreme values the clonal prevalences may assume. To find these values, we must consider the frequencies $[\mathbf{f}^-, \mathbf{f}^+]$ using constraints from the tree T following the sum condition (2.1) and the featurettes in a distinguishing feature of T . We do this using a heuristic that we describe in Appendix A.2.2. We set the overall interval $[k_-^*, k_+^*]$ conservatively as the confidence interval $[k_-^*(T), k_+^*(T)]$ from the tree T with the

maximum $k_+^*(T)$ among all trees $T \in \mathcal{T}$.

2.2.7 `phydoser` R package

We developed PhyDOSE and the associated optimizations into a freely available R package named `phydoser`. The functions in the `phydoser` R package are grouped into four areas: (i) I/O support (ii) pre and post processing (iii) PhyDOSE implementation iv) visualization. For I/O support, `phydoser` offers a suite of functions to read and convert external data into the data structures required by the R implementation of PhyDOSE. The pre and post processing capabilities include generation of the distinguishing feature families of each tree, the frequency inputs for the k^* confidence interval and the computation of the support metric for a completed SCDNA-SEQ experiment. The implementation of PhyDOSE includes both a single biopsy and a multiple biopsy mode. Lastly, the visualization functions facilitate creation of high resolution tree graphics while annotating the distinguishing feature or a specific feature of a tree. `phydoser` is available at <https://github.com/elkebir-group/phydoser>.

2.3 RESULTS

In this section, we demonstrate the application of PhyDOSE to simulated and real data. We begin by validating our method using simulated data. Next, we provide retrospective results for a leukemia patient [85] and an acute myeloid leukemia cohort [88] where both bulk and single-cell DNA sequencing have been performed [85]. Finally, we use PhyDOSE to perform a prospective analysis to determine the required number of single cells to identify the true phylogeny in a non-small cell lung cancer patient cohort [89]. Source data and results can be found at <https://github.com/elkebir-group/PhyDOSE>.

2.3.1 Simulations

Design. We used simulations to assess (i) the benefit of PhyDOSE’s distinguishing feature analysis, (ii) robustness to uncertainty due to sequencing errors and (iii) robustness to violations of PhyDOSE’s model assumptions. We generated simulated data where the ground truth tree T^* is known. Given a fixed number c of clones and n mutations, we first generated a ground truth tree T^* with c vertices uniformly at random using Prüfer sequences [95] and randomly distributed the n mutations to the c clones while ensuring that every clone had at least one mutation. Next, we generated clonal prevalences $\mathbf{u} = [u_i]$ by drawing from

ID	% of Trees	Clones	Mutations	Prevalence Noise	FNR β	Doublet δ
sim1a	100%	7	7	0%	0	0
sim1b	10%	7	7	0%	0	0
sim2a	100%	7	7	5%	0	0
sim2b	10%	7	7	5%	0	0
sim2c	100%	7	7	20%	0	0
sim3a	100%	7	7	5%	0.2	0.1
sim3b	10%	7	7	5%	0.2	0.1
sim4a	100%	10	100	5%	0.2	0.1

Table 2.1: **Simulation conditions.** We generate simulated data under eight conditions with 100 instances each. These conditions have varying subsets of candidate trees, number of clones, number of mutations per clone, clonal prevalence distortions, false negative and doublet rates. To analyze violations of the infinite sites assumption analysis, we introduced mutation losses in the sim1a instances. To analyze uncertainty in cancer cell fractions, we used the sim3a instances with a coverage of 1000x.

a symmetric $(n + 1)$ -dimensional Dirichlet distribution with concentration parameter 0.2. We used rejection sampling to ensure that each clonal prevalence u_i was at least 0.05. Let $\sigma(i)$ be the set of clones that contain mutation i . We generated frequencies $\mathbf{f} = [f_i]$ by setting $f_i = \sum_{j \in \sigma(i)} u_j$ for each mutation $i \in \{1, \dots, n\}$. We used the SPRUCE algorithm to enumerate the set $\mathcal{T}$ of trees given frequencies $\mathbf{f}$ [75].

To account for common single-cell sequencing errors, we varied false negative rates $\beta \in \{0, 0.2\}$ and doublet rates $\delta \in \{0, 0.1\}$. We generated, for each simulation instance, 10,000 single cells sampled under the specified false negative rates β and doublet rates δ according to the bulk clonal prevalence $\mathbf{u}$. To account for uncertainty in bulk sequencing, we additionally obtained confidence intervals on the cancer cell fractions of simulation instances sim3a, using a binomial distribution (with confidence $\alpha = 0.05$) and a mean coverage of 1000x (drawn from a Poisson distribution). Modeling additional uncertainty in bulk sequencing, sim4a instances consist of $n = 100$ mutations each and use PyClone [96] to cluster the 100 mutations before enumerating $\mathcal{T}$.

Recall that PhyDOSE has three model assumptions: (i) ground truth tree T^* is among the candidates trees $\mathcal{T}$, (ii) correspondence between clonal prevalences in bulk and subsequent single-cell sequencing samples, and (iii) infinite sites assumption for mutations. To assess (i), for simulation conditions ‘b’ (sim1b, sim2b and sim3b), we randomly sampled 10% of the trees outputted by SPRUCE [75]. To assess (ii), we varied single-cell clonal prevalences from the bulk clonal prevalences by resampling $\hat{\mathbf{u}} \sim \text{Dir}(\lambda \mathbf{u})$. We tuned the parameter λ so that the clonal prevalence varied by an absolute average of 5% and 20% from the

clones of the ground truth tree T^* , which resulted in $\lambda = 2000$ and $\lambda = 50$, respectively (Fig. A.3). To assess (iii), we introduced violations of the infinite sites assumption (ISA) in the form of mutation losses. Specifically, we introduced one mutation loss in each of the instances of sim1a as follows. First, we randomly picked two distinct mutations (i, j) where i is introduced prior to j in the ground truth tree T^* . Then, we designated the descendant mutation j as a loss of mutation i in each candidate tree $T \in \mathcal{T}$.

In total, we generated 100 simulation instances under eight varying conditions as specified in Table 2.1.

The benefit of PhyDOSE’s distinguishing feature analysis. We compared PhyDOSE against SCOPIT [50], an existing method to design SCDNA-SEQ experiments which takes as input a confidence level and the prevalence rate of each clone. Since SCOPIT does not connect the clones to be observed with a phylogenetic tree, we ran SCOPIT under two regimes. In the first regime (called ‘SCOPIT’), we input the clonal prevalence rates for each $T \in \mathcal{T}$ and take the maximum SCOPIT output of all trees as an upper bound. In the second regime (called ‘SCOPIT (true clones)’), we supplied SCOPIT with the prevalence rates of the simulated ground truth clones. The comparison to PhyDOSE was conducted at confidence level $\gamma = 0.95$. PhyDOSE yielded a significant reduction in the number of cells to sequence compared to SCOPIT (Fig. 2.4A). This is even the case when we provided the clonal prevalences of the ground truth tree to SCOPIT but not to PhyDOSE. In particular, for sim1a, SCOPIT required a median of 18.7 times as many cells than PhyDOSE, whereas SCOPIT (true clones) required 1.5 times as many cells (Fig. 2.4A). In absolute numbers, PhyDOSE computed a median number of $k^* = 35$ cells compared to 544 cells computed by SCOPIT and 59 cells computed by SCOPIT (true clones) (Fig. A.4).

To assess the accuracy of PhyDOSE’s k^* value, we generated follow-up *in silico* SCDNA-SEQ experiments. Specifically, we ran our approach for prioritizing candidate trees and SPhyR [42] on sampled single cells. For the former, we performed 100 experiments for each simulation instance, reporting the number of experiments that successfully recovered the ground truth tree T^* . We counted an experiment as successful if we correctly and uniquely selected the ground truth tree as T^* . For the latter, we also considered the performance of SPhyR when sampling half and double the number k^* of cells determined by PhyDOSE.

Fig. A.2 shows that the prioritization approach worked particularly well for sim1a with a median success rate of 96%. Similarly, SPhyR [42] was able to identify the true tree in the majority of cases after sampling PhyDOSE computed number k^* of cells. To quantify the similarity between the tree T estimated by SPhyR and the true tree T^* , we used two commonly-used tree distance metrics, ancestral and incomparable pair recall. *Ancestral pair*

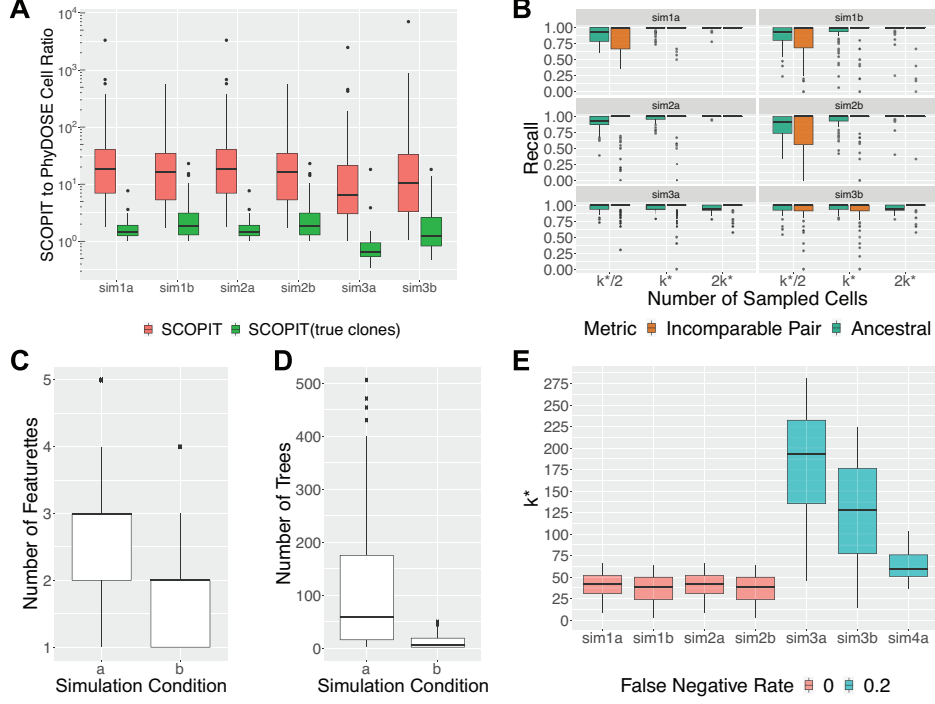


Figure 2.4: **Simulations demonstrate that PhyDOSE’s calculated number of single cells resolves tree ambiguity in bulk sequencing data.** We used confidence level $\gamma = 0.95$ to determine the number k^* of single cells to sequence. (A) SCOPIT to PhyDOSE cell ratio on a log scale when considering SCOPIT in a worst case regime where the true phylogeny is unknown and a best case regime where SCOPIT utilizes the clonal prevalence of the clones in the simulated ground truth tree. (B) Recall metrics of the tree inferred by SPhyR [42] by randomly sampling $k^*/2$, k^* and $2k^*$ simulated single cells. (C) Number $|\Pi|$ of featurettes among minimal distinguishing features Φ^* when compared between the enumerated candidate set $\mathcal{T}$ (conditions a) and the downsampled candidate set (conditions b). (D) Number $|\mathcal{T}|$ of trees in the candidate set when enumerated by SPRUCE [75] (condition a) and when downsampling the enumerated candidate set (condition b). (E) Number k^* of cells identified by PhyDOSE.

recall is defined as $|\mathcal{A}(T) \cap \mathcal{A}(T^*)|/|\mathcal{A}(T^*)|$ where $\mathcal{A}(T)$ ($\mathcal{A}(T^*)$) is the set of ordered pairs of mutations that occur on distinct edges of the same branch of T (T^*). *Incomparable pair recall* is defined as $|\mathcal{I}(T) \cap \mathcal{I}(T^*)|/|\mathcal{I}(T^*)|$ where $\mathcal{I}(T)$ ($\mathcal{I}(T^*)$) is the set of unordered pairs of mutation that occur on edges in distinct branches in T (T^*). For sim1a, the median of both metrics is 1 when sampling k^* cells, reflecting that SPhyR identified the true tree in the majority of cases (Fig. 2.4B). Moreover, we found greater gains in performance between sampling k^* cells versus $k^*/2$ cells than sampling $2k^*$ cells versus k^* cells.

Fig. 2.4C shows the number of clones in each minimal distinguishing feature identified by PhyDOSE, ranging from 1 to 5 with a median of 3 for simulation conditions ‘a’. Importantly, this number is smaller than the total number of 7 clones, demonstrating that a distinguishing

feature yields an efficient representation of a tree. This led to a smaller number k^* of cells inferred by PhyDOSE compared to SCOPIT without sacrificing performance in the tree reconstruction from the follow-up SCDNA-SEQ experiment.

Robustness to violations of PhyDOSE’s model assumptions. We used simulations to assess PhyDOSE’s performance when model assumptions are violated. We begin with the case where the ground truth tree T^* is not guaranteed to be present among the candidate trees $\mathcal{T}$. In sim1b, we downsampled the set of candidate trees of the instances in sim1a to 10% (Fig. 2.4D). Similarly to sim1a, PhyDOSE significantly reduced the required number k^* of single cells compared to SCOPIT (Fig. 2.4A). We analyzed follow-up *in silico* SCDNA-SEQ experiments using the prioritization approach and SPhyR [42]. Since the prioritization approach only considers candidate trees $\mathcal{T}$, which may not contain the ground truth tree, it is not surprising that performance dropped to a median success rate of 0%. However, SPhyR was able to identify the ground truth tree in the majority of cases (median of 1 for both incomparable and ancestral pair recall).

Next, we assessed the impact of clonal prevalence distortions between bulk and single cell data in sim2a, sim2b and sim2c. We found PhyDOSE to be robust to random clonal prevalence noise between bulk and single-cell sequencing as evidenced by a drop of only 1% in the median percentage of successful *in silico* SCDNA-SEQ experiment when there is no downsampling of trees (Fig. A.3). Additionally, the recall performance metrics (Fig. 2.4B, Fig. A.3) are also not substantially different between sim1a versus sim2a and sim2c, showing similar trends when using $k^*/2$ and $2k^*$ cells. We attribute this to the fact that PhyDOSE’s use of distinguishing features relies on the clonal prevalence of a few key clones. Furthermore, PhyDOSE performs well for the case where candidate trees have been downsampled and clonal prevalences have been distorted (sim2b, see Fig. 2.4B).

Finally, we assessed the sensitivity of PhyDOSE to ISA violations by applying it to a candidate set $\mathcal{T}$ of 1-Dollo phylogenies obtained from sim1a. Specifically, the resulting simulation instances had candidate trees composed of 6 mutations, one of which having undergone a loss. We performed 100 *in silico* experiments for each of the 100 simulation instances. Like sim1a under the ISA, the median percentage of successful experiments is 95% for the 1-Dollo phylogenies (Appendix A.8A). However, 23 simulation instances had a success rate of 0%. This increased variance is to be expected as distinguishing features may no longer be distinguishable from those of other trees when mutation losses occur. Nevertheless, PhyDOSE’s suggested SCDNA-SEQ experiments for these 23 instances significantly reduced the number of candidate trees from a median of 161 trees pre-experiment to a median of 5 trees post-experiment (Appendix A.8B). In all cases, the ground truth tree was included in the

candidate set after performing the SCDNA-SEQ experiment. Hence, as distinguishing features provide an efficient representation for each tree using only a subset of clones, PhyDOSE performed well on data with violations of the infinite sites assumption.

Robustness to uncertainty due to sequencing errors. Sequencing errors occur in both the initial bulk sequencing experiment as well as the follow-up single-cell sequencing experiment. We begin by considering common SCDNA-SEQ errors using false negative rate $\beta = 0.2$ and doublet rate $\delta = 0.1$ in sim3a and sim3b. While the number k^* significantly increased compared to the simulations without false negatives (Fig. 2.4E), PhyDOSE’s computed number k^* of cells remained an order of magnitude smaller than SCOPIT but not for SCOPIT (true clones) (Fig. 2.4). The latter is due to PhyDOSE’s adjustments of the clonal prevalence when factoring in a false negative rate of $\beta = 0.2$ in trees other than T^* (A.5). Tree inference using SPhyR in subsequent SCDNA-SEQ experiments based on PhyDOSE’s k^* identified the ground truth tree in the majority of cases (median of 1 for both incomparable and ancestral pair recall).

Fig. A.3 shows PhyDOSE performance in sim4a instances with mutation clusters inferred by PyClone [96]. We included the clustered pair recall in our analysis defined as $|\mathcal{C}(T) \cap \mathcal{C}(T^*)|/|\mathcal{C}(T^*)|$, where $\mathcal{C}(T)$ ($\mathcal{C}(T^*)$) is defined as the set of unordered pairs of mutations that are introduced on the same edge in T (T^*). At k^* cells, the median ancestral pair recall was 0.96, the incomparable pair recall was 0.86 and the clustered pair recall was 0.94, showing a reduction in performance from the first sim1, sim2 and sim3 simulations due to the additional errors introduced by PyClone [96].

Finally, we evaluated the performance of PhyDOSE in the presence of cancer cell fraction uncertainty by adapting sim3a instances. We simulated variant and total read counts with a coverage of 1000x. Then, we constructed a binomial proportion 95%-confidence interval using Jeffrey’s prior interval [97]. Next, we used the mean of this interval to enumerate the candidate set of trees with SPRUCE [34]. Using the method described in Appendix A.2.2, we constructed a confidence interval $[k_-, k_+]$ for each replication (Fig. A.9). We performed 100 *in silico* for each replication using both k_- and k_+ , selecting the tree $T \in \mathcal{T}$ with the maximum support(T) as T^* . We obtained a median percentage of successful trials of 86% (IQR: 44% – 93%) and 86% (IQR: 31% – 98%) when randomly sampling k_- and k_+ *in silico* cells, respectively (Fig. A.9).

paragraphRunning time. We performed an empirical run time analysis on a server with two Intel Xeon Gold 5120 CPUs @ 2.20GHz and 512 GB RAM. Performing the power calculation of k^* is fast [50] when $|\Phi| = 1$ and the median of $|\Phi|$ in our simulations was 1. Therefore, the main bottleneck in PhyDOSE is the determination of Φ for each tree in the candidate using

the algorithm presented in Appendix A.2. Note that this step is embarrassingly parallelized because Φ is computed independently for each $T \in \mathcal{T}$. However, we leave the parallel `phydoser` implementation as future work. To additionally explore how PhyDOSE scales with $|\mathcal{T}|$, we generated an additional simulation set of with 10 mutations and no mutations clusters and set a ten minute time limit on finding the distinguishing features. We calculated the runtime in seconds for each simulation instance when Φ is solved sequentially for each $T \in \mathcal{T}$. The largest input size to complete was 9901 required 619 seconds with 92% of the time spent on finding the distinguishing features. The results are displayed in Fig. A.10.

In summary, our simulations demonstrate that PhyDOSE’s distinguishing feature analysis results in significantly fewer cells to sequence than SCOPIT [50] without a subsequent loss in power to identify the true phylogeny. Moreover, we find that PhyDOSE is robust to typical sequencing errors in both the bulk and SCDNA-SEQ data as well as violations of model assumptions.

2.3.2 Retrospective analysis of an acute lymphoblastic leukemia patient

We considered a cohort of six childhood acute lymphoblastic leukemia (ALL) patients whose blood was sequenced using bulk and targeted single-cell DNA sequencing [85]. The number of sequenced single cells per patient varied between 96 and 150. To validate our approach, we used PhyDOSE to calculate the number $k(T^*)$ of cells needed to identify the true phylogeny T^* that is consistent with both data types, thereby retrospectively determining whether fewer single cells suffice to determine T^* , decreasing the cost of replicate experiments. In addition, we assessed whether the calculated number $k(T^*)$ yielded T^* using *in silico* SCDNA-SEQ experiments.

Due to the absence of published copy-number aberration information for this dataset, we focused our attention on patient 2 whose single-cell phylogeny adhered to the infinite sites assumption and the variant allele frequencies suggested the absence of copy-number aberrations (as detailed in Appendix A.3.1). For this patient, 16 autosomal mutations in 115 cells were sequenced [85]. We note that the authors had no knowledge of the number of cells that would suffice to infer the tumor phylogeny of the patient. Using the infinite sites assumption and assuming the absence of copy-number aberrations, we define the cancer cell fraction, or frequency f_i of each mutation i in the bulk data as $2 \cdot \text{VAF}(i)$. We define the *SCDNA-SEQ mutation frequency* as the fraction of single cells that harbor the mutation. Strikingly, there is a clear correlation between the bulk and SCDNA-SEQ mutation frequencies, supporting PhyDOSE’s first assumption (Fig. 2.5A). We excluded mutation *CMTM8* because of a notable discrepancy in frequencies (0.4 in bulk vs. 0.2 in SCDNA-SEQ). Using SPRUCE [75],

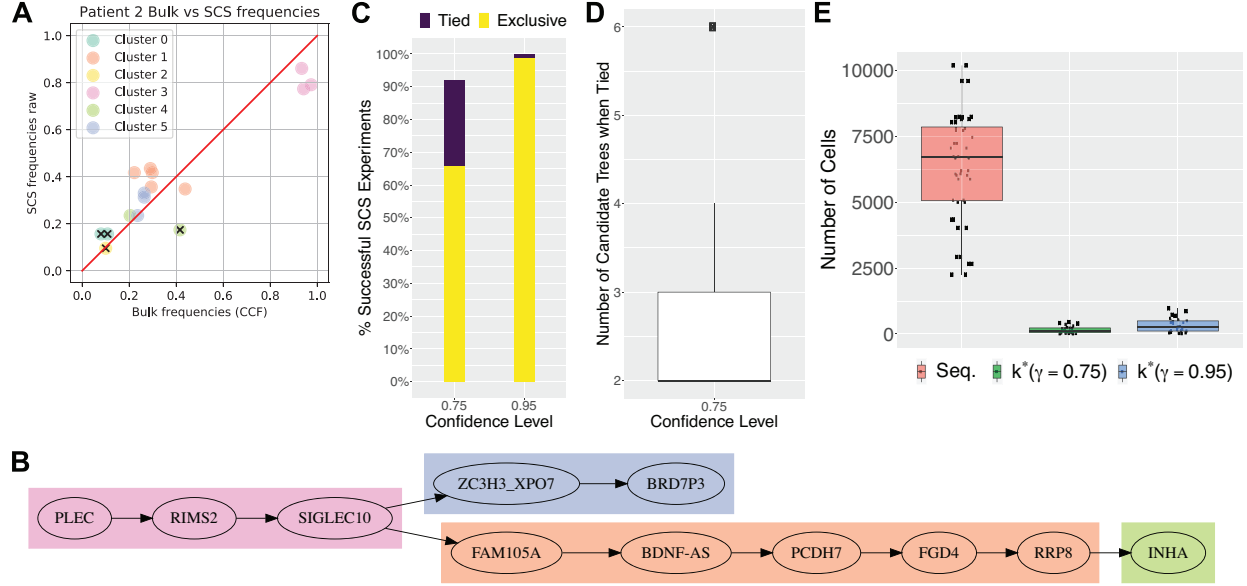


Figure 2.5: **Retrospective analysis of ALL patient 2 [85] and AML cohort [88] demonstrates that fewer cells suffice for replication.** Panels (a)-(d) consider ALL patient 2 [85] and panel (e) considers the AML cohort [88]. (A) There is a strong correlation between bulk and single-cell mutation frequencies. Colors indicate mutation clusters from SCDNA-SEQ data and excluded mutations are indicated by ‘x’. (B) Phylogeny T^* that is consistent with the SCDNA-SEQ and bulk data. (C) Percent of successful outcomes in 100 *in silico* SCDNA-SEQ experiments, obtained by sampling from the 115 sequenced cells without replacement following PhyDOSE’s calculated number $k(T^*)$ of cells (103 for $\gamma = 0.95$ and 50 for $\gamma = 0.75$). Exclusive outcomes (yellow) uniquely identified T^* whereas tied outcomes (purple) yielded a small set of candidate phylogenies that include T^* . (D) Number of candidate phylogenies in the case of ties. (E) The distribution of PhyDOSE’s k^* for $\gamma \in \{0.75, 0.95\}$ of all patients in the AML cohort with $|\mathcal{T}| > 2$ as well as the number of cells that were originally sequenced.

we enumerated the set $\mathcal{T}$ of trees from the bulk data, yielding over 2.5 million trees. This number is mainly driven by 3 mutations (*ATRNL1*, *LINC00052* and *TRRAP*) with a VAF less than 0.05. Excluding these 3 mutations resulted in a more tractable number of 2,576 trees. We note that in practice we may similarly exclude mutations because of very low VAFs or less importance in downstream analyses. Fig. 2.5B shows the single tree $T^* \in \mathcal{T}$ that was consistent with the cleaned single-cell data, supporting PhyDOSE’s second assumption.

We ran PhyDOSE using varying confidence levels $\gamma \in \{0.75, 0.95\}$ and an estimated false negative rate of $\beta = 0.2$ reported by the authors [85]. PhyDOSE calculated that $k(T^*) = 103$ cells suffice to identify T^* with confidence level $\gamma = 0.95$. Indeed, performing 100 *in silico* SCDNA-SEQ experiments, by sampling $k(T^*)$ cells among the 115 sequenced cells without replacement, yielded a success rate of 99% (Fig. 2.5C).

To reduce costs, we explored what would have happened retrospectively with a lower confidence level γ of 0.75. PhyDOSE calculated that $k(T^*) = 50$ cells are needed for $\gamma = 0.75$, which is a significant cost savings over $\gamma = 0.95$. Performing 100 *in silico* SCDNA-SEQ experiments yielded a success rate of uniquely identifying T^* of 66%, which was lower than the expected rate of 75%. Furthermore, we noted that in an additional 26% of experiments the correct phylogeny T^* was among the trees with the highest overall support (Fig. 2.5C). The number of trees in the tied set of successes varied from 2 to 6 (Fig. A.12), showing that although PhyDOSE did not uniquely identify the tree, it was able to significantly reduce the original set of 2576 trees (Fig. A.12 and Fig. A.13).

In summary, this retrospective analysis shows that the true tree for patient 2 could have been identified confidently with fewer cells than the 115 cells initially sequenced [85]. With a lower confidence level γ , PhyDOSE computes that far fewer cells are required, significantly reducing costs but at the expense of a lower success rate of uniquely identifying the true phylogeny. Nevertheless, the resulting SCDNA-SEQ experiment will eliminate a large fraction of the original set of candidate phylogenies due to the incorporation of distinguishing features in the PhyDOSE power calculation.

2.3.3 Retrospective analysis of an acute myeloid leukemia cohort

Morita et al. [88] performed high-throughput targeted microfluidic single-sequencing using the Tapestry platform [11] on a cohort of 77 patients with acute myeloid leukemia (AML). The authors additionally performed bulk sequencing in order to confirm the presence of a mutation in the single-cell data. We note that the authors restricted their analysis to somatic mutations (SNVs and indels) that did not occur in regions affected by additional copy number aberrations.

Here, we utilized the published bulk sequencing VAFs of the SNVs in each patient, eliminating any mutations not detected via bulk sequencing, to enumerate a set of candidate trees using SPRUCE [75]. We restrict our analysis to the 24 patients where bulk sequencing data was available and SPRUCE identified more than one candidate tree. The median number of mutations for these patients was 4 (IQR: 3-5). We retrospectively used PhyDOSE at confidence levels $\gamma \in \{0.75, 0.95\}$ to estimate the cells needed to perform an equivalent single-cell experiment. We used false negative rate $\beta = 0.049$, which is the mean of the per patient published false negative rate. In the original study, a median of 7,584 cells per patient (IQR: 6,194 – 8,361) were sequenced. Fig. 2.5E shows the distribution of PhyDOSE k^* for the 24 patients (median is 2, IQR: 2 – 6, max is 316) at $\gamma \in \{0.75, 0.95\}$ versus the total number of cells sequenced in [88]. For $\gamma = 0.95$, the median value of k^* was 274 cells

(IQR: 230 – 497). This is a significant reduction from the number of cells sequenced per patient in [88] with a median percent reduction at confidence level $\gamma = 0.95$ of 95.4% (IQR: 92.2% – 98.0%)(Table A.1).

For these 24 patients, Morita et al. [88] sequenced 153,558 cells while the PhyDOSE design at confidence level $\gamma = 0.95$ requires 8,144 cells (Table A.1). Using that a Tapestri run of 10,000 cells costs \$795 and including additional sequencing costs of \$200 per run with NovaSeq or \$1000 per run with MiSeq [98, 99], we estimate the costs of the original study as \$15,920 and the estimated costs of the PhyDOSE design as \$1,995. This assumes the original study utilized 16 Tapestri runs with NovaSeq while PhyDOSE requires 1 Tapestri run with the more expensive MiSeq to avoid multiplexing. Thus, designing the experiment with PhyDOSE would have yielded a 93.75% cost savings. Further, we note that this study design requires targeted sequencing and that the potential cost savings when the experimental design requires whole genome sequencing would further increase.

2.3.4 Prospective analysis of a non-small cell lung cancer cohort

Using PhyDOSE, we prospectively determined the number of cells needed to uniquely identify the true phylogeny for the 25 out of 100 patients in the TRACERx non-small-cell lung cancer cohort that have multiple candidate trees [89]. The authors previously identified the set $\mathcal{T}$ of candidate trees for each patient using CITUP [76] after clustering mutations with PyClone [96]. The authors also reported the cancer cell fraction of each mutation cluster in each bulk sample. The number of trees in the candidate set for each patient ranged from 2 to 17, with each containing mutation clusters with between 5 and 882 mutations (Table A.2).

Assuming high confidence on the co-occurrence of mutations in a cluster, mutation clusters alleviate the issue of false negatives, i.e. it suffices to only observe a small number of mutations to impute the presence of the other mutations in the same cluster. Here, with a typical SCDNA-SEQ false negative rate of 0.2, the probability of all mutations in the smallest cluster (with size 5) dropping out thus equals $0.2^5 = 0.00032$, a probability that can be neglected. As such, we set $\beta = 0$. Unlike in the simulations and the previous real datasets, multiple bulk samples corresponding to distinct spatial locations were available for analysis per patient. In addition to the naive method where we select a single biopsy that minimizes k^* , we used the multiple biopsy heuristic to infer numbers $\mathbf{k}^*$ of cells for each biopsy. For both methods, we used confidence level $\gamma = 0.95$.

Following the naive approach, PhyDOSE returned a finite value of k^* for 24 out of the 25 patients. The naive approach yielded $k^* = \infty$ for patient CRUK0037 because for each of the 5 biopsies there is a tree where every distinguishing feature is not observable. That

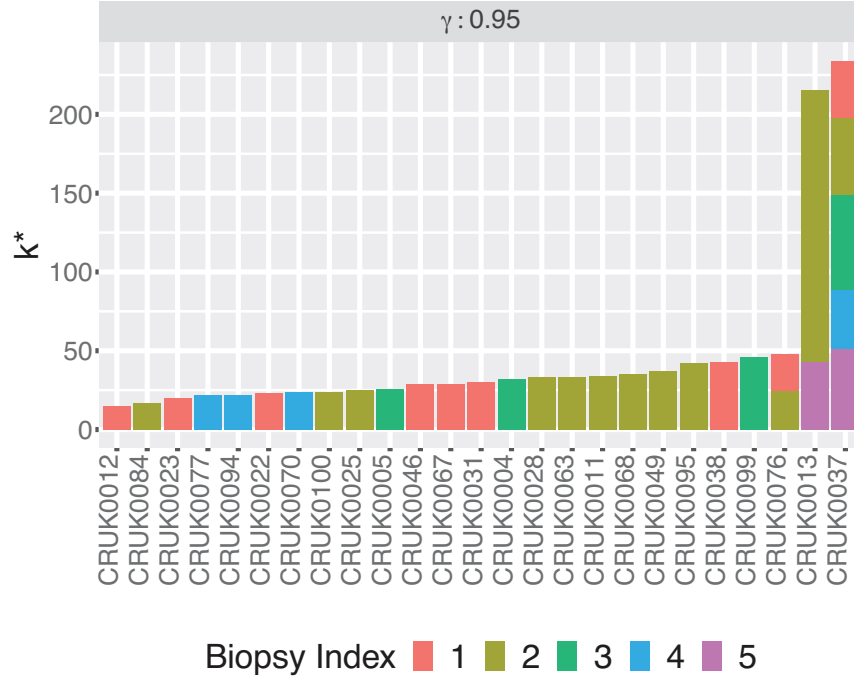


Figure 2.6: **PhyDOSE multiple biopsy heuristic calculated numbers k^* of cells per biopsy for the lung cancer cohort [89] at confidence level $\gamma = 0.95$.**

is, the clonal prevalence of one of the comprising featurettes is 0. By contrast, the heuristic calculated a total of 243 cells (R1: 36; R2: 49; R3: 60; R4: 38 and R5: 51 cells) for this patient. For patients CRUK0013 and CRUK0076, the naive approach required the sequencing of more cells from a single biopsy than the multiple biopsy heuristic (CRUK0013: 1,051 vs. 215 cells; CRUK0076 47,479 vs. 48 cells). For the remaining 22 patients, treating the samples independently yields the same number of cells from the selected biopsy as the heuristic. Table A.2 and Fig. 2.6 provide detailed numbers.

These strikingly low values of total number of cells for the 25 patients with multiple candidate trees and multiple biopsies demonstrate the benefit of using PhyDOSE to strategically optimize the design of follow-up single cell experiments.

2.4 DISCUSSION

In this work, we showed that the mutation frequencies $\mathbf{f}$ and the set $\mathcal{T}$ of tumor phylogenies inferred from initial bulk data contain valuable information to provide guidance for follow-up SCDNA-SEQ experiments. We introduced PhyDOSE, a method to calculate the number k^* of single cells needed to infer the true phylogeny T^* given $\mathbf{f}$, $\mathcal{T}$ and a user-specified confidence level γ . Underpinning our method is the observation that often only a subset of

clones suffices to distinguish one tree $T \in \mathcal{T}$ from the remaining trees $\mathcal{T} \setminus \{T\}$. Although PhyDOSE is motivated by the output of deconvolution methods for bulk sequencing, it is agnostic to the method used to obtain the candidate set as long as the clonal prevalence rates of the distinguishing features can be estimated. Thus, the input set $\mathcal{T}$ of candidate trees can be obtained from preliminary single-cell and/or bulk sequencing data. Similarly, PhyDOSE is agnostic to the phylogeny inference method used to analyze data from the proposed SCDNA-SEQ experiment. We also provided heuristics for realistic scenarios that arise in practice, such as handling uncertainty in the estimation of cancer cell fractions and the availability of multiple biopsies.

We validated PhyDOSE using simulations and a retrospective analysis of leukemia patients [85, 88], concluding that PhyDOSE’s computed number k^* of cells resolves tree ambiguity, even in the presence of SCDNA-SEQ errors. Our simulations showed that PhyDOSE remains robust in the presence of sequencing errors and violations of model assumptions, outperforming the competing method, SCOPIT [50]. In a prospective analysis, we demonstrated that only a small number of cells suffice to disambiguate the solution space of trees in a recent non-small cell lung cancer cohort [89]. In summary, PhyDOSE proposes cost-efficient SCDNA-SEQ experiments that will yield high-fidelity phylogenies, which may consequently improve downstream analyses in cancer genomics aimed at deepening our understanding of cancer biology.

There are several future research directions. First, in the case of multiple bulk samples, although we propose an exact calculation, we only implement a heuristic since the exact calculation does not scale to realistic problem sizes. Developing an implementation of the exact calculation in the case of multiple samples would yield a further cost reduction in the experimental design since the heuristic overestimates the number of cells at given confidence level γ . Second, to further reduce SCDNA-SEQ costs, we might want to include a mutation selection step as part of our approach to perform targeted rather than whole-genome sequencing. Third, similar ideas can be used to design follow-up sequencing experiments using alternative sequencing technologies such as long read sequencing. Alternatively, performing additional bulk sequencing rather than single-cell sequencing might be more cost-effective, especially when obtaining a bulk sample with distinct clonal prevalences [34, 37]. Fourth, we plan to develop an easy-to-use Shiny user interface to facilitate the use of PhyDOSE for the design of sequencing experiments. Fifth, to improve robustness in the presence of SCDNA-SEQ errors, we plan to explore alternative definitions of successful SCDNA-SEQ experiment outcomes, requiring that more than one cells is observed of each featurette of a distinguishing feature. This will enable us to address errors such as doublets and false positives in an SCDNA-SEQ experiment. Similar ideas can be used to address uncertainty in mutation

clusters inferred from bulk sequencing data. Sixth, the concept of distinguishing features may be useful to summarize diverse solution spaces in cancer phylogenetics [100]. Finally, we plan to explore evolutionary models beyond the infinite sites model, such as the Dollo parsimony model where mutations might be lost [42], requiring a more careful approach to find the distinguishing features of a tree.

CHAPTER 3: DOUBLET²D: DETECTING DOUBLETS IN SINGLE-CELL DNA SEQUENCING DATA

3.1 INTRODUCTION

The smaller amount of DNA material within a cell compared to RNA poses additional sequencing challenges than those faced in single-cell RNA sequencing (scRNA-seq) [102]. Medium to high coverage SCDNA-SEQ technology, suitable for detecting single-nucleotide variants, suffers from elevated rates of technical errors due to whole-genome amplification that may impact downstream analyses, including allelic dropout (ADO), copying mistakes in the amplification reaction, unbalanced amplification and doublets. Specifically, when ADO occurs, one or more of the alleles may fail to be amplified during the early stages of the process and thus the allele is said to “drop out” prior to sequencing. While technological advances have decreased the frequency of these errors, one remaining technical challenge is when multiple cells, or *multiplets*, are captured within a droplet and linked to a single barcode making all subsequent reads appearing as if they originated from one cell. To mitigate this effect, practitioners utilize a Poisson distribution to estimate the probability that a droplet contains a specified number of cells. The rate parameter of the Poisson distribution is then determined by a function of the cell solution concentration and droplet volume to obtain the desired probability of multiplets [103]. This results in the majority of droplets containing zero cells and multiplets with more than two cells are rare. However, *doublets*, which are droplets containing two cells, occur frequently and are therefore the focus of this work [16, 104, 105].

Adapting terminology from the scRNA-seq literature [51], we introduce three categories for doublets in SCDNA-SEQ: (i) selflet, (ii) nested and (iii) neotypic (Fig. 3.1a). *Selflets* are comprised of cells with identical genotypes. *Nested* doublets occur when the set of mutations in one cell is a proper subset of the mutations in the other cell. A *neotypic* doublet is a doublet that is not nested or a selflet and implies the existence of a novel genotype not present in the sample. Neotypic doublets thus distort the signal of mutation co-occurrence patterns and makes it challenging to distinguish the presence of rare clones, that may be resistant to certain treatments, from a neotypic doublet [11]. Although nested doublets and selflets will not impact the analysis of mutation co-occurrence or mutual exclusivity patterns, they may impact the estimation of clonal abundances, which are used to model both the

²This chapter contains identical material with an adapted introduction previously published in “doubletD: detecting doublets in single-cell DNA sequencing data” [101]. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with M. El-Kebir and P. Sashittal, MEK conceived the project; LLW, MEK and PS establish theory; LLW performed experiments; LLW, MEK and PS contributed to the writing of this chapter.

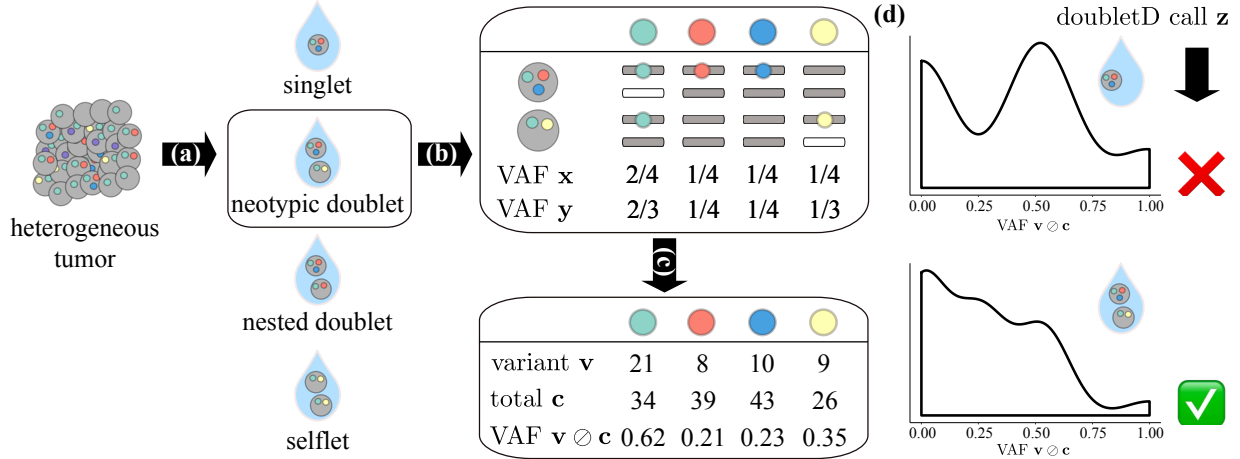


Figure 3.1: **doubletD calls doublets in medium to high coverage scDNA-SEQ data.** (a) The first step of most single-cell sequencing technologies involves cell capture where the goal is to encapsulate single cells into droplets, known as *singlets*. However, errors in this process (details in Section 3.3) can lead to three kind of doublets – *neotypic* doublets, *nested* doublets and *selflets*. (b) The cells in each isolated droplet i undergo whole-genome amplification and sequencing independently. These processes introduce errors such as allelic dropouts and imbalance in amplification. (c) The resulting aligned reads are used for variant calling yielding alternate $v_{i,j}$ and total $c_{i,j}$ read counts at each locus of interest j . (d) DOUBLET D uses the observed variant allele frequencies $v_{i,j}/c_{i,j}$ as the key signal, while accounting for sequencing and amplification errors to detect doublets in the sample. The symbol $\oslash$ denotes element-wise division.

evolutionary trajectory and the fitness landscape of a tumor [106, 107].

While there are downstream analysis methods, such as genotype and/or phylogeny inference methods, that account for the presence of doublets, to the best of our knowledge, there exists no standalone method for doublet detection in scDNA-SEQ data. There are a number of drawbacks to methods that jointly infer the doublets during any downstream analysis. First, methods like ∞ SCITE [108], SCG [40] and SiCloneFit [43] utilize Bayesian inference in the form of Markov chain Monte Carlo (MCMC) or variational inference, which scale poorly with the inclusion of doublets and size of the input [40, 43, 108]. Second, methods, such as ScisTree [109], are able to identify doublets only under the infinite sites model of evolution. Third, most methods require a binarized or discretized experiment by loci matrix input as opposed to positional variant and reference allele read counts. This results in the loss of useful information for doublet identification. Lastly, as a result of the discrete input and/or utilizing the infinite sites assumption, methods that do identify doublets are at best only able to identify neotypic doublets.

In contrast, there exist a number of standalone methods for detecting doublets in single-cell

RNA sequencing data. [51, 52, 110]. See [111] for an excellent overview and benchmarking of scRNA-seq doublet detection methods. Doublets in single-cell RNA sequencing (scRNA-seq) result in the observation of neotypic gene expression profiles, which impacts cell clustering and the identification of cell-state trajectories ([111]). In general, these methods follow a four step process. First, simulated doublets are created by mixing observed gene expression profiles. Second, the observed and simulated data are embedded into a latent space using dimensionality reduction. Third, machine learning methods are used to estimate the probability that a droplet is a doublet. Finally, a threshold scheme is enacted based on knowledge of the experimental doublet rate to classify experiments as either a singlet or doublet. The main variation within these methods is the choice of embedding/dimension reduction and classifier. Additionally, these methods are designed to capture neotypic doublets and struggle to identify embedded doublets, which are often located within clusters of singlets in the embedded space. While it is possible to directly apply scRNA-seq doublet detection methods on DNA variant read counts, such methods do not properly account for the distinct error profile of SCDNA-SEQ data.

As a first step in addressing the need for a fast, standalone method for SCDNA-SEQ doublet detection, we introduce doubletD, which performs doublet detection in medium to high coverage SCDNA-SEQ data. Critically, doubletD does not make any assumptions about the model of evolution, the number of distinct clones or assume a threshold on the minimum clonal abundance in the sample. doubletD operates directly on variant and reference allele counts without the need to discretize the input, thus retaining a critical signal for doublet detection in the form of the variant allele frequency (VAF) (Fig. 3.1c). Specifically, underlying doubletD is the observation that doublets in SCDNA-SEQ data have a characteristic VAF spectrum due to increased number of copies and/or allelic dropout (Fig. 3.1d). Others have noted the presence of some of these characteristics in a *post hoc* analysis of either single-nucleotide variant [112] or copy-number aberration calling [113]. doubletD considers each droplet independently but borrows strength from the entire dataset while using a maximum likelihood approach in order to rapidly classify an experiment as either a doublet or singlet prior to downstream analyses. We demonstrate on both simulated and real datasets that these design choices allow doubletD to be utilized in conjunction with any downstream analysis of choice and therefore obviates the need for more complex downstream methods to individually account for the presence of doublets within their own models.

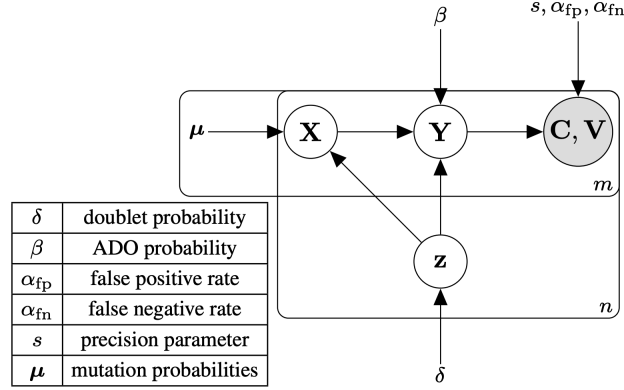


Figure 3.2: **Plate diagram of the doubletD's graphical model.** Observed total and variant read counts ($\mathbf{C}, \mathbf{V}$) of m loci in n droplets are affected by doublet status $\mathbf{z}$, allelic dropout and additional errors during sequencing.

3.2 MATERIALS AND METHODS

3.2.1 Generative model

Similarly to scRNA-seq, there are two main types of high-throughput cell capture strategies in SCDNA-SEQ: microfluidics and well-based protocols, which, respectively, distribute a cell suspension into either droplets or wells [114, 115]. Here, we use the term ‘droplet’ independent of the used technology. Consider a SCDNA-SEQ experiment with n droplets and m mutation loci that were identified after read alignment and variant calling. Each mutation locus has two alleles: a reference allele and a variant allele. Thus, we are given $\mathbf{C} = [c_{i,j}] \in \mathbb{N}^{n \times m}$ total read counts and $\mathbf{V} = [v_{i,j}] \in \mathbb{N}^{n \times m}$ variant counts, which are independent across droplets and loci. Read counts $v_{i,j}$ and $c_{i,j}$ of mutation locus j in droplet i are affected by (i) whether droplet i is a doublet (Section 3.2.1), (ii) the genotype(s) at locus j in the droplet (Section 3.2.1), and errors during sequencing including (iii) allelic dropout (Section 3.2.1) and (iv) amplification bias and sequencing errors (Section 3.2.1). We make these relationships explicit in a generative model for $\mathbf{C}$ and $\mathbf{V}$ (Fig. 3.2).

Doublet model. In the following we will define random variables $\mathbf{z} \in \{0, 1\}^n$, where z_i indicates whether droplet i is a doublet (i.e. $z_i = 1$) or a singlet (i.e. $z_i = 0$). During the capture step, cells are released into a nozzle with a constant rate r and there is a fixed time-interval t in which a droplet is formed. The number of cells in a droplet is given by the number of cells that enter the nozzle in the time-interval during which the droplet is formed. Therefore, the prior on the doublet probability is a Poisson distribution with mean $\lambda = rt$. Moreover, only non-empty droplets will yield sequence reads. This combined with the fact

that doublets are composed of two cells, we have that $z_i = 1$, i.e. the event of droplet i being a doublet, equals

$$P(z_i = 1) = \frac{\Lambda(2; \lambda)}{\sum_{k=1}^{\infty} \Lambda(k; \lambda)} = \frac{\Lambda(2; \lambda)}{1 - \Lambda(0; \lambda)}, \quad (3.1)$$

where $\Lambda(k; \lambda)$ is the probability of $k \in \mathbb{N}$ occurrences (here cells) under a Poisson distribution with mean λ . In practice rt is very small (i.e. $\lambda \ll 1$), and thus the mass of the Poisson distribution $\Lambda(k; \lambda)$ is concentrated around two outcomes $k \in \{1, 2\}$. Therefore, z_i can be approximately modeled by a Bernoulli distribution with probability of success $\delta = \Lambda(2; \lambda) / (\Lambda(1; \lambda) + \Lambda(2; \lambda))$ so that

$$P(z_i = 1) = \delta. \quad (3.2)$$

Considering independence between distinct droplets, we get

$$P(\mathbf{z}) = \prod_{i=1}^n \delta^{z_i} (1 - \delta)^{(1-z_i)}. \quad (3.3)$$

Genotype model. We make the simplifying assumption that each mutation locus has copy number 2 in a single cell — we show robustness of violations to this assumption in Section 3.3.1. Thus the genotype of a locus j in a single cell can be in one of three states: (i) wild type (wt) where both copies have the reference allele, (ii) heterozygous (het) with one variant and one reference copy, and (iii) homozygous (hom) where both copies have the variant allele. Let $\mu_{\text{wt},j}$, $\mu_{\text{het},j}$ and $\mu_{\text{hom},j}$ be the mutation probabilities at locus j of the three types, respectively, such that $\mu_{\text{wt},j} + \mu_{\text{het},j} + \mu_{\text{hom},j} = 1$. Let $x_{i,j}$ indicate the variant allele frequency (VAF) at locus j in droplet i . In case i is a singlet, we have that $x_{i,j} \in \Sigma_{\text{singlet}}$ where $\Sigma_{\text{singlet}} = \{0, 1/2, 1\}$ for any locus j . On the other hand, if i is a doublet, we have that $x_{i,j} \in \Sigma_{\text{doublet}}$ where $\Sigma_{\text{doublet}} = \{0, 1/4, 1/2, 3/4, 1\}$ for any locus j . For a droplet i comprising of a single cell ($z_i = 0$), the probability $P(x_{i,j} \mid z_i = 0)$ equals

$$P(x_{i,j} \mid z_i = 0) = \begin{cases} \mu_{\text{wt},j}, & \text{if } x_{i,j} = 0, \\ \mu_{\text{het},j}, & \text{if } x_{i,j} = 1/2, \\ \mu_{\text{hom},j}, & \text{if } x_{i,j} = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (3.4)$$

Following current single-cell literature [116, 117], we assume that a doublet contains two

cells with independent genotypes. Therefore, we may define $P(x_{i,j} \mid z_i = 1)$ using probabilities $P(x_{i,j} \mid z_i = 0)$ as

$$\frac{\sum_{g,h \in S(f)} P(x_{i,j} = g \mid z_i = 0) P(x_{i,j} = h \mid z_i = 0)}{\sum_{g,h \in \Sigma_{\text{singlet}} \times \Sigma_{\text{singlet}}} P(x_{i,j} = g \mid z_i = 0) P(x_{i,j} = h \mid z_i = 0)}, \quad (3.5)$$

where $S(f) = \{(g, h) \in \Sigma_{\text{singlet}} \times \Sigma_{\text{singlet}} \mid 2g + 2h = 4f\}$ gives all pairs (g, h) of VAFs in Σ_{singlet} that result in the doublet VAF f . For example, a doublet VAF $f = 1/2$ results from two cells with pairs (g, h) of VAFs in the set $S(1/2) = \{(1/2, 1/2), (1, 0), (0, 1)\}$.

Allelic dropout model. We follow the work in [117, 118] to model the shift in variant allele frequency due to allelic dropouts (ADO). In this model, ADO is introduced by deciding for each cell whether a given allele is amplified or not according to a specific probability β known as the ADO rate. Dropout of distinct alleles is assumed to be independent and the ADO rate β is assumed to be constant for all cells and all loci. Although this could be easily extended to account for site-specific ADO as considered in other work [119], here we opt for a global allelic dropout rate to reduce the number of parameters. The VAF $y_{i,j}$ at locus j in droplet i after the dropout event depends on the VAF $x_{i,j}$ and doublet indicator z_i (Fig. 3.2). Specifically, each possible pair $(x_{i,j}, z_i)$, where $x_{i,j} \in \Sigma_{\text{singlet}}$ when $z_i = 0$ and $x_{i,j} \in \Sigma_{\text{doublet}}$ when $z_i = 1$, can yield varying $y_{i,j}$ with probabilities that depend on the number of alleles that are dropped during amplification. Using that each mutation locus has copy number 2 in a single cell and allowing any number of copies to drop out, we have $y_{i,j} \in \Theta_{\text{singlet}}$ where $\Theta_{\text{singlet}} = \{0, 1/2, 1\}$ if droplet i is a singlet. Conversely, if i is a doublet, we have $y_{i,j} \in \Theta_{\text{doublet}}$ where $\Theta_{\text{doublet}} = \{0, 1/4, 1/3, 1/2, 2/3, 3/4, 1\}$. Table B.1 lists all values of $P(y_{i,j} \mid x_{i,j}, z_i)$ for varying $(x_{i,j}, z_i)$ and given ADO rate β . Fig. B.1 shows an illustrative example of allelic dropout.

Read count model. Beyond ADO, there are two types of additional errors that affect read counts $(c_{i,j}, v_{i,j})$ and lead to an observed VAF $v_{i,j}/c_{i,j}$ that differs from the latent VAF $y_{i,j}$ after ADO: (i) copy errors, which occur early during PCR and lead to a propagation of incorrect nucleotides, and (ii) allelic imbalance, where amplification is biased towards one of the alleles [102]. We model the resulting overdispersion with a beta-binomial as is standard in the field [116, 117, 119]. We use an uninformative prior on total read counts $c_{i,j}$ yielding

$$P(c_{i,j}, v_{i,j} \mid y_{i,j}) = P(v_{i,j} \mid c_{i,j}, y_{i,j}) P(c_{i,j}) \propto P(v_{i,j} \mid c_{i,j}, y_{i,j}). \quad (3.6)$$

While copy errors and uneven amplification errors happen simultaneously during the am-

plification stage, here, following [119], we employ a simpler model that assumes that the copy errors precede the allelic imbalance during amplification. We capture copy errors using a specified false positive rate α_{fp} , which is the probability of generating an alternate allele in the copy when the template has the reference allele, and false negative rate α_{fn} , which is the probability of generating a reference allele in the copy when the template has the alternate allele. Specifically, the probability $p_{i,j}$ of producing a copy with the alternate allele at locus j in experiment i is given by

$$p_{i,j} = y_{i,j}(1 - \alpha_{\text{fn}}) + (1 - y_{i,j})\alpha_{\text{fp}} = \alpha_{\text{fp}} + (1 - \alpha_{\text{fp}} - \alpha_{\text{fn}})y_{i,j}. \quad (3.7)$$

The number $v_{i,j}$ of variant reads resulting after amplification in presence of allelic imbalance is modeled by the following beta-binomial distribution.

$$\pi_{i,j} \sim \text{beta}(p_{i,j}, s), \quad (3.8)$$

$$v_{i,j} \mid c_{i,j}, \pi_{i,j} \sim \text{Binom}(c_{i,j}, \pi_{i,j}), \quad (3.9)$$

where s is the precision parameter that quantifies allelic imbalance error. A low precision s signifies high unevenness in amplification.

3.2.2 Posterior probability

To determine which droplets are doublets, we are interested in the posterior probability of $\mathbf{z}$ for the given single-cell sequencing data $(\mathbf{C}, \mathbf{V})$, which is defined as

$$P(\mathbf{z} \mid \mathbf{C}, \mathbf{V}) = \frac{P(\mathbf{C}, \mathbf{V} \mid \mathbf{z})P(\mathbf{z})}{P(\mathbf{C}, \mathbf{V})} \propto P(\mathbf{C}, \mathbf{V} \mid \mathbf{z})P(\mathbf{z}). \quad (3.10)$$

In line with current methods [43, 117], we use independence of read counts across mutation loci and droplets and obtain

$$P(\mathbf{C}, \mathbf{V} \mid \mathbf{z}) = \prod_{i=1}^n \prod_{j=1}^m P(c_{i,j}, v_{i,j} \mid z_i). \quad (3.11)$$

We now express $P(c_{i,j}, v_{i,j} \mid z_i)$ in terms of $P(x_{i,j} \mid z_i)$ (described in Section 3.2.1), $P(y_{i,j} \mid x_{i,j}, z_i)$ (described in Section 3.2.1) and $P(c_{i,j}, v_{i,j} \mid y_{i,j})$ (described in Section 3.2.1). Marginal-

izing over $x_{i,j}$ and $y_{i,j}$ yields,

$$P(c_{i,j}, v_{i,j} \mid z_i) = \sum_{x_{i,j} \in \Sigma_i} \sum_{y_{i,j} \in \Theta_i} P(c_{i,j}, v_{i,j}, x_{i,j}, y_{i,j} \mid z_i) \quad (3.12)$$

$$= \sum_{x_{i,j} \in \Sigma_i} \sum_{y_{i,j} \in \Theta_i} P(c_{i,j}, v_{i,j} \mid x_{i,j}, y_{i,j}, z_i) P(x_{i,j}, y_{i,j} \mid z_i) \quad (3.13)$$

$$= \sum_{x_{i,j} \in \Sigma_i} \sum_{y_{i,j} \in \Theta_i} P(c_{i,j}, v_{i,j} \mid y_{i,j}) P(y_{i,j} \mid x_{i,j}, z_i) P(x_{i,j} \mid z_i) \quad (3.14)$$

where

$$\Sigma_i = \begin{cases} \Sigma_{\text{singlet}}, & \text{if } z_i = 0, \\ \Sigma_{\text{doublet}}, & \text{otherwise.} \end{cases} \quad \text{and} \quad \Theta_i = \begin{cases} \Theta_{\text{singlet}}, & \text{if } z_i = 0, \\ \Theta_{\text{doublet}}, & \text{otherwise.} \end{cases} \quad (3.15)$$

3.2.3 DoubletD

Our goal is to find $\mathbf{z} \in \{0, 1\}^n$ such that the likelihood function (Equation (3.10)) is maximized. Substituting the doublet prior from Equation (3.3) in Equation (3.10) and taking log, we get

$$\log P(\mathbf{z} \mid \mathbf{C}, \mathbf{V}) = \sum_{i=1}^n \sum_{j=1}^m \log P(c_{i,j}, v_{i,j} \mid z_i) + \sum_{i=1}^n \log P(z_i) + K \quad (3.16)$$

where K is the constant of proportionality. Since z_i is an indicator variable (i.e. $z_i \in \{0, 1\}$), we linearize the above equation in terms of $\mathbf{z}$ using,

$$\log P(c_{i,j}, v_{i,j} \mid z_i) = \log P(c_{i,j}, v_{i,j} \mid z_i = 0) + z_i \Omega_{i,j} \quad (3.17)$$

where

$$\Omega_{i,j} = \log \left(\frac{P(c_{i,j}, v_{i,j} \mid z_i = 1)}{P(c_{i,j}, v_{i,j} \mid z_i = 0)} \right). \quad (3.18)$$

$$\log P(z_i) = \log P(z_i = 0) + z_i \left(\frac{\log P(z_i = 1)}{\log P(z_i = 0)} \right) \quad (3.19)$$

$$= \log P(z_i = 0) + z_i \log \left(\frac{\delta}{1 - \delta} \right). \quad (3.20)$$

where the last equality uses doublet prior model (Equation (3.3)). Note that, since the read counts $(c_{i,j}, v_{i,j})$ are observed, the matrix $\mathbf{\Omega} = [\Omega_{i,j}] \in \mathbb{R}^{n \times m}$ is constant. Ignoring

the constant of proportionality K , which is independent of $\mathbf{z}$, and using linearization of $\log P(c_{i,j}, v_{i,j} \mid z_i)$ and $\log P(z_i)$ in Equation (3.16), we get the following linear objective function

$$J(\mathbf{z}) = \Phi + \sum_{i=1}^n z_i \left(\sum_{j=1}^m \Omega_{i,j} + \log \left(\frac{\delta}{1-\delta} \right) \right) \quad (3.21)$$

where Φ is a constant defined as follows,

$$\Phi = \sum_{i=1}^n \sum_{j=1}^m \log P(c_{i,j}, v_{i,j} \mid z_i = 0) + \sum_{i=1}^n \log P(z_i = 0). \quad (3.22)$$

Since $J(\mathbf{z})$ is linear, we have the following closed form solution maximizing $J(\mathbf{z})$,

$$z_i = \begin{cases} 1, & \text{if } \sum_{j=1}^m \Omega_{i,j} + \log \left(\frac{\delta}{1-\delta} \right) > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.23)$$

Implementation details. Our resulting method, DOUBLET D, identifies $\mathbf{z} \in \{0, 1\}^n$ given total and variant read counts $(\mathbf{C}, \mathbf{V})$ with maximum posterior probability $P(\mathbf{z} \mid \mathbf{C}, \mathbf{V})$. To do so, DOUBLET D requires input mutation probabilities $\mu_{\text{wt}}, \mu_{\text{het}}$ and μ_{hom} at each locus j used in the genotype model (Section 3.2.1), and the precision parameter s used in the read count model (Section 3.2.1). Appendix B.1.1 describes a data-driven approach to estimate these parameters. Moreover, the doublet prior probability δ can either be taken as input or estimated by maximizing the posterior probability. DOUBLET D is implemented in Python 3, is open source (BSD-3-Clause license), and is available at <https://github.com/elkebir-group/doubletD>.

3.3 RESULTS

We evaluated the performance of doubletD via *in silico* experiments with known ground truth doublets (Section 3.3.1) as well as two real datasets: (i) a two cell line mixture (Section 3.3.2) and (ii) six patients with acute lymphoblastic leukemia [120] (Section 3.3.3).

3.3.1 *In silico* experiments

We aim to answer the following questions: (i) Is doubletD agnostic to the choice of SCDNA-SEQ assay and experimental design? (ii) How robust is doubletD to the presence of

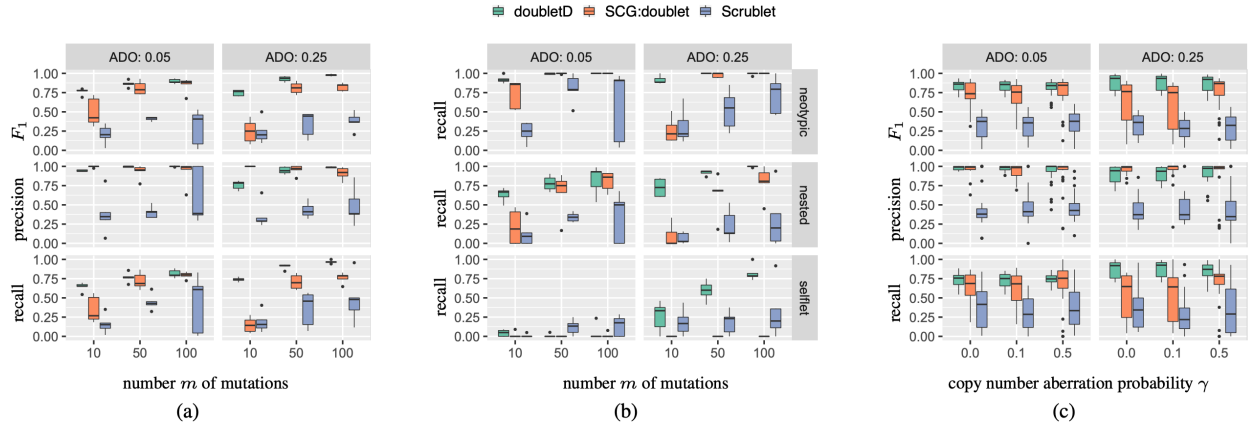


Figure 3.3: **Simulations show that doubletD has high recall and precision in doublet detection, outperforming SCG and SCRUBLET across various experimental regimes and improving performance in downstream genotyping.** (a) F_1 score, precision and recall of doublet detection for the three competing methods (DOUBLETD, SCG:doublet and SCRUBLET) in simulations with varying ADO rate β and number of mutations m in the absence of CNAs ($\gamma = 0$). (b) Recall of the three kind of doublets, *i.e.* neotypic, nested and selflet. (c) F_1 score, precision and recall by method in the presence of CNAs ($\gamma \in \{0, 0.1, 0.5\}$) and varying ADO rate β . All results are for simulations with doublet probability $\delta = 0.2$, mean read depth $c = 50\times$ and precision parameter $s = 15$.

copy number aberrations? (iii) Will the removal of doublets improve downstream analyses? To this end, we simulated SCDNA-SEQ data of 10 genotypes under an evolutionary model that incorporates copy number aberrations (CNAs) and SNVs, varying the number of SNVs $m \in \{10, 50, 100\}$, the doublet probability $\delta \in \{0.1, 0.2, 0.4\}$, the mean sequencing coverage $c \in \{10\times, 50\times, 100\times\}$ and ADO probability $\beta \in \{0.0, 0.05, 0.25\}$. Each combination of simulation parameters was replicated with five different random number generator seeds, amounting to a total of 405 experiments. In each experiment, we simulated 500 *in silico droplet*. We benchmarked our method against SCG [40], a genotyping method for SCDNA-SEQ data whose model optionally incorporates doublet detection, which we refer to as SCG:doublet, and Scrublet [51], a standalone doublet detection method designed for scRNA-seq data. We were not able to benchmark against SiCloneFit [43] and ∞ SCITE [108], which are tree inference methods that also incorporate doublets, due to their prohibitive runtimes when run in doublet mode. Appendix B.2.1 further details the simulation design, evolutionary model and method arguments. In particular, for SCG we performed 25 restarts unless specified otherwise, using the restart with the maximum evidence lower bound (ELBO).

Assay and design agnosticism: We focus on simulations with a mean coverage of $c = 50\times$ and simulated doublet probability of $\delta = 0.2$. We refer to Appendix B.2.1 for other

simulation regimes. While all three methods show increasing F_1 scores (the harmonic mean between precision and recall) with increasing number m of mutations, doubletD achieves the highest F_1 score (median: 0.88) compared to SCG:doublet (median: 0.76) and Scrublet (median: 0.37) (Fig. 3.3a). Specifically, we find that Scrublet has the worst performance in terms of both recall (median: 0.35) and precision (median: 0.38), demonstrating that doublet detection methods developed for scRNA-seq data *cannot* be directly applied to scDNA-SEQ data. While both doubletD and SCG:doublet have equivalently high precision (SCG:doublet median: 0.99 vs doubletD median: 0.98), doubletD has superior recall (median: 0.78) among all methods (median recall of 0.67 for SCG:doublet and 0.35 for Scrublet). Strikingly, SCG:doublet performs poorly in the regime of a small number $m = 10$ of mutations, with a median recall and precision of 0.21 and 1.00, compared to 0.70 and 0.87 for doubletD, respectively. Such small number of mutations do occur in practice — e.g. the ALL data analyzed in Section 3.3.3.

Zooming in on doublet type in Fig. 3.3b, we find that all methods have the highest recall for neotypic doublets (median: 1.00 for doubletD, 1.00 for SCG:doublet and 0.50 for Scrublet), and that the recall increases for both nested and neotypic doublets with increasing number of mutations and increasing ADO. Notably, doubletD has the highest recall for nested doublets (median: 0.85) compared to SCG:doublet (median: 0.57) and Scrublet (median: 0.15). As expected, doubletD and SCG:doublet are unable to detect selflets for ADO rate 0.05 while Scrublet does detect a small proportion of selflets (median: 0.05). However, when ADO rate is 0.25, doubletD has significantly higher recall (median: 0.6) as compared with SCG:doublet (median: 0) and Scrublet (median: 0.2). Note that SCG:doublet is unable to detect selflets due to VAF discretization. Further, both SCG:doublet (IQR: 0.34-0.80) and Scrublet (IQR: 0.13-0.50) show large variance in recall rates as opposed to doubletD (IQR: 0.73-0.92).

Additionally, we find that our method maintains its good performance in simulations when varying coverage and doublet probabilities (Fig. B.2). The lower bound of coverage for the *in silico* experiments was $10\times$. Even at such a low coverage, DOUBLET D maintains its good performance (median precision: 0.83 and median recall 0.78, see Fig. B.2a). It is also important to note that doubletD’s improved performance does not come at the expense of running time (Fig. B.4a, median: 14.9 s vs. 11,000.0 s for SCG:doublet and 4.1 s for Scrublet). Finally, DOUBLET D is robust to the choice of user-specified parameters such as the precision s (Appendix B.2.1, Fig. B.5, Fig. B.6, Fig. B.7 and Fig. B.8). In summary, we find that doubletD is robust to many variations in experimental assays and design, outperforming SCG:doublet and Scrublet.

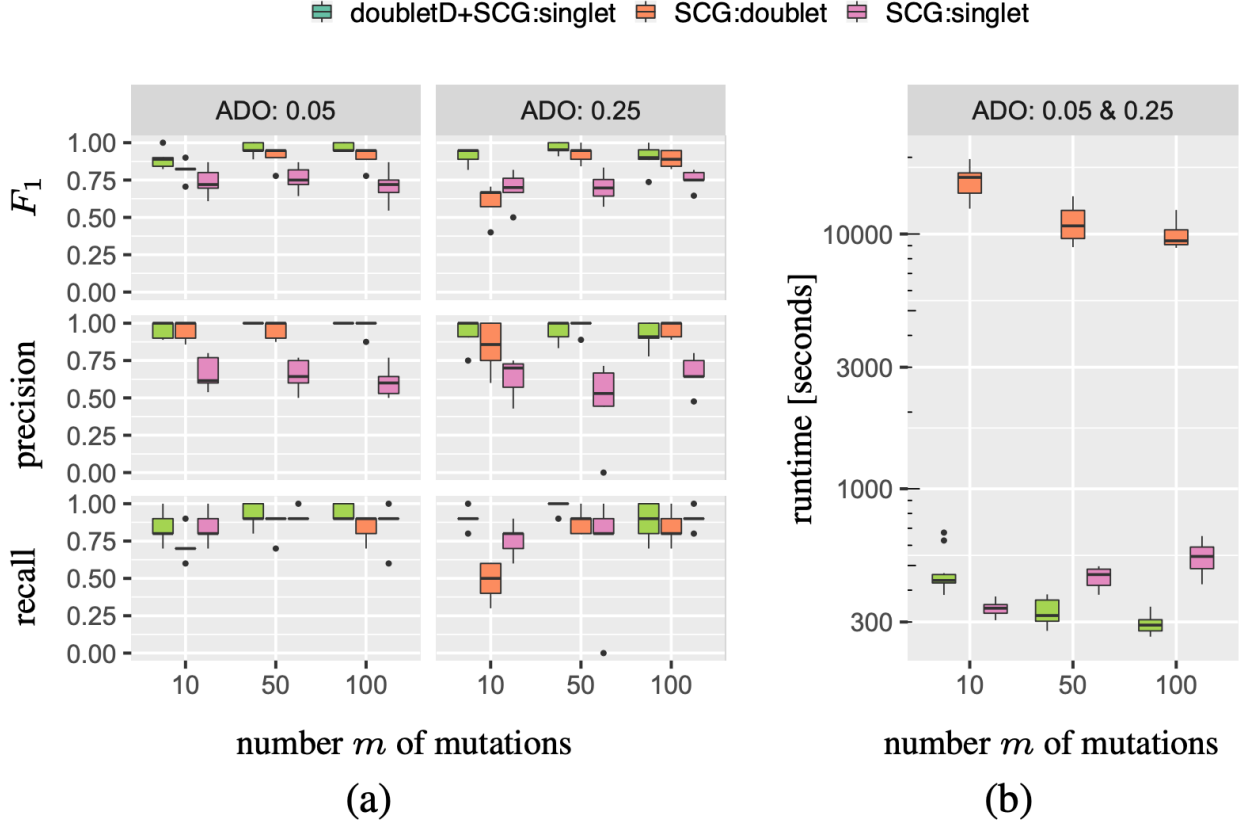


Figure 3.4: **Simulations show that removal of doublets using DOUBLET-D improves downstream genotype calling with reduced runtime.** (a) F_1 score, precision and recall of genotypes for doubletD+SCG:singlet, SCG:doublet and SCG:singlet for varying number of mutation m and ADO rate β and without CNAs ($\gamma = 0$). (b) Running time for genotype calling using doubletD+SCG:singlet, SCG:doublet and SCG:singlet for simulations with varying number of mutations m without CNAs ($\gamma = 0$). All results are for simulations with doublet probability $\delta = 0.2$, mean read depth $c = 50\times$ and precision parameter $s = 15$.

Robustness with respect to CNAs: In order to evaluate the robustness of DOUBLET-D to the presence of copy number aberrations (CNAs), we generated simulations with varying probability of CNAs $\gamma \in \{0, 0.1, 0.5\}$, where $\gamma = 0$ represents simulations with no CNAs. More specifically, for each locus that undergoes a copy number aberration (with probability γ), we introduced a loss with probability $\ell \in \{0.1, 0.5\}$ and a gain otherwise. We ran SCG:doublet with 5 restarts due to increased runtimes compared to the copy-neutral simulations.

Although DOUBLET-D does not explicitly account for CNAs, Fig. 3.3c shows that DOUBLET-D is robust to varying copy number aberration probability γ , outperforming SCG:doublet and SCRUBLET in most regimes. Specifically, DOUBLET-D yields the highest recall (median: 0.79) with good precision (median: 0.98) resulting in the highest F_1 score (median: 0.87)

compared to SCG:doublet (median: 0.80) and SCRUBLET (0.36). While SCG:doublet has the same precision as DOUBLET D (median: 0.98), this comes at the cost of lower recall (median: 0.73) compared to DOUBLET D (median: 0.79).

The robustness of DOUBLET D can be explained by the observation that losses (deletions) introduced by CNAs behave similarly to allelic dropouts, which is a key signal used by DOUBLET D to detect doublets. We demonstrate the vulnerability of DOUBLET D to copy number gains on simulations with highest possible copy number aberration probability $\gamma = 1$ and lowest possible loss probability $\ell = 0$ (Fig. B.3). Note that this kind of extreme presentation of CNAs is not observed in practice and that copy number losses including loss of heterozygosity events are common in cancer [42, 121, 122].

In summary, we find that DOUBLET D is robust to the presence of CNAs and outperforms both SCG:doublet and SCRUBLET in doublet detection.

Improving downstream genotype calling: SCG is a genotyping method for scDNA-SEQ data of tumors that includes doublet detection. It has two modes: in *singlet mode* (SCG:singlet) all droplets are considered singlets, whereas in *doublet mode* (SCG:doublet) genotypes and doublets in the sample are jointly inferred. Here, we assess whether the sequential use of DOUBLET D followed by SCG:singlet (doubletD+SCG:singlet) performs better than SCG:singlet and SCG:doublet. In each of these settings, SCG is run with 25 restarts.

Recall that each of our simulated instances contain 10 genotypes. To assess the performance of the three methods, we compute recall, precision and F_1 score with respect to these ground-truth genotypes, considering a genotype as correctly inferred (i.e. a true positive) if it precisely matches a ground-truth genotype. Thus, if a method infers the exact set of 10 ground-truth genotypes, its recall, precision and F_1 score will be 1.

We find that doubletD+SCG:singlet has the highest F_1 score (median: 0.95) compared to SCG:singlet (median: 0.73) and SCG:doublet (median 0.89) across all experimental regimes (Fig. 3.4a). SCG:singlet has good genotype recall (median: 0.9) but reduced precision (median: 0.64) since it misidentifies doublets as cells with distinct genotypes. SCG:doublet, on the other hand, has better precision (median: 1.0) but filters out rare genotypes misidentified as doublets resulting in reduced recall (median: 0.80). doubletD+SCG:singlet yields the highest recall (median: 0.90) and precision (median: 1.0). In general, SCG:singlet calls more genotypes (median: 14) while SCG:doublet calls fewer genotypes (median: 8.5) compared to the ground truth of 10 genotypes (Fig. B.9). On the other hand, doubletD+SCG:singlet is closer to ground truth with a median of 9.5 distinct genotypes. Furthermore, Fig. 3.4b shows that doubletD+SCG:singlet takes orders of magnitude less time compared to SCG:doublet.

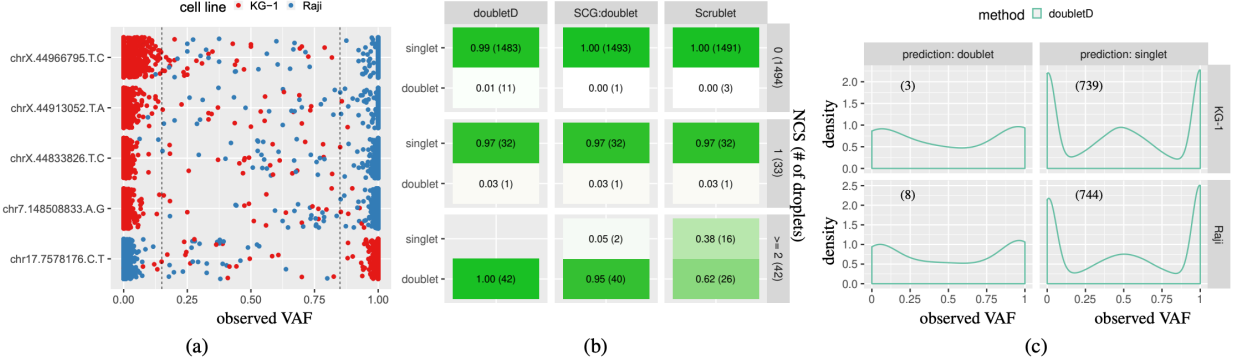


Figure 3.5: **DOUBLET D** successfully recalls all 42 orthogonally validated high confidence neotypic doublets and identifies 11 putative selflets in a two cell line mixture dataset. (a) The VAF for each droplet at each of the five validation loci. Droplets are assigned a neotypic doublet confidence score (NCS), which is the number of validation loci whose VAF was in the range $[0.15, 0.85]$ (dotted lines). (b) The resulting proportion (total) of droplet calls by method (doubletD, SCG:doublet, and Scrublet) by prediction (singlet, doublet) and NCS. (c) The aggregated observed VAF distribution by doubletD prediction and cell line for droplets with NCS = 0. The number of droplets in the aggregate are shown in the parenthesis.

While SCG:singlet takes the least time to run, it also yields the lowest F_1 score (Fig. 3.4a).

In summary, we find that the use of DOUBLET D improves genotype calling of SCG while incurring runtimes comparable to SCG in singlet mode. This suggests that doublet removal using DOUBLET D is a useful pre-processing step for downstream analyses of SCDNA-SEQ data of tumors.

3.3.2 Mixture of two cell lines

We validated DOUBLET D on a dataset of $n = 1,569$ droplets comprised of a 50%-50% mix of KG-1 and Raji cell lines (with $m = 26$ loci) captured by Mission Bio’s Tapestry platform and sequenced by Illumina NextSeq³. Appendix B.2.2 details the data preparation, including the exclusion of 23 cells that had a genotype distinct from the two cell lines. KG-1 had 12 heterozygous (het), 7 wild type and 7 homozygous loci, while Raji had 11 heterozygous, 7 wild type and 8 homozygous loci (Fig. B.10). The mean sequencing coverage c was $110\times$. Following the procedure outlined in Appendix B.1.1, we fit beta-binomial precision $s = 10.5$, $\alpha_{fp} = 0.015$, $\alpha_{fn} = 0.0073$ and locus-specific mutation probabilities μ to the observed variant $\mathbf{V}$ and total read counts $\mathbf{C}$. We used the experimental ADO rate ($\beta = 0.06$) previously estimated by Morita et al. [30] on a large patient cohort using Mission

³<https://portal.missionbio.com/datasets/KG-1-Raji-50-50-Myeloid>

Bio’s Tapestri platform.

There are two unique characteristics of this dataset that permit identification of neotypic doublets for orthogonal validation: (i) the droplets are easily clustered into two clones by the cell line of origin (Fig. B.10) and (ii) the droplets are comprised of distinct cell lines with distinct evolutionary histories. These characteristics are uncommon in regular datasets where the number of clones and associated genotypes is unknown *a priori* and droplets originate from a single tumor whose clones have a shared evolutionary history. As such, we conclude that doublets will be either neotypic (one cell from each cell line), or selflets (two cells from one cell line).

Using the property that the two cell lines have independent origins and relaxing Mission Bio’s standard filtering criteria, we identified an additional set of 5 *validation loci* with distinct wild type/homozygous states among the two cell lines, i.e. each validation locus has state wt (hom) in one cell line and hom (wt) in the other (Fig. 3.5a). Recall that a singlet i will have an observed VAF $v_{i,j}/c_{i,j}$ of approximately 1 if locus j is homozygous and VAF 0 if locus j is wild type. As such, any droplets with observed VAF not close to either 0 or 1 (Fig. 3.5a) indicate that the droplet may be a neotypic doublet comprised of a cell from each cell line. We therefore assign a *neotypic doublet confidence score* (NCS) to each droplet, counting the number of validation loci with VAF between 0.15 and 0.85. This approach yielded 1,494 droplets with $\text{NCS} = 0$, 33 droplets with $\text{NCS} = 1$ and 42 droplets with $\text{NCS} \geq 2$. Note that the NCS is specifically designed to express confidence that a doublet is neotypic but does not capture selflets. Fig. B.10 shows a comparison of the observed variant allele frequency (VAF) of droplets categorized by cell line droplets with a neotypic doublet confidence score ($\text{NCS} \geq 2$).

We ran DOUBLET-D, SCG:doublet (with 5 restarts), and SCRUBLET. Since we did not know the true doublet probability δ , we used the maximum likelihood criterion to establish the estimate the doublet probability for DOUBLET-D as $\delta = 0.05$ (Fig. B.11a). However, we provided SCG and SCRUBLET with the doublet probability $\delta = 0.09$ as estimated by Mission Bio in similar cell line experiments [123]. For each method and NCS, we calculated the proportion of predicted singlets and doublets (Fig. 3.5b). DOUBLET-D identified the most droplets as doublets (54), followed by SCG:doublet (42) and Scrublet 30. DOUBLET-D predicted 100% of doublets with $\text{NCS} \geq 2$ whereas SCG:doublet identifies 95.2% of these droplets with similarly high NCS. Scrublet is the worst performing, identifying only 61.9% of such droplets (Fig.3.5b). In terms of running time, SCG:doublet took 16,259.7 s, DOUBLET-D took 24.1 s and SCRUBLET took 2.4 s.

All three methods designated the same droplet at $\text{NCS} = 1$ as a doublet. This suggests that for the remaining 32 droplets at $\text{NCS} = 1$ the observed VAF in $[0.15, 0.85]$ at one of these

5 validation loci is likely attributable to amplification and sequencing error. The one doublet identified by all methods does appear to be neotypic as evidenced by an observed VAF of 0.39 for the validation locus on chromosome 17, which is far from the cut off criterion of 0.15 and is hard to explain by other errors. Furthermore, the VAF distribution across the 26 inference loci for this droplet has a peak at 0.25 and is strikingly different from the distribution of the other Raji droplets with NCS equal to 1 (Fig. B.11b). Lastly, DOUBLET-D identifies 11 (proportion: 0.007) putative selflets at NCS = 0, 3 of which are KG-1 and 8 are Raji. SCG calls 1, which was also called by DOUBLET-D, and SCRUBLET calls 3 such droplets with only one called by DOUBLET-D. Corroborating this, we note a visual difference in the aggregated VAF distribution across the inference 26 loci between DOUBLET-D predicted singlets and doublets with NCS = 0 (Fig. 3.5c). A Venn diagram of the droplets with different NCS score that were predicted as doublets by the three competing methods is shown in Fig. B.12.

In summary, DOUBLET-D is able to recall all orthogonally validated high confidence neotypic doublets (with NCS ≥ 2) as well as successfully distinguish the VAF signal of neotypic doublets from sequencing-related error. In addition, we suspect that DOUBLET-D is able to recall a small number of selflets even in the presence of low ADO rates ($\beta = 0.05$).

3.3.3 Phylogeny inference of an acute lymphoblastic leukemia patient

As discussed in Section 3.1, while nested doublets and selflets do not yield new genotypes, neotypic doublets can be mistaken as an additional clone with a unique genotype [104]. In the extreme case of a phylogeny with only two branches, neotypic doublets that correspond to the two leaves of this tree will include all mutations. Consequently, phylogeny inference under the infinite sites assumption will yield a linear phylogeny. Here, we investigate the impact of doublets on phylogeny inference for a patient (Patient 1) in an acute lymphoblastic leukemia cohort previously suspected to contain doublet droplets [120] — we refer to Table B.2 for DOUBLET-D results of the other patients.

Gawad et al. [120] sequenced 243 droplets and identified 20 mutations for Patient 1. We analyzed this patient using PhISCS-B [?], which is a phylogeny inference method that seeks to identify a tree constrained by the infinite sites assumption. Since it does not account for doublets, PhISCS-B requires doublets be removed in a pre-processing step. While SCG:doublet was unable to identify any doublets, DOUBLET-D identified 50 doublets for this patient. Fig. B.13 corroborates these doublets, showing distinct VAF distributions between singlets and doublets for an orthogonal set of holdout loci. We ran PhISCS-B in single-cell data mode on the complete set of droplets (including doublets) as well as the set of droplets without doublets (details in Appendix B.2.3). Fig. 3.6 shows that doublet removal in this pa-

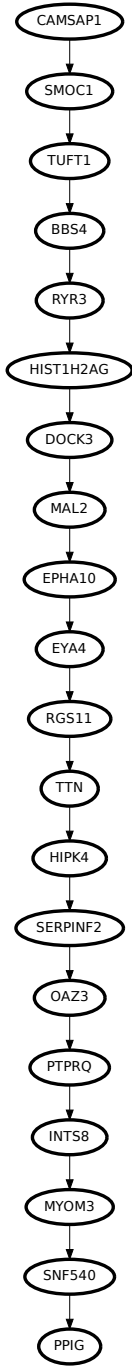
tient results in a branching phylogeny with a higher mean likelihood ($-1157.39/193 = -6.00$) compared to a linear phylogeny ($-2806.49/243 = -11.55$) on the complete set of droplets. Furthermore, the branching pattern observed in the inferred phylogeny after doublet removal is in agreement with several other trees published for Patient 1, with identical grouping of the mutations across the two branches [108, 120, 124].

Thus, phylogeny inference is an additional example of a downstream analysis where the inclusion of doublets may yield incorrect conclusions.

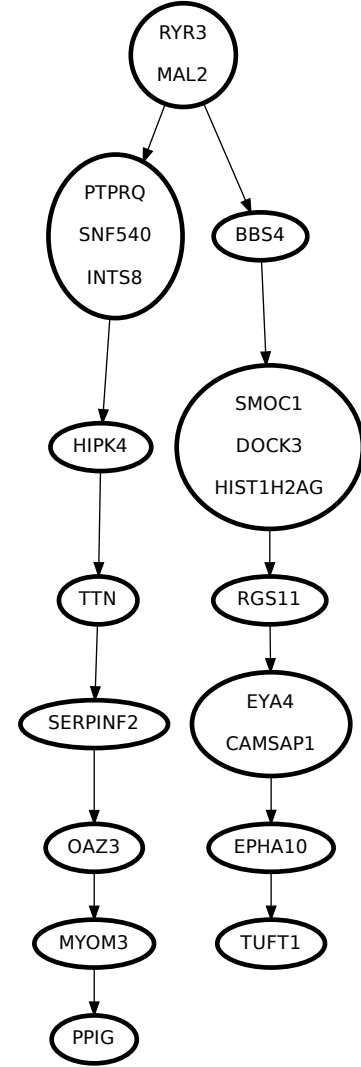
3.4 DISCUSSION

In this work we introduced doubletD, the first standalone method for detecting doublets in single-cell DNA sequencing (SCDNA-SEQ) data with medium to high-coverage ($\geq 5\times$) suitable for single-nucleotide variants. Our method operates directly on variant and total read counts of mutation loci. Underlying our method is the observation that doublets in SCDNA-SEQ data have a characteristic VAF distribution. An additional signal that we exploit is the shift in VAFs due to allelic dropout. This unique approach enables DOUBLET D to capitalize on a major downside of single-cell sequencing in order to identify selflets and nested doublets, that are notoriously hard to detect by current methods. doubletD utilizes a probabilistic model that specifically accounts for allelic imbalance and dropout during whole-genome amplification in SCDNA-SEQ as well as sequencing errors. We introduced a closed-form solution for the inference problem. We demonstrated that our method outperforms current methods for downstream analysis of SCDNA-SEQ data that jointly infer doublets and genotypes [40] as well as standalone approaches for doublet detection in scRNA-seq data [51]. Moreover, we showed that removing doublets using DOUBLET D improves the accuracy and efficiency of downstream analyses such as genotype calling and phylogeny inference.

There are several opportunities for future work. First, while this paper focused on cancer, doubletD can be applied to normal samples as well using heterozygous germline SNPs. Moreover, the same characteristic signal used by our method to detect doublets can be used to detect cells that have undergone whole-genome duplication or are in S-phase with actively replicating DNA. Second, our approach can be extended to support low ($0.1 - 0.5\times$) to ultra-low ($< 0.05\times$) coverage SCDNA-SEQ samples, suitable for copy-number aberrations, by pooling heterozygous germline SNPs located within haplotype blocks. Third, our current formulation assumes that normal cells are diploid. As noted in our simulations, performance slightly decreased in the presence of copy-number aberrations. We plan to extend our probabilistic model to account for copy number. Finally, we envision that DOUBLET D will improve



(a) Including doublets



(b) Excluding doublets

Figure 3.6: **Doublets lead to incorrect phylogeny inference in acute lymphoblastic leukemia patient 1.** (a) PhISCS-B returns an linear phylogeny with mean log likelihood of $-2806.49/243 = -11.55$ if the 50 doublets detected by doubletD are retained. (b) PhISCS-B returns a branching phylogeny with higher mean likelihood of $-1157.39/193 = -6.00$.

downstream analysis of current and future methods, making doublet detection and removal a standard practice in scDNA-SEQ analysis.

CHAPTER 4: PHYOLIN: DISTINGUISHING LINEAR AND BRANCHED EVOLUTION GIVEN SINGLE-CELL DNA SEQUENCING DATA OF TUMORS

4.1 INTRODUCTION

By performing bulk and/or single-cell DNA sequencing of a heterogeneous tumor biopsy, researchers and clinicians may infer reasonable models of this evolutionary process for important downstream analysis and clinical decision-making. Due to trade-offs between the two data types, techniques for phylogeny inference have been developed for either bulk sequencing or single-cell data individually [35, 38, 41, 42, 127] as well as combined for joint inference [39, 43, 124]. Bulk sequencing data is less costly than single-cell data but results in a set of plausible phylogenies making it difficult to uniquely determine the true evolutionary history of a tumor [34, 37]. Conversely, single-cell data allows more precise reconstructions of a tumor’s evolutionary history but is subject to high rates of sequencing errors and is more expensive than bulk sequencing. In particular, single-cell sequencing has a high false negative rate, as much as 40% [94], implying that actual mutations present in a cell might not be indicated correctly in the resulting data. Doublets, where multiple cells are simultaneously sequenced as a single cell, are also a unique challenge of single-cell data. Less problematic are false positives, which indicate the presence of a mutation that is not present in a cell. While these rates are low (ranging from 0.0001 to 0.001 [94]) in comparison to false negative rates, it is important to account for false positives when inferring the evolutionary history of tumor.

One important open question is whether certain types or subtypes of cancers follow specific evolutionary patterns. Since tumors are typically biopsied at only a single point in time for reasons related to patient care, there does not yet exist sufficient longitudinal data to fully answer this question. However, it is believed that there are four high-level categories of tumor evolution: linear evolution, branched evolution, neutral evolution and punctuated evolution [53]. Linear and branched evolution are the focus of this work as neutral and punctuated evolution are special cases of these two high level categories. Linear evolution results when subsequent driver mutations develop a strong selective advantage and outcompete other

⁴This chapter contains identical material with an adapted introduction previously published in “Distinguishing linear and branched evolution given single-cell DNA sequencing data of tumors” [125], which is an extended version of the paper from WABI 2020, “Phyolin: Identifying a linear perfect phylogeny in single-cell DNA sequencing data of tumors by the same title” [126]. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with M. El-Kebir, MEK conceived the project; LLW, MEK established theory; LLW performed experiments; LLW and MEK contributed to the writing of this chapter.

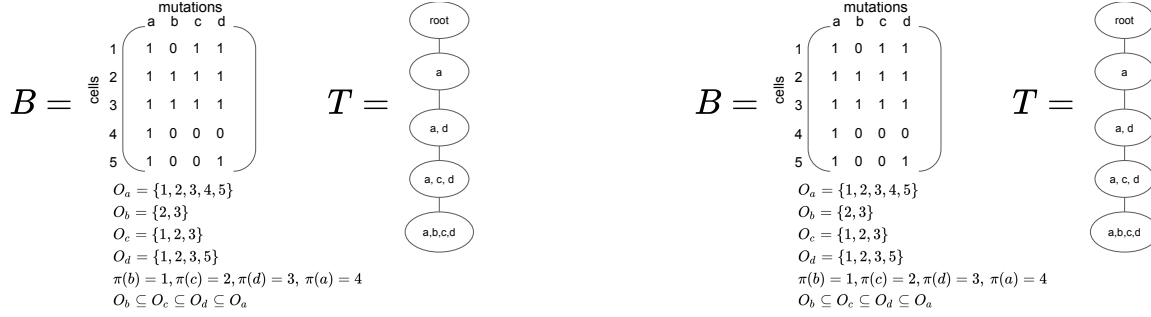


Figure 4.1: a) A graphical depiction of PHYOLIN identifying a linear perfect phylogeny in single-cell DNA sequencing data when given a binary matrix B and a false negative rate β^* . b) An example of error free single-cell data that represents a linear perfect phylogeny and the equivalent clonal tree representation.

clones during a clonal expansion [53]. By contrast, in branched evolution, a clone can diverge into separate lineages resulting in distinct branches and a tree-like model of evolution.

A useful first step in gaining insight into the evolutionary patterns of different cancer types is to compare the likelihood of the observed single-cell data under these two alternative hypotheses, linear and branched. Suppose we are given single-cell data in the form of a matrix where each row in the data is a cell and each column is a single-nucleotide variant (SNV), hereafter referred to as mutation. The entries in the data would then be either 1 or 0 indicating the presence or absence of a mutation in a particular cell. Suppose also that we assume a model of linear evolution and are given a false negative rate for the technology under which the single-cell sequencing was performed.

We could then determine the minimum number of changes from 0 to 1, indicating the entry was a false negative, such that the data is representative of a linear perfect phylogeny. This would then allow us to estimate the likelihood of the data under a hypothesis of linear evolution with false negative rate distribution in accordance with the sequencing technology. Under an alternative model of branched evolution, the number of flips required to obtain a linear perfect phylogeny and the associated false negative rate would follow a different distribution since many more flips would be required to not only correct actual false negatives but also to remove the implied branching.

Azer et al. [128] utilized a deep learning approach to decide if single-cell data indicates whether a tumor followed a linear or branched evolutionary process. Although their method is fast at prediction time and performs well on simulated data, it has not yet been proved whether the problem of identifying the minimum number of flips to obtain a linear perfect phylogeny is NP-hard. Additionally, the neural networks are trained on inputs of a fixed size. While padding could be used in predicting an input smaller than the fixed size [128],

a new network would have to be trained if the input size is larger than the trained network. This drawback significantly reduces the speed advantage of such an approach as training of neural networks is a time consuming process.

Here, we prove that the problem of determining the minimum number of flips from 0 to 1 in single-cell data in order for the data to represent a linear phylogeny under the infinite sites model is NP-hard. We develop a method called PHYOLIN that makes use of constraint programming to find the minimum number of flips required to represent a linear perfect phylogeny. The outputted number of flips from PHYOLIN is then used to compute the strength of the evidence of the two alternatives of the pattern of evolution (Fig. 4.1a). We evaluate the performance of PHYOLIN on both simulated and real datasets, demonstrating that our method is an accurate and reasonably fast method for classifying an evolutionary trajectory as linear or branched.

4.2 MATERIALS AND METHODS

Let n be the number of single cells sequenced and m be the number of unique mutations present in the n cells. In the following, after introducing the LINEAR PERFECT PHYLOGENY FLIPPING problem, we describes two alternatives models of evolution.

4.2.1 Problem statement

Under the infinite sites assumption (ISA), where each mutation i is gained exactly once and never subsequently lost, each sequenced cell corresponds to a leaf and we may infer a two-state perfect phylogeny using a polynomial time algorithm [129] where the binary character states encode the presence of mutation j in a cell i . We may equivalently represent a perfect phylogeny T as a binary matrix $B \in \{0, 1\}^{n \times m}$ where $b_{ij} = 1$ if cell i harbors mutation j and 0 otherwise. We provide the following definition [130] for convenience.

Definition 4.1. Given an n by m binary-character matrix B for n cells and m mutations, a *perfect phylogeny for B* is a rooted tree T with exactly n leaves provided that:

1. Each of the n cells labels exactly one leaf of T .
2. Each of the m mutations labels exactly one edge of T .
3. For any cell p , the mutations that label the edges along the unique path from the corresponding leaf to the root specify all of the mutations of p whose state is one.

Next, we formalize the notation of a set of cells that contain a mutation as the *one state*.

Definition 4.2. The *one state* O_j of mutation j is the set of single cells i where $b_{ij} = 1$.

A perfect phylogeny T either depicts linear evolution or branched evolution. Intuitively, a matrix B represents linear evolution if there exists a total order of the set of *one states* with respect to the subset relation. Otherwise, we say perfect phylogeny T represents branched evolution. Also, we note that perfect phylogeny T is not necessarily bifurcating.

Utilizing the collection of *one states* for all m mutations, we determine if a given binary matrix B represents a linear perfect phylogeny as follows (Fig. 4.1b).

Definition 4.3. A binary matrix $B \in \{0, 1\}^{n \times m}$ is a *linear perfect phylogeny* if there exists a permutation $\pi : [m] \rightarrow [m]$ such that $O_{\pi(1)} \subseteq \dots \subseteq O_{\pi(m)}$.

However, single-cell sequencing is not error free and matrix B can fail to represent a linear perfect phylogeny even when it is representative of the true evolutionary process. False negatives, where a mutation that is present is not indicated as such, are particularly problematic with rates of up to 0.4 [94]. False positives, where absent mutations are indicated as present, are less of an issue in practice with rates less than 0.0005 for typically used whole-genome amplification strategies [94]. If the false positive error rate is known, we may estimate the expected number z of false positives in the observed data.

Given that false negatives are particularly prevalent, we would like to know how many false negatives would have had to occur in order for the inferred perfect phylogeny under the ISA to have a linear structure? This leads to the following problem statement.

Problem 4.1 (LINEAR PERFECT PHYLOGENY FLIPPING (LPPF)). Given a matrix $B \in \{0, 1\}^{n \times m}$ and an integer z , find the minimum number of bit flips from 0 to 1 in B yielding a linear perfect phylogeny and the number of bit flips from 1 to 0 is at most z .

As we describe in Methods, upon obtaining the solution to the LPPF, we utilize the total number of flips to distinguish whether the observed single-cell binary matrix B better supports linear or branched evolution.

4.2.2 Complexity

Following [131], we prove that LPPF is NP-hard by a reduction from the chain graph insertion problem, a known NP-complete problem [132].

Theorem 4.1. LPPF is NP-hard.

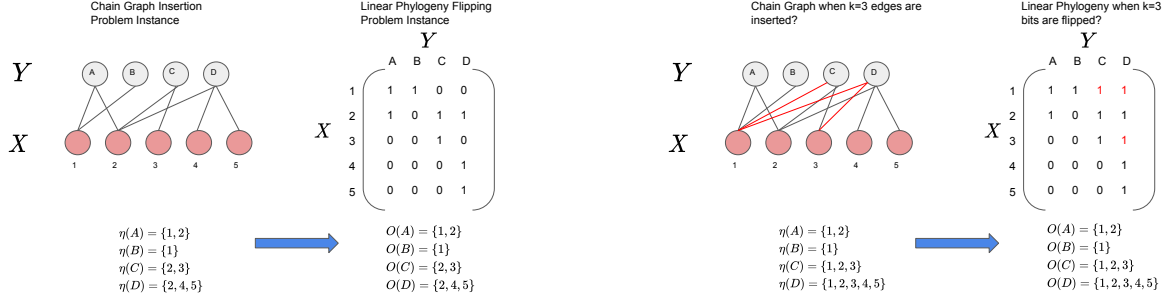


Figure 4.2: **Chain graph insertion problem reduction.** a) Polynomial time reduction of the CGI to LPPF. b) An equivalent solution of the CGI and LPPF when $k = 3$.

Proof. We prove that LPPF is NP-hard by considering a decision version k -LPPF asking whether there exist k bit flips in input matrix B from 0 to 1 yielding a linear perfect phylogeny when the number z of the reverse flips (1 to 0) is 0. We claim that k -LPPF is NP-complete by reduction from the chain graph insertion problem.

We begin by stating the definition of a chain graph and introduce the chain graph insertion problem.

Definition 4.4. A bipartite graph $G = (X \cup Y, E)$ is a *chain graph* if there exists a permutation $\phi : \{1, \dots, |Y|\} \rightarrow Y$ such that $\eta(\phi(1)) \subseteq \dots \subseteq \eta(\phi(|Y|))$ where $\eta(v) = \{w \in X \mid (v, w) \in E\}$ is the set of adjacent nodes of v .

Problem 4.2 (CHAIN GRAPH INSERTION (CGI) [132]). Given a bipartite graph $G = (X \cup Y, E)$ and integer k , does there exist a chain graph $G' = (X \cup Y, E')$ such that $E \subseteq E'$ and $|E| + k = |E'|$?

k -LPPF $\in$ NP because given a certificate (set of k flips from 0 to 1) to k -LPPF, we could order the columns by increasing cardinality of the resulting one state sets and then check if that permutation satisfies the definition of a linear perfect phylogeny. We will now show that $\text{CGI} \leq_p k\text{-LPPF}$.

Starting from an instance $(G = (X \cup Y, E), k)$ of CGI, we construct an instance B of k -LPPF in the following manner. Binary matrix B has $|X|$ rows and $|Y|$ columns, and its entries are directly obtained from the edge set E of G : For each $v \in Y$, we set $b_{iv} = 1$ if i is a neighbor of v for all $i \in X$ and let $b_{iv} = 0$ otherwise. This can be done in polynomial time. Fig. 4.2a demonstrates the polynomial time reduction. We claim that k edge insertions suffice to obtain a chain graph from G if and only if B represents a linear perfect phylogeny when k bits are flipped from 0 to 1 and no bits are flipped from 1 to 0 (i.e. $z = 0$).

($\implies$) Suppose $(G, k) \in \text{CGI}$. Then there exists an edge set $D \subseteq \{(u, v) \mid u \in X, v \in Y\} \setminus E$

such that $|D| = k$ and $H = (X \cup Y, E \cup D)$ is a chain graph. Then by definition of a chain graph, there exists a permutation $\phi : \{1, \dots, |Y|\} \rightarrow Y$ such that $\eta(\phi(1)) \subseteq \dots \subseteq \eta(\phi(|Y|))$. It is easy to see that by construction, $\eta(v) = O_v$, for all $v \in Y$. Since ϕ exists, a permutation π of the one states also exists. Therefore, B represents a linear perfect phylogeny after flipping the 0-bits encoded in set D .

($\Leftarrow$) Suppose $(B, k) \in k\text{-LPPF}$. Then there exists a set F of positions (i, j) where $b_{ij} = 0$ to $b_{ij} = 1$ such that $|F| = k$. Let B^* be the resulting matrix after each flip at position $(i, j) \in F$ is made. Then B^* represents a linear perfect phylogeny and there exists a permutation $\pi : \{1, \dots, m\} \rightarrow [m]$ such that $O_{\pi(1)} \dots \subseteq O_{\pi(m)}$. Using the equivalence between one-states and neighbors, $H = (X \cup Y, E)$ can be constructed from B^* in the following manner. First, create the set $X = [n]$ from the rows of B^* and the set $Y = [m]$ from the columns of B^* . Then, create the set E of edges as $\{(x, y) \in X \times Y \mid b_{xy}^* = 1\}$. By construction, $H = (X \cup Y, E)$ is a chain graph. QED.

4.2.3 PHYOLIN

The following section describes PHYOLIN, which solves the LPPF problem and computes the Bayes factor for distinguishing linear and branched evolution. We implemented PHYOLIN in C++ utilizing IBM ILOG CP OPTIMIZER⁵. PHYOLIN is publicly available at <https://github.com/elkebir-group/phyolin>.

To solve the LPPF, we formulate a constraint optimization problem (COP) [133]. A COP is a constraint satisfaction problem (CSP) with an objective function that specifies which feasible solutions are preferred based on an optimization criteria. A CSP is defined by a tuple $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where $\mathcal{X} = \{x_1, \dots, x_n\}$ is the set of decision variables, $\mathcal{D} = \{d_1, \dots, d_n\}$ is the set of domains for $\mathcal{X}$ and $\mathcal{C}$ is a set of constraints that must be satisfied. A solution $a \in \mathcal{A}(\mathcal{X}, \mathcal{D}, \mathcal{C})$ to a CSP is an assignment of values $\{x_1 \mapsto v_1, \dots, x_n \mapsto v_n\}$ such that $v_i \in d_i$ for all $i \in [n]$ and all constraints C are satisfied. To facilitate an objective function $f : \mathcal{A}(\mathcal{X}, \mathcal{D}, \mathcal{C}) \rightarrow \mathbb{R}$, an initial assignment $\hat{a} \in \mathcal{A}(\mathcal{X}, \mathcal{D}, \mathcal{C})$ is found. Then, a preference constraint is added to $\mathcal{C}$, such that $f(a) \leq f(\hat{a})$ for a minimization problem or $f(a) \geq f(\hat{a})$ for a maximization problem. The search is continued and the preference constraint updated each time a feasible assignment $\hat{a}$ is found until no more feasible assignments exist. When this occurs, the assignment $\hat{a}$ is returned and $f(\hat{a})$ is the objective value. We note that in problems where multiple optimal solutions exist, it is possible to return all such valid solutions. But even though multiple solutions may exist, our focus is on assessing the plausibility of the

⁵<https://www.ibm.com/analytics/cplex-cp-optimizer>

null hypothesis of linear evolution. Therefore, it is sufficient to consider any optimal solution even if the respective assignments yield different linear perfect phylogenies.

First, we describe the set $\mathcal{X}$ of decision variables and the associated domains $\mathcal{D}$ used in the formulation. The set $\mathcal{X}$ contains the variables $\mathbf{x}$ and c). Intuitively, the values taken by $\mathbf{x}$ represent a binary matrix B' used to represent a linear perfect phylogeny after flipping. More formally, given a set n of cells and a set m of mutations, let $x_{ij} = 1$ if cell i has mutation j in the linear perfect phylogeny B' after flipping and 0 otherwise for each cell $i \in [n]$, and mutation $j \in [m]$. Then, $\mathcal{D}(x_{ij}) = \{0, 1\}$, for all $i \in [n], j \in [m]$. The variables c), are used to define a permutation of the columns in B' , such that after flipping is completed, B' will adhere to the definition of a linear perfect phylogeny. Recall that in order to represent a linear perfect phylogeny, there must exist permutation $\pi : [m] \rightarrow [m]$ such that $O_{\pi(1)} \subseteq O_{\pi(2)} \subseteq \dots \subseteq O_{\pi(m)}$. Let $c_j = \pi(j)$ for all $j \in [m]$. Then $\mathcal{D}(c_j) = [m]$ for all $j \in [m]$.

Since, our goal is to find the linear perfect phylogeny that requires as few flips as possible, that is minimizing the number of false negatives we infer, we define an objective function that minimizes the number of flips from 0 to 1.

$$\min \sum_{i=1}^n \sum_{j=1}^m \mathbb{1}(b_{ij} = 0, x_{ij} = 1). \quad (4.1)$$

The set $\mathcal{C}$ ensures that the outputted binary matrix B' meets the conditions of representing a linear perfect phylogeny and also that the number of false positives, or flips from 1 to 0, is bounded above by z^* . The set $\mathcal{C}$ consists of the following three constraints.

$$\text{ALLDIFFERENT}(c), \quad (4.2)$$

$$\sum_{i=1}^n \sum_{j=1}^m \mathbb{1}(b_{ij} = 1, x_{ij} = 0) \leq z^* \quad (4.3)$$

$$(c_k < c_j) \Rightarrow (x_{ij} \leq x_{ik}) \quad \forall k, j \in [m], \forall i \in [n]. \quad (4.4)$$

Equation (4.2) is a global constraint that ensures that every mutation is assigned a unique ordering in the permutation. Equation (4.3) ensures that the number of false positives or flips from 1 to 0 is bounded above by the number of allowable false positives z^* , as determined by a given false positive rate α . Finally, Equation (4.4) ensures the defining property of a linear perfect phylogeny is met by ensuring that if $\pi(k)$ is less than $\pi(j)$ then it must hold that $O_k \subseteq O_j$ for all $k, j \in [m]$.

Two alternative hypotheses for the pattern of evolution. We have two competing hypotheses to explain the observed data B : (i) linear evolution (H_{linear}) and (ii) branched evolution (H_{branched}). The solution to the LPPF is the total number y of flips from 0 to 1 in order to represent linear evolution. In addition, we obtain a matrix B' containing the resultant bits after flipping. Let N be the total number of ones in B' and $\hat{\beta} = y/N$ represent the fraction of false negatives in the solution to the LPPF. We then hypothesize y was drawn from one of two different beta-binomial distributions.

Under the null hypothesis H_{linear} of linear evolution, we have

$$p_{\text{linear}} \sim \text{beta}(\mu_{\text{linear}}, s_{\text{linear}}), \quad (4.5)$$

$$y \sim \text{binomial}(N, p_{\text{linear}}), \quad (4.6)$$

where the mean μ_{linear} equals the expected false negative rate β^* and s_{linear} is the beta-binomial precision parameter (controlling dispersion) of the used sequencing technology.

In the worst case, every 0 can be flipped to a 1. This results in a binary matrix with all values equal to 1, suggesting a linear perfect phylogeny with a single clone harboring all of the mutations. The number of flips required to achieve such a solution may be implausible under a model of linear evolution, which seeks to characterize the true distribution of false negatives for the sequencing. Thus, we consider an alternative hypothesis that characterizes the distribution of flips when a branched phylogeny is forced to be linear.

Under the alternative hypothesis H_{branched} of branched evolution, H_{ϕ} , we have

$$p_{\text{branched}} \sim \text{beta}(\mu_{\text{branched}}, s_{\text{branched}}), \quad (4.7)$$

$$y \sim \text{binomial}(N, p_{\text{branched}}), \quad (4.8)$$

where μ_{branched} and s_{branched} parameterize a distribution that results from converting data generated under branched evolution into one that represents a linear perfect phylogeny. In other words, we expect the estimated false negative rate $\hat{\beta}_{\phi} = y/N$ to follow a different distribution from that of the false negative rate β^* of the sequencing technology since the number of flips for branched evolution will be greater for data generated under linear evolution.

While μ_{linear} and s_{linear} are typically known, parameters μ_{branched} and s_{branched} are more challenging to estimate. One approach is to generate simulated data under branched evolution and the given sequencing profile and utilize $\hat{\beta}_{\phi}$ to fit this distribution. As this is time consuming in practice, we offer two heuristics. First, given some hard threshold β^* , which could be based on knowledge of the system estimated false negative rate or conservatively set at 0.4 [94], we reject H_{linear} that the phylogeny is linear whenever $\hat{\beta} > \beta^*$. Unlike the Bayes

factor, a hard threshold does not provide any evidence in support of H_{linear} , only allowing us to conclude the pattern of evolution is branched. Second, as a more robust alternative, one can establish a hypothetical threshold beta-binomial distribution with parameters μ_{branched} and s_{branched} representing an implausible distribution for the false negative rate of the sequencing technology.

Bayes factor To determine the hypothesis that best explains the observed data B , we utilize the Bayes factor. The Bayes factor

$$K = \frac{P(B \mid H_{\text{linear}})}{P(B \mid H_{\text{branched}})} \quad (4.9)$$

quantifies the likelihood ratio of the two computing hypotheses (H_{linear} and H_{branched}) for the evolutionary pattern of the observed binary matrix B . A Bayes factor $K > 1$ is evidence in support of H_{linear} and $K < 1$ is evidence in support of H_{branched} . Furthermore, Bayes factor K expresses how strongly the observed data supports one of the two alternative hypotheses.

Given two shape parameters θ_1 and θ_2 , the likelihood of the beta-binomial distribution modeling each of our two hypotheses is defined as

$$f(y \mid, N\theta_1, \theta_2) = \binom{N}{y} \frac{\mathcal{B}(\theta_1 + y, N - y + \theta_2)}{\mathcal{B}(\theta_1, \theta_2)}, \quad (4.10)$$

where $\mathcal{B}$ is the beta function.

We parameterize the statistical distribution of our hypotheses in terms of θ_1 and θ_2 via the conversion

$$\theta_{i,1} = \mu_i s_i, \quad (4.11)$$

$$\theta_{i,2} = s_i - \theta_{i,1}, \quad (4.12)$$

for $i \in \{\text{linear}, \text{branched}\}$. Finally, substituting Equation (4.10) into Equation (4.9) for both hypotheses H_{linear} and H_{branched} yields Bayes factor K equal to

$$\frac{\mathcal{B}(\theta_{\text{linear},1} + y, \theta_{\text{linear},2} + N - y) \mathcal{B}(\theta_{\text{branched},1}, \theta_{\text{branched},2})}{\mathcal{B}(\theta_{\text{branched},1} + y, \theta_{\text{branched},2} + N - y) \mathcal{B}(\theta_{\text{linear},1}, \theta_{\text{linear},2})}. \quad (4.13)$$

4.3 RESULTS

In order to evaluate PHYOLIN, we perform *in silico* experiments as well as run PHYOLIN on real data. First, we seek to evaluate the performance of PHYOLIN when the simulated

data closely approximates a recently published high throughput single-cell DNA sequencing study of an acute myeloid leukemia (AML) cohort [30]. Second, we establish the robustness of PHYOLIN to false positives and doublets. Third, we describe the application of PHYOLIN to patients with childhood acute lymphoblastic leukemia [120]. Finally, we apply PHYOLIN to the AML cohort [30] that inspired the initial set of simulations. All experiments were conducted on a server with Intel Xeon Gold 5120 dual CPUs with 14 cores each at 2.20GHz and 512 GB RAM.

4.3.1 Simulations approximating an acute myeloid leukemia cohort

In a recent study, Morita et al. [30] performed DNA single-cell sequencing on a cohort of 123 patients with acute myeloid leukemia (AML) and inferred the evolutionary tree of each patient using SCITE [38]. Utilizing high-throughput sequencing resulted in a median of 7,584 cells sequenced per patient [30]. AML is a cancer type where both linear and branching evolutionary patterns are suspected [30]. As a result, the set of trees published in Ref. [30] were a mix of linear and branched patterns. We design a simulation study to approximate a subset of patients in this cohort but where the ground truth evolutionary pattern is known. We use the published trees and clonal prevalences in conjunction with estimated per false negative rate $\beta = 0.05$ for the sequencing technology to generate simulated matrices B for a subset of 12 AML patients [30]. Specifically, the prevalence of clone i is the number of cells in the sample mapped to clone i divided by the total number of cells in the sample. We consider six patients with linear trees and mutations ranging from 3-5 and six patients with branched trees and mutations ranging from 3-7. A total of 10 replications were performed per simulated patient. Table 4.1 shows a summary of the patients selected for inclusion in the simulation study.

To utilize the Bayes factor, we require knowledge of the parameters of the prior distribution, μ_{branched} and s_{branched} , under the branched evolution hypothesis H_{branched} . Since these are challenging to estimate, we demonstrate the first heuristic of classification utilizing a strict threshold at true false negative rate $\beta^* = 0.05$ and the second heuristic of applying Bayes factor on a theoretical distribution with $\mu_{\text{branched}} = 0.15$ and $s_{\text{branched}} = 10$. For the linear evolution hypothesis H_{linear} , we utilize $\mu_{\text{linear}} = 0.05$, with precision $s_{\text{linear}} = 10$, matching known values for this sequencing platform [30].

We set an upper limit on runtime of PHYOLIN at 500 seconds with 80% of the replications returning an optimal solution in under the time limit. We chose the 500 second time limit to facilitate timely analysis of the input data. Fig. 4.3a shows the distribution of runtime by the simulated evolutionary pattern. Linear patterns resulted in a median runtime of 33

patient	pattern	# mutations m	# cells n	median flips	PHYOLIN median flips	median $\hat{\beta}$	med prob.
AML-2	Linear	5	7931	1826	1039	0.037	0.70
AML-8	Linear	3	4675	759	294	0.029	0.42
AML-10	Linear	4	8729	1427	584	0.030	0.56
AML-33	Linear	3	8120	1091	350	0.027	0.41
AML-47	Linear	3	6491	1135	488	0.032	0.42
AML-58	Linear	3	8170	1280	472	0.029	0.40
AML-53	Branched	3	8013	544	2220	0.44	0.39
AML-62	Branched	6	4027	726.5	2299	0.19	0.58
AML-63	Branched	4	8347	1238.5	1432	0.084	0.44
AML-67	Branched	7	6024	1061.5	6440	0.31	0.71
AML-69	Branched	3	7462	651.5	1037	0.16	0.29
AML-74	Branched	5	9279	1020	2122	0.17	0.38

Table 4.1: **Simulation study based on characteristics of a published AML cohort [30].** Shown is the patient identifier, the published evolutionary pattern of the tree, the number of mutations, the total cells sequenced [30], the median number of simulated false negatives over 10 replications, PHYOLIN estimated number of false negatives over 10 replications, the median $\hat{\beta}$ over 10 replications, and the median probability of a linear perfect phylogeny as determined by the comparison deep learning method [128].

seconds (IQR: 8-82 seconds) and branched patterns resulted in a median of 156 seconds (IQR: 117-502 seconds). The median input size (cells $\times$ mutations) of the linear patterns was 24, 435 and 28, 775 for branched patterns. The largest input was for AML-74 with 9, 279 cells by 5 mutations and no replications completed within the time limit. Only 1 replication with a linear pattern did not complete within the time limit. This implies that optimal solutions are found much faster when the true pattern is linear.

Fig. 4.3b compares the distribution of the estimated false negative rate $\hat{\beta}$ for the patients with linear versus branched published trees over all 10 replications. The simulated system false negative rate was 5% and is shown as a dashed line where relevant. From Fig. 4.3b, we note a significant difference in the distributions between linear and branched instances. Additionally, the median of the linear perfect phylogeny patients is 0.03 (IQR: 0.028-0.032) and every linear replication is less than $\beta^* = 0.05$ while the median of the branched perfect phylogeny patients is 0.19 (IQR: 0.16-0.31) and every branched replication is greater than $\beta^* = 0.05$.

Fig. 4.3c shows the mean $\hat{\beta}$ and standard error for each simulated patient over the 10 replications. The number of mutations are also shown in order to investigate if increasing number of mutations increases $\hat{\beta}$. Although there appears to be some effect when increasing the number of mutations, it does not strictly hold. However, utilizing a strict threshold of $\beta^* = 0.05$ results in perfect classification of the topology for all patients and all replications.

Fig. 4.3d compares the difference between Bayes factor K for patients with simulated linear

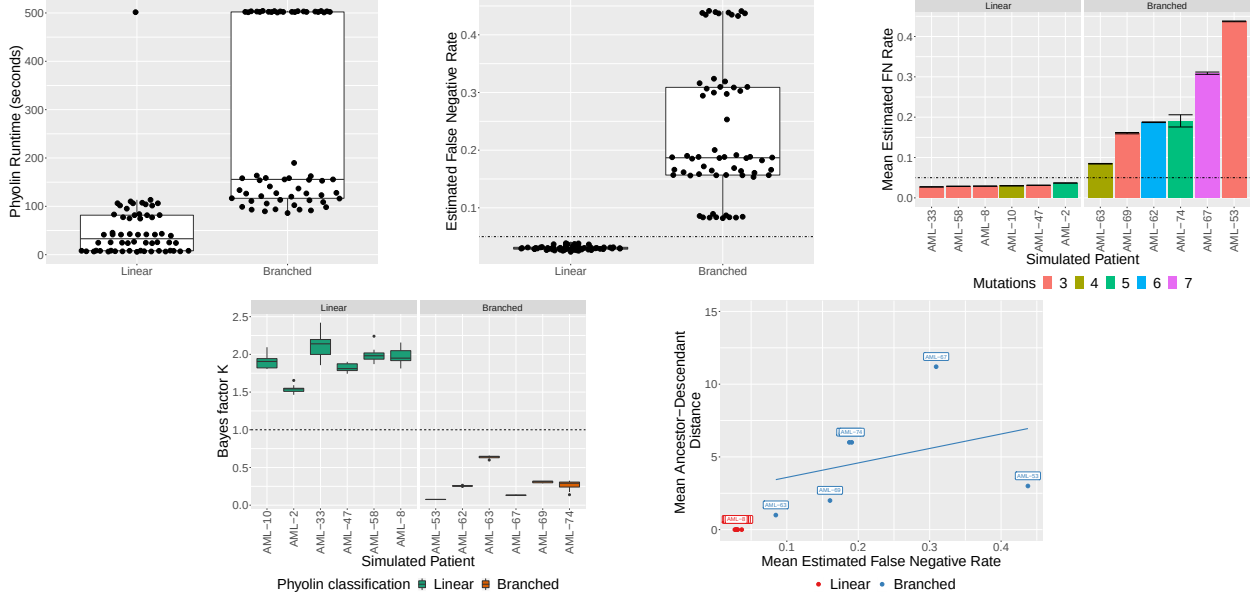


Figure 4.3: **PHYOLIN** results of the *in silico* experiments on a simulated cohort of patients with AML. a) A comparison of PHYOLIN runtimes in seconds between instances with different evolutionary patterns. b) A comparison of the distribution of $\hat{\beta}$ between simulated linear and branched topologies over 10 replications. c) The mean value of $\hat{\beta}$ for each patient along with the standard error. d) The distribution of Bayes factor K for each simulated patient separated by the true evolutionary pattern. The horizontal line at $K = 1$ represents the classification criteria ($K > 1$ is classified as linear). e) Relationship between estimated false negative rate and the ancestor-descendant distance. Each point represents the mean value over 10 replications and is labeled by the numerical patient identifier. A linear trend line is shown with a 95% confidence interval.

and branched perfect phylogenies. Furthermore, without the presence of other sequencing errors, classification via a threshold $\beta^* = 0.05$ and Bayes factor K are in agreement on all simulation replications. Additionally, Bayes factor K provides a more robust interpretation of the evidence for the two alternative hypotheses. For example, the median Bayes factor K for AML-63 is 0.65, which is maximum among all patients with simulated branching. AML-63 also had the lowest estimated median estimated false negative rate ($\hat{\beta} = 0.084$).

Since the number of mutations does not necessarily impact the estimated false negative rate $\hat{\beta}$, another consideration is whether or not $\hat{\beta}$ increases with the amount of branching. To this end, we compare the ancestor-descendant distance between the simulated true tree B^* and the inferred linear tree B' . A mutation x is an *ancestor* of mutation y if x occurs on the path from the root to y , in which case y is said to be a descendant of x . *Ancestor-descendant* (AD) *distance* is defined as the size of the symmetric difference between the sets of ordered pairs of characters, or ancestor-descendant pairs, introduced on distinct edges of

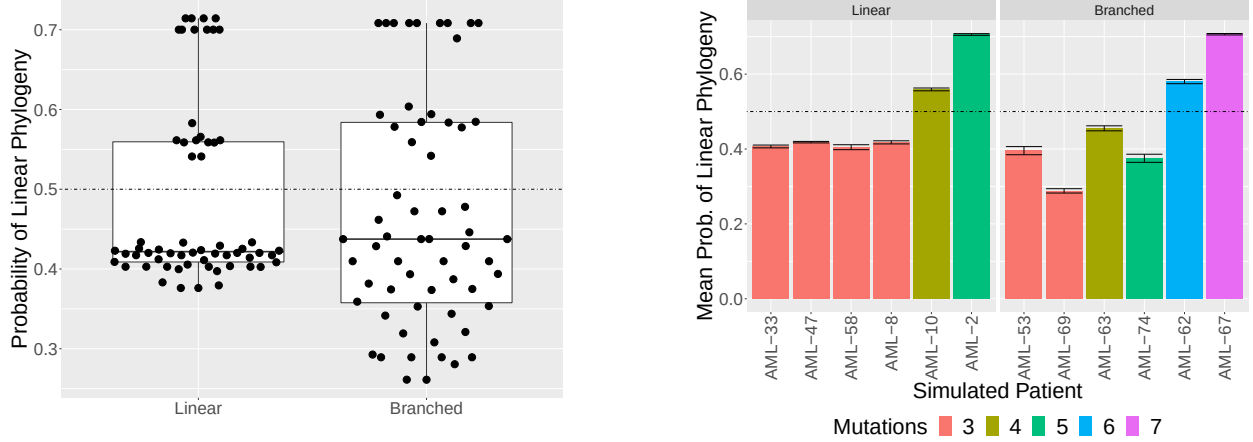


Figure 4.4: **Results of the deep learning approach [128] applied to the *in silico* experiments on a simulated cohort of patients with AML.** a) A comparison of the distribution of the probability of a linear perfect phylogeny between simulated linear and branched topologies over 10 replications. d) The mean value of the probability of a linear perfect phylogeny for each patient along with the standard error. A horizontal line indicates the threshold probability (0.5) used to classify an input as linear.

perfect phylogenies B^* and B' . A higher AD distance implies a greater degree of branching in the true tree. For example AML-63 has only one branch and AML-67 has three distinct branching events. Fig. 4.3e shows the relationship between the mean estimated false negative rate $\hat{\beta}$ and the mean AD distance per simulated patient over 10 replications. The AD distance is 0 for all patients with a simulated linear perfect phylogeny. This means that PHYOLIN correctly infers the true tree when it is linear. In particular, for the branched instances, there is evidence of a correlation between the estimated false negative rate $\hat{\beta}$ and the degree of branching as captured by the AD distance (Fig. 4.3e).

Azer et al.'s [128] deep learning method for classifying topology is the most similar method for comparison with PHYOLIN. Therefore, we retrained this deep neural network to support our input size of 9300 cells and 7 mutations. We used a default hidden layer size of 100 and drop-out rate of 0.9, 5000 training examples and 500 epochs. We did not modify any other hyperparameters. The input size was selected so that only one network needed to be trained for all *in silico* experiments and we used padding for any instances where $n < 9300$ or $m < 7$. After 200 epochs the best validation accuracy was 64.1% and completed in 2168 seconds (36.1 minutes). After 500 epochs, the best validation accuracy was 64.8% and completed in 3997 seconds (66.6 minutes). This suggests that further learning was unlikely. We report the probability that the phylogeny is linear on the same simulation instances when evaluated with the trained model.

Table 4.1 shows the median probability that the phylogeny is linear over the 10 replications

per simulated patient. We use a cutoff of 0.5 as the threshold for classifying a topology as linear. Fig. 4.4a shows the distribution of the probabilities over all patient replications by ground truth topology. Classification accuracy was 100% for 6 of the 12 simulated patients (Linear: AML-2, AML-10, Branched: AML-53, AML-63, AML-69 and AML-74) and 0% for the remaining 6 simulated patients. Fig. 4.4b shows mean estimated probability per patient and standard error for the 10 replications. The predicted probability tends to increase as the number of mutations increases.

In summary, the simulated AML cohort results show that, in contrast to the deep learning approach [128], PHYOLIN correctly and quickly classifies large instances as linear with a strict threshold β^* set at the system estimated false negative rate. Furthermore, as the amount of branching increases, the estimated $\hat{\beta}$ tends to increase. Thus, the greater the difference between $\hat{\beta}$ and β^* , the more confident we can be in rejecting the hypothesis of linear evolution in favor of the alternative hypothesis of branched evolution.

4.3.2 Robustness to false positives and doublets

While the previous simulation study focused only on false negatives, here we examine the robustness of PHYOLIN to the presence of false positives and doublets. In addition, we demonstrate the technique of estimating the parameters of branched evolution hypothesis H_{branched} . First, we generated a set of matrices representing a perfect phylogeny with 500 cells and mutations $m \in \{10, 25, 50\}$ using the `ms` simulator [134]. We then introduced errors to each simulated matrix B with false negative probability $\beta \in \{0.05, 0.15, 0.25\}$, false positive probability $\alpha \in \{0.001, 0.01\}$ and doublet probability $\delta \in \{0.0, 0.2\}$. Note that $z = [\alpha \sum_{i=1}^n \sum_{j=1}^m x_{ij}]$ for each simulation instance. We performed 5 replications for each combination of conditions and each type of evolution, linear or branched, for a total of 360 instances. To account for the added complexity of false positives and doublets, we allowed each PHYOLIN instance to run for a maximum of 900 s. Independently, we created an additional set of experiments with only branched evolution with the same parameters as the simulation. We utilized this set of experiments to estimate $\mu_{\text{branching}}$ and $s_{\text{branching}}$ from the PHYOLIN outputted false negative probability $\beta_{\text{branching}}$ under branched evolution separately for each $m \in \{10, 25, 50\}$. Using these parameters, we determined the category of evolution as linear or branched in accordance with the Bayes factor K .

We found PHYOLIN to be robust to both the presence of false positives and doublets with an overall accuracy of 97% (Fig. 4.5a). For doublet probability $\delta = 0.0$, 85 out of 90 branched instances were correctly classified with 5 instances incorrectly classified as linear (median Bayes factor $K = 0.94$). A total of 87 out of 90 linear instances were correctly classified with 3

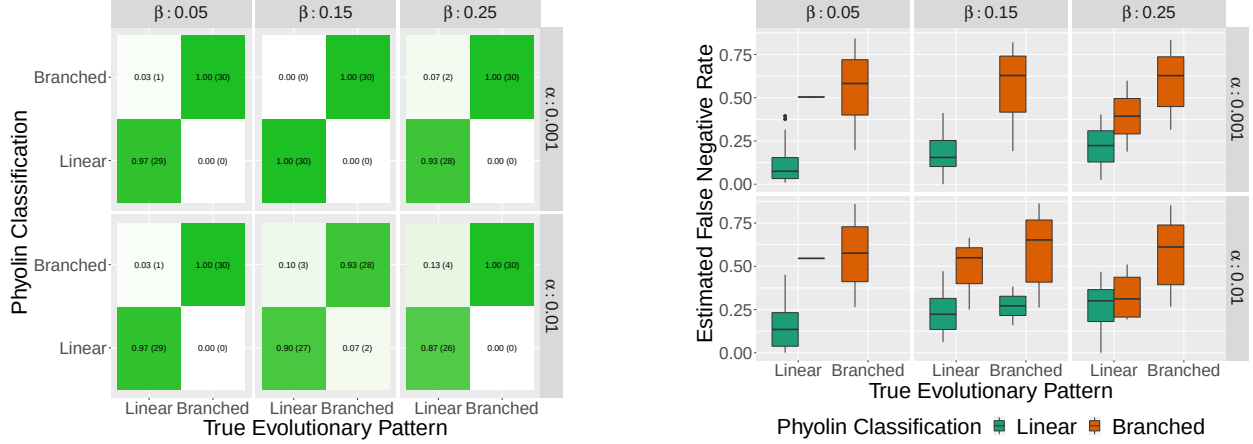


Figure 4.5: **PHYOLIN demonstrates robustness to false positives and doublets.** a) The confusion matrix for PHYOLIN classifications versus the true evolutionary pattern annotated by the proportion (total) of instances. b) The distribution of PHYOLIN estimated false negative rates by true evolutionary pattern with colors depicting the PHYOLIN classification at simulated false negative rates $\beta \in \{0.05, 0.15, 0.25\}$, simulated false positive rates $\alpha \in \{0.01, 0.001\}$ and for double rates $\delta \in \{0.0, 0.2\}$.

instances incorrectly classified as branched (median Bayes factor $K = 0.005$). Paradoxically, for doublet probability $\delta = 0.2$ performance improves with only two linear and two branched evolutionary patterns incorrectly classified. This is explained by the fact that when the true evolutionary pattern is branched, a doublet comprised of cells on distinct branches of a tree will result in fewer cases of branching that must be flipped by PHYOLIN. Therefore, when high doublet rates are suspected, experiments believed to be doublets should be removed from the data prior to utilizing PHYOLIN. In addition, we observe a significant difference in the PHYOLIN estimated false negative rate $\hat{\beta}$ between simulated instances with linear (median: 0.18) and branched (median 0.61) evolution (Fig. 4.5b).

4.3.3 Real data of childhood acute lymphoblastic leukemia patients

Gawad et al. [120] performed single-cell DNA sequencing on a cohort of six patients with childhood acute lymphoblastic leukemia (ALL). As a subtype of leukemia, ALL is also postulated to follow both linear and branched trajectories [135]. We evaluate PHYOLIN on two of the six patients in this cohort: Patient 2 and Patient 6 (Table 4.2). The input size for Patient 2 was 115 cells by 16 mutations with an estimated false negative rate $\beta^* = 0.18$ [120]. Two independent, previous analyses of the sequencing data of Patient 2 suggested a branched topology [74, 124]. For Patient 2, PHYOLIN estimated a false negative rate of 0.36, which is much greater than the previously estimated rate of 0.18 [120]. Moreover, the Bayes

patient	cells sequenced [120]	mutations	PHYOLIN flips	$\hat{\beta}$	β^* [120]	K
Patient 2	115	16	403	0.36	0.18	0.3
Patient 6	146	10	191	0.15	0.18	14.0

Table 4.2: **Summary of PHYOLIN analysis of two patients with ALL.** Shown is the patient identifier, the number of cells sequenced [120], the number of mutations, the number of flips performed by PHYOLIN, the estimated false negative rate $\hat{\beta}$, the false negative rate threshold β^* [108] that was estimated for the sequencing technology and Bayes factor K .

factor $K = 0.26$ implies accepting the branched evolution H_{branched} hypothesis. This concurs with the branched trees published in [74, 124].

In addition, we consider Patient 6 because this patient was analyzed by the deep learning method [128]. The input size for Patient 6 was 146 cells by 10 mutations with an estimated false negative rate $\beta^* = 0.18$. For Patient 6, PHYOLIN estimated a false negative rate of 0.15, which is less than the published false negative rate $\beta^* = 0.18$ [120]. Bayes factor $K = 14.0$ substantially supports the linear evolution hypothesis H_{linear} . The comparison deep learning approach [128] also concluded that the phylogeny was linear. Table 4.2 summarizes results obtained by PHYOLIN.

4.3.4 Application to an acute myeloid leukemia cohort

We assess the performance of PHYOLIN by applying our method to single-cell DNA sequencing data for 9 of the 12 patients included in our simulation study. We exclude 3 patients (AML-10, AML-33, AML-58) with greater than 0.1 fraction of data missing. In addition, we analyze an additional 5 patients with trees highlighted in the original study [30] for a total of 14 patients (9 branched, 5 linear). To obtain the input matrices B , we extract variant and total read counts from the loom files generated via the Mission Bio Tapestry pipeline. To discretize the read counts, we use the binomial exact test with null error rate 0.001 and p-value 10^{-6} . Sites with 0 total reads are labeled as missing data and ignored by PHYOLIN.

Given that the median estimate false negative rate for the cohort was $\beta = 0.058$, we utilize prior parameters mean $\mu_{\text{linear}} = 0.058$ and precision $s_{\text{linear}} = 10$ for our linear evolution hypothesis H_{linear} . We parameterize the branched evolution hypothesis H_{branched} with $\mu_{\text{branched}} = 0.22$ and $s_{\text{branched}} = 11.8$ guided by our simulation study. PHYOLIN’s classification of the evolutionary pattern corresponded to the reported trees [30] in 12 out of the 13 instances. Fig. 4.6a demonstrates a significant difference between the PHYOLIN estimated false negative rate for linear (median: 0.04) and branched evolutionary (median: 0.16) as determined by Morita et al. [30, 88]. Furthermore, the Bayes factor K (Fig. 4.6b) similarly

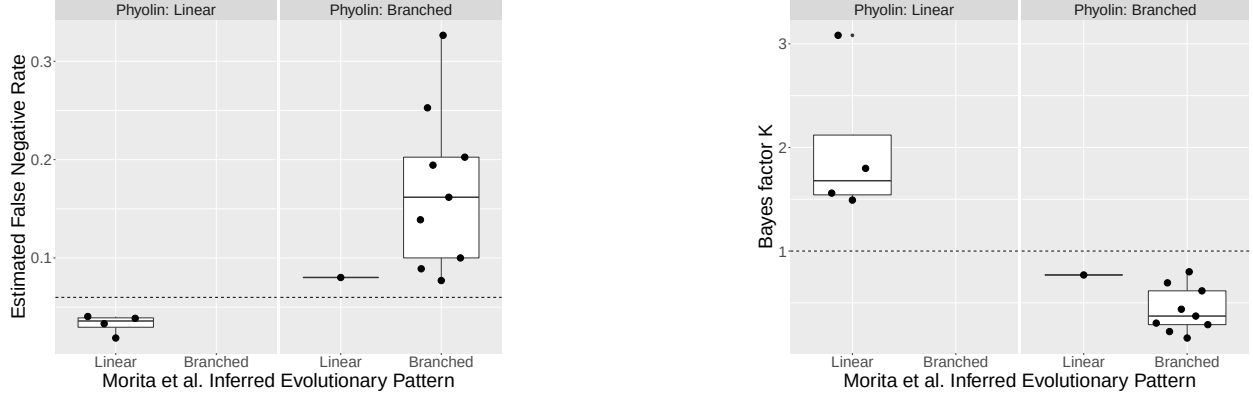


Figure 4.6: **PHYOLIN concurs with Morita et al.[30, 88] on the evolutionary pattern on a cohort patients with acute myeloid leukemia.** a) The distribution of PHYOLIN estimated false negative rates by evolutionary pattern reported by Morita et al. and by PHYOLIN classification with published false negative rate $\beta^* = 0.06$ (horizontal line) and false positive $\alpha = 0.01$ [30]. b) The distribution of the Bayes factor K by evolutionary pattern reported by Morita et al. and by PHYOLIN classification with published false negative rate $\beta^* = 0.06$ and false positive $\alpha = 0.01$ [30]. The horizontal line at $K = 1$ depicts the threshold for classification.

separates the two classes of evolutionary patterns as Morita et al. [30, 88] Only patient AML-02 differed with PHYOLIN classifying the evolutionary pattern as branched ($\hat{\beta} = 0.08$). However, in the initial bioRxiv supplement [88], Morita et al. reported 4 trees for each patient by running SCITE in different modes. We note that 2 trees were branched while 2 were linear. Thus, the Bayes factor $K = 0.77$, helps contextualize the ambiguity surrounding the true evolutionary trajectory for this patient.

4.4 DISCUSSION

In this work, we introduced the LINEAR PERFECT PHYLOGENY FLIPPING problem and showed that it is NP-hard. To solve this problem, we developed a method named PHYOLIN that takes as input a binary matrix of single-cell DNA sequencing data and then identifies a linear perfect phylogeny in the data by assuming that any implied branching are actually false negatives. The outputted number of flips from PHYOLIN is then used to compute the strength of the evidence of the two alternatives of the pattern of evolution. We tested PHYOLIN on both simulated data and real data and showed that it is more accurate than a recent deep learning method [128] and concurs with previously inferred evolutionary patterns on two real datasets. In conclusion, PHYOLIN is a reliable, easy to use and fast method to assess the likelihood of a linear evolution before more complex reconstruction methods are

utilized.

There are several future research directions. First, PHYOLIN could easily incorporate mutation and cell clustering through additional constraints when supplied with a number of cell clusters and/or mutation clusters. A search could be performed to find the optimal number of clusters such that the likelihood of the data is maximized. Second, even when the phylogeny is branched, the trunk of the tree may be linear or there might be a long branch with linear evolution within that branch. This means that a subset of the mutations form a linear perfect phylogeny. A future direction is to explore if PHYOLIN can identify a subset of mutations that are likely to be truncal or form a long branch of the tree, thus potentially providing fast partial inference of the tree.

Finally, exploring evolutionary models that allow ISA violations, such as mutation loss, is an exciting direction for future study. In particular, the 1-Dollo evolutionary, where each mutation is gained exactly once and subsequently lost at most once, could potentially be achieved by replicating each column once and then using PHYOLIN. If the two columns representing the same mutation are distinct in the inferred linear perfect phylogeny, then that implies that the mutation was lost once. The plausibility of a linear perfect phylogeny under both ISA and under a 1-Dollo model could be compared [42].

CHAPTER 5: PHERILIZER: GROWING A CLONAL TREE FROM ULTRA-LOW COVERAGE SINGLE-CELL DNA SEQUENCING OF TUMORS

5.1 INTRODUCTION

Two types of somatic mutations that occur in tumors that arise as the evolutionary process of cancer ((Fig 5.1a) include single-nucleotide variants (SNVs) that alter a single base and copy-number aberrations (CNAs) that amplify or delete large genomic regions. The ongoing development and application of high-throughput, ultra-low coverage SCDNA-SEQ technologies ($< 1\times$), such as direct library preparation (DLP+) [24] and acoustic cell tagmentation (ACT) [23], have paved the way for enriching our understanding of the role CNAs play in cancer progression and tumor evolution [23, 24, 25].

The advantage these ultra-low coverage SCDNA-SEQ technologies have over other high-throughput SCDNA-SEQ methods ($> 1\times$), like Mission Bio Tapestry [11], is the uniformity of coverage. This uniformity implies that the observed read counts for a genomic region is proportional to copy number, making it ideal for the analysis of subclonal CNAs that occur in only a small subset of tumor cells. However, this uniformity comes at the cost of sequencing depth, making it very difficult to identify and characterize the evolution of SNVs from ultra-low coverage SCDNA-SEQ. Critically, to comprehensively study the evolution of a tumor from the same set of cells, both CNAs and SNVs should ideally be characterized by a single tumor phylogeny that depicts their coevolution. While this remains a long-term goal for the field, a first step in this direction is to increase our understanding of SNV evolution from ultra-low coverage SCDNA-SEQ data by incorporating reliable copy number information into the inference of SNV tumor phylogenies. Although phylogeny inference methods from bulk sequencing and cell clustering from single-cell RNA sequencing are expanding to incorporate both SNV and CNA features, such as TUSV-ext [136] and CASIC [137], current methods for tumor phylogeny and/or clone inference from single-cell sequencing naturally tend to focus on the features (SNV or CNA events) for which the data is ideally suited [38, 39, 42, 43, 44, 46, 138, 139, 140, 141, 142, 143, 144, 145]. One exception is BiTSC2 [143], which infers a phylogeny containing both SNV and CNA events.

⁶This chapter contains identical material with an adapted introduction previously published in “Phertilizer: Growing a clonal tree from ultra-low coverage single-cell DNA sequencing of tumors” [69], which is an extended version of the paper from RECOMB-CCB 2022 by the same title. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with C. Zhang, I. Ochoa and M. El-Kebir. MEK and IO conceived the project; LLW and MEK established theory; LLW and CZ performed experiments; LLW, CZ, IO and MEK contributed to the writing of this chapter.

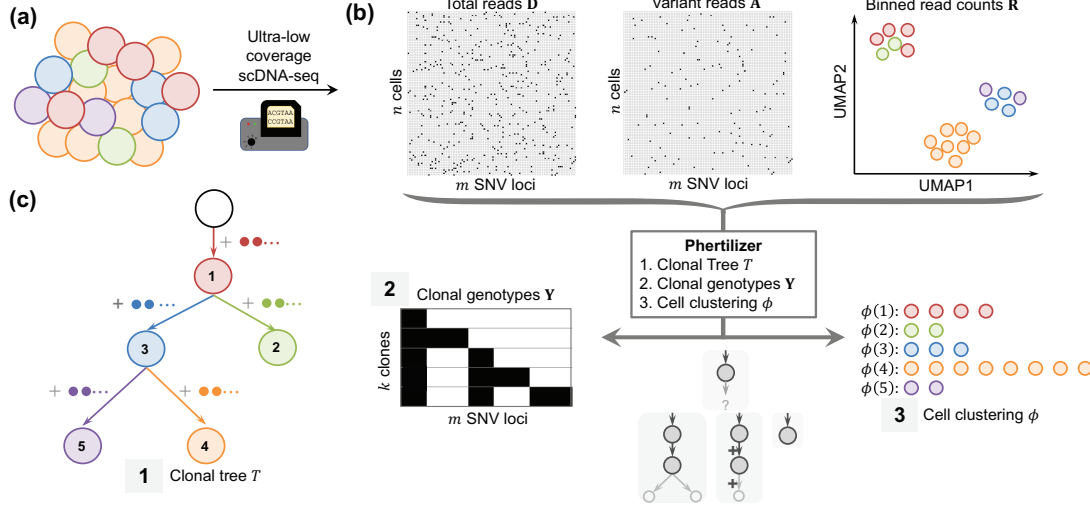


Figure 5.1: **PHERTILIZER** infers a clonal tree T , clonal genotypes $\mathbf{Y}$ and a cell clustering ϕ given ultra-low coverage single-cell sequencing data. (a) A tumor consists of clones with distinct genotypes. (b) Ultra-low coverage SCDNA-SEQ produces total read counts $\mathbf{D} \in \mathbb{N}^{n \times m}$ and variant read counts $\mathbf{A} \in \mathbb{N}^{n \times m}$ for n cells and m SNV loci, and low dimension embedding of binned read counts $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$. (c) Given maximum copy number c and sequencing error probability α , PHERTILIZER infers a clonal tree T , clonal genotypes $\mathbf{Y}$ and cell clustering ϕ with maximum posterior probability $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$.

However, the method is not designed for high-throughput ultra-coverage data, with efficacy demonstrated on datasets containing at most 500 cells and coverages as low as $3\times$. Another exception in the medium to high coverage SCDNA-SEQ regime ($> 10\times$) is SCARLET [122], which refines a given copy number tree using SNV read counts under a CNA loss supported evolutionary model. While SCARLET accounts for sequencing errors and missing data, it was not designed to handle the extreme sparsity of ultra-low coverage SCDNA-SEQ. SBMCLONE [146] took the first step of using ultra-low coverage sequencing data to infer SNV clones via stochastic block modeling. Despite good performance on simulated data, especially with higher coverage ($> 0.2\times$), SBMCLONE was unable to identify clear structure in a 10x Genomics breast cancer dataset [87] without *ad hoc* use of additional copy number clone information. Moreover, it is non-trivial to convert the inferred parameters of SBMCLONE’s stochastic block model to clonal genotypes, which may impact downstream analyses. Similarly, given a set of candidate SNV loci, SECEDO [147] first calls SNVs using a Bayesian filtering approach and then subsequently clusters cells using the called SNVs. While both of these clustering methods capitalize on the ever-increasing throughput of ultra-low coverage SCDNA-SEQ methods, neither method constrains the output by a tree and CNA features are used only in an *ad hoc* manner or for orthogonal validation. As a result, both meth-

ods imply that CNA and SNV data features should be segregated and analyzed in separate bioinformatics pipelines. The other emerging trend from the analysis of ultra-low coverage SCDNA-SEQ is pseudobulk analysis [24]. This approach begins by identifying copy number clones using existing methods, followed by pooling cells that belong to the same CNA clones into pseudobulk samples, which are then independently analyzed to identify SNVs. Finally, phylogeny inference is performed with the copy number clones as the leaves of the tree. By doing so, this method does not allow for further refinement of these clones based on SNV evolution.

Here we introduce PHERILIZER, the first method to infer an SNV clonal tree from ultra-low single-cell DNA sequencing of tumors. To overcome SNV coverage sparsity in this type of data, we leverage the strong copy number signal inherent in the data to guide clonal tree inference. By analogy to the planting and growing of trees, PHERILIZER seeks to grow a clonal tree with maximum posterior probability by recursively inferring elementary clonal trees as building blocks. Our simulations demonstrate that PHERILIZER accurately infers phylogenies and cell clusters when the number of cells matches current practice. In particular, PHERILIZER outperforms a current method [146] for simultaneously clustering SNVs and cells as well as another commonly used *ad hoc* approach [24]. On real data, we find that PHERILIZER effectively utilizes the copy-number signal inherent in the data to uncover clonal structure, yielding high-fidelity clonal genotypes.

5.2 MATERIALS AND METHODS

5.2.1 Problem statement

Our goal is to infer an SNV phylogeny, guided by copy number aberrations, from ultra-low coverage sequencing data consisting of n cells and m identified single-nucleotide variants (SNVs). More precisely, we are given variant reads $\mathbf{A} = [a_{iq}]$ and total reads $\mathbf{D} = [d_{iq}]$, where a_{iq} and d_{iq} are the variant and total read counts for SNV locus $q \in [m]$ in cell $i \in [n]$, respectively (Fig 5.1b). While the number of cells is large with the latest generation of ultra-low coverage single-cell DNA sequencing technology ($n \approx 1000$ cells), the *coverage*, or the average number of reads that span a single locus is uniform but low ($0.01\times$ to $0.5\times$). For example with a coverage of $0.01\times$, we would on average observe $a_{iq} = d_{iq} = 0$ reads for 99 out of every 100 loci q for each cell i . This sparsity renders phylogeny inference using SNVs extremely challenging. We propose to overcome this challenge using the following three key ideas.

First, similarly to current methods, we leverage the clonal structure present in tumors, i.e.,

cells typically cluster into a small number of clones. Thus, we seek to group the n observed cells into k clones ($k \ll n$) via a cell clustering and corresponding clonal genotypes defined as follows:

Definition 5.1. Function $\phi : [k] \rightarrow 2^{[n]}$ is a *cell clustering* provided its image encodes a partition of cells $[n]$ into k (disjoint and non-empty) parts.

Definition 5.2. Matrix $\mathbf{Y} \in \{0, 1\}^{k \times m}$ encodes *clonal genotypes* where $y_{jq} = 1$ indicates that SNV q is present in clone j and $y_{jq} = 0$ indicates that SNV q is absent in clone j .

Second, because ultra-low coverage SCDNA-SEQ is sequenced uniformly, we may leverage the copy number signal inherent in the data to improve cell clustering performance [137] and guide tree inference. More specifically, we expect that all cells in a clone have identical copy number profiles. As we do not observe copy number directly, we will use observed reads counts $\mathbf{R} \in \mathbb{N}^{n \times b}$, where b is the number of genomic bins, as a proxy for copy number. From read counts $\mathbf{R}$, we derive distances that reflect copy-number similarity between pairs of cells on a low-dimensional embedding of binned read counts $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$ of $\mathbf{R}$ (i.e., $\ell \ll b$)—see Appendix C.1.1. Third, similarly to methods such as SCITE [38], SciCloneFit [43] and SPhyR [42], which operate on medium-to-high coverage SCDNA-SEQ data, we consider that the observed cells are generated as the result of a tree-like evolutionary process that constrains the order of SNV clusters. In particular, we use the infinite sites model [148] defined as follows.

Definition 5.3. A tree T with nodes $\{v_1, \dots, v_k\}$ rooted at node v_1 is a *clonal tree* for clonal genotypes $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_k]^\top$ provided (i) each node v_j is labeled by clonal genotype $\mathbf{y}_j$ and (ii) each SNV q is gained exactly once and subsequently never lost. That is, there exists no directed edge $(v_{j'}, v_j)$ where the SNV is lost, i.e., $y_{jq} = 1$ and $y_{j'q} = 0$. Moreover, either the root node contains the SNV, i.e., $y_{1q} = 1$, or there exists exactly one directed edge $(v_{j'}, v_j)$ where the SNV is introduced, i.e., $y_{j'q} = 0$ and $y_{jq} = 1$.

To relate the data to our latent variables of interest $(T, \mathbf{Y}, \phi)$, we introduce a generative model in Appendix C.1.2 that describes the generation of variant read counts $\mathbf{A}$ and the binned read count embedding $\tilde{\mathbf{R}}$. This model (Fig. C.1) requires two hyperparameters c and α , where $c \in \mathbb{N}$ is the upper bound on the total number of chromosomal copies at any locus in the genome and $\alpha \in [0, 1]$ is the probability of misreading a single nucleotide during sequencing. Importantly, while Definitions 5.1 to 5.3 explicitly indicate the number k of clones, the number k of clones is not a hyperparameter and will be part of the inference. Specifically, our generative model enables us to approximate the posterior probability

$P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$ for a clonal tree T with any number k of nodes and associated clonal genotypes $\mathbf{Y}$ and cell clustering ϕ (derivation in Appendix C.1.2). However, due to limitations of the sequencing technology the number of clones that are detectable from the data may be fewer than the number of nodes in clonal tree T . The ability to detect clone j from the observed data is a function of the sequencing coverage, the number of cells in clone j and the number of SNVs newly introduced in clone j . To prevent overfitting of the data, the pre-specified detection threshold $t \in \mathbb{N}$ controls the minimum amount of observed data in support of each inferred clone. See Appendix C.1.3 for a formal definition and additional details.

This leads to the following problem.

Problem 5.1 (CLONAL TREE INFERENCE (CTI)). Given variant reads $\mathbf{A} \in \mathbb{N}^{n \times m}$, total reads $\mathbf{D} \in \mathbb{N}^{n \times m}$, binned read count embedding $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$, maximum copy number $c \in \mathbb{N}$, sequencing error probability $\alpha \in [0, 1]$ and detection threshold $t \in \mathbb{N}$, find a clonal tree T with detectable clonal genotypes $\mathbf{Y}$ and cell clustering ϕ with maximum posterior probability $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$.

5.2.2 PHERILIZER

To solve the CTI problem, PHERILIZER maintains a set $\mathcal{T}$ of candidate trees throughout three phases: (i) initialization, (ii) growing, and (iii) ranking each tree in $\mathcal{T}$ by its posterior probability. First, in the initialization phase, the set $\mathcal{T}$ is initialized with a single tree containing only a root node v_1 . All n cells are assigned to the node's cell cluster $\phi(v_1)$ and all genotypes are initialized to $y_{v_1, q} = 1$ for each SNV q .

Second, in the growing phase (Appendix C.1.4), PHERILIZER recursively constructs the candidate set $\mathcal{T}$ of clonal trees and the respective clonal genotypes and cell clusterings by performing three different elementary tree operations (**Linear**, **Branching** and **Identity**) on each leaf node v_j of each candidate tree $T \in \mathcal{T}$ (Fig 5.2). Specifically, each operation takes as input $(T, \mathbf{Y}, \phi)$ and yields a new clonal tree T' with updated genotypes $\mathbf{Y}'$ and cell clustering ϕ' by extending leaf v_j of T (Fig 5.2). The key idea is that each elementary tree operation breaks down the CTI problem into smaller subproblems. Intuitively, a **Linear** operation (Fig 5.2a) replaces a leaf node with a two node linear subtree, while a **Branching** operation (Fig 5.2b) replaces the leaf node with a three node binary subtree. The former represents stepwise acquisition of SNVs with evidence of intermediary clones present in the observed data, while the latter indicates evidence of divergence from a common ancestor [53]. The **Identity** (Fig 5.2c) operation does not modify the tree and is useful during the growing

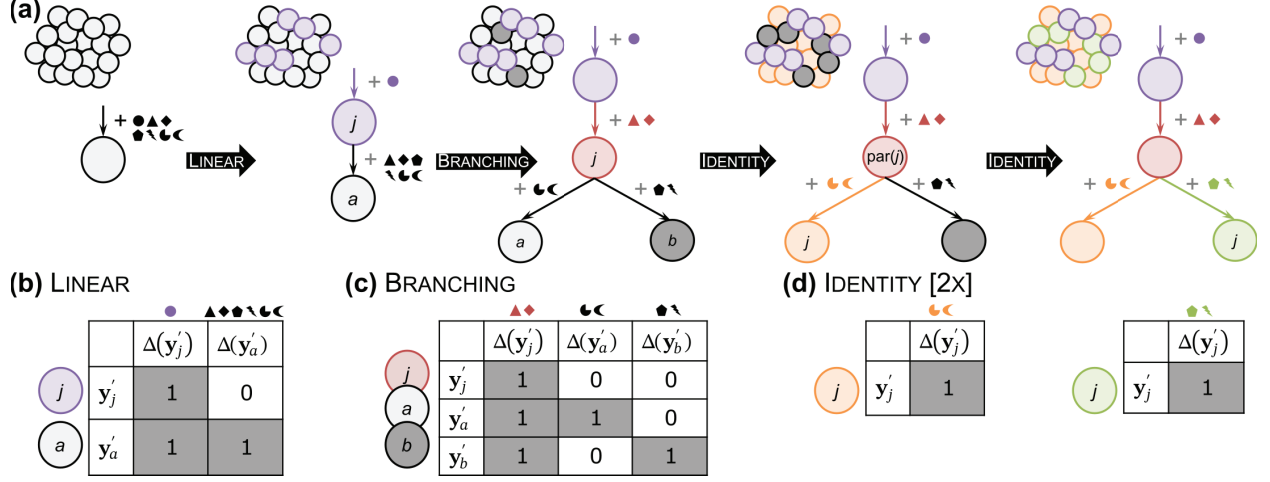


Figure 5.2: **PHERTILIZER** solves the CTI problem by enumerating clonal trees using three elementary operations in a recursive fashion. (a) Each operation yields a new clonal tree T' by extending a leaf v_j of the previous clonal tree T and reassigning its SNVs $\Delta(y_j)$ and cells $\phi(j)$. The resulting clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' are constrained as depicted: (b) Linear, (c) Branching, and (d) Identity.

process where all candidate trees are enumerated.

These operations are defined more formally in Appendix C.1.4. While the specifics vary slightly, both Linear and Branching are solved using a coordinate descent approach. That is, we fix clonal genotypes $\mathbf{Y}'$ and solve for the cell clustering ϕ' and alternate. Drawing parallels between image segmentation, where combining both pixel and pixel location features results in better clustering [149], we incorporate the binned read count embedding $\tilde{\mathbf{R}}$ and variant read counts $\mathbf{A}$ into a single feature [137]. We then use this feature as input to the normalized cut algorithm [149] (worst case running time $O(n^3)$) to obtain a cell clustering with two clusters. The advantage of combining SNV and CNA signal into a single feature is that cell clustering is improved when one or both of these signals are weak, subsequently improving the SNV partition which we solve for next. Given a fixed cell clustering ϕ' , we use our generative model to update clonal genotypes $\mathbf{Y}'$ by assigning each SNV to the node in extended tree T' with maximum posterior probability. This is done in time $O(nm)$. We terminate this process upon convergence or when a maximum number of iterations is reached. This results in the running time of one elementary tree operation as $O(n^3 + nm)$. The resulting clonal tree T' is then appended to the candidate set $\mathcal{T}$ provided all its clone are detectable for a specified detection threshold t and meet additional regularization criteria (Appendix C.1.4).

Third, once no new clonal trees are added to the candidate set $\mathcal{T}$, we return the clonal tree T , clonal genotypes $\mathbf{Y}$ and cell clustering ϕ with maximum posterior probability after post-processing (Appendix C.1.4). Importantly, the top down approach by PHERTILIZER requires

that no assumptions be made *a priori* regarding the number of nodes k in the inferred clonal tree. PHERILIZER is implemented in Python 3, open source (BSD-3-Clause), and available at <https://github.com/elkebir-group/phertilizer>.

5.3 RESULTS

5.3.1 Simulation study

Overview. To assess the performance of PHERILIZER and compare it with previously proposed methods, we performed a simulation study with known ground truth clonal trees, evaluating the following four questions:

(i) How accurate are the inferred clonal trees? (ii) How well is each method able to identify clusters of cells with similar clonal genotypes? (iii) How accurate are the inferred clonal genotypes? (iv) How sensitive is each method to violations of the infinite sites assumption? We designed our simulation study to match the characteristics of current datasets generated from ultra-low coverage SCDNA-SEQ. To achieve this, we generated simulation instances with varying number $k \in \{5, 9\}$ of nodes, number $n \in \{1000, 2000\}$ of sequenced cells and number $m \in \{5000, 10000, 15000\}$ of SNVs with a mean sequencing coverage $g \in \{0.01\times, 0.05\times, 0.1\times\}$. We replicated each of these combinations 10 times for a total of 360 instances. See Appendix C.2.1 for details on the simulation instances.

We assessed the quality of an inferred solution $(T, \phi, \mathbf{Y})$ against a ground-truth tree T^* , cell clustering ϕ^* and clonal genotypes $\mathbf{Y}^*$ using ancestral pair recall (APR), incomparable pair recall (IPR), and clustered pair recall (CPR) metrics for cells and SNVs [127], as well as genotype similarity. In addition, we computed a single *accuracy* value (domain: $[0, 1]$) composed of the weighted average of APR, IPR and CPR — where the weights are proportional to the number of pairs in each class — such that an SNV and cell accuracy of 1 imply that the inferred solution perfectly matches ground truth. The *genotype similarity* equals 1 minus the normalized Hamming distance between ground truth genotypes and inferred genotypes of cells. We refer to Fig 5.3a Appendix C.2.1 for examples and more formal definitions.

We benchmarked against SBMCLONE [146] because it is the only existing clustering/genotyping method that co-clusters cells and SNVs. We also benchmarked against a commonly adopted *ad hoc* practice which we refer to as BASELINE [23, 24]. In BASELINE, cells are first clustered into clones from the read count embedding $\hat{\mathbf{R}}$. Clonal genotypes are obtained by pooling the reads of all cells assigned to clone j and setting $y_{jq} = 1$ when the ratio of alternate to total reads at each locus q exceeds a threshold of 0.05 (details provided in Appendix C.2.1). Since there exists no standalone method that performs cell clustering, geno-

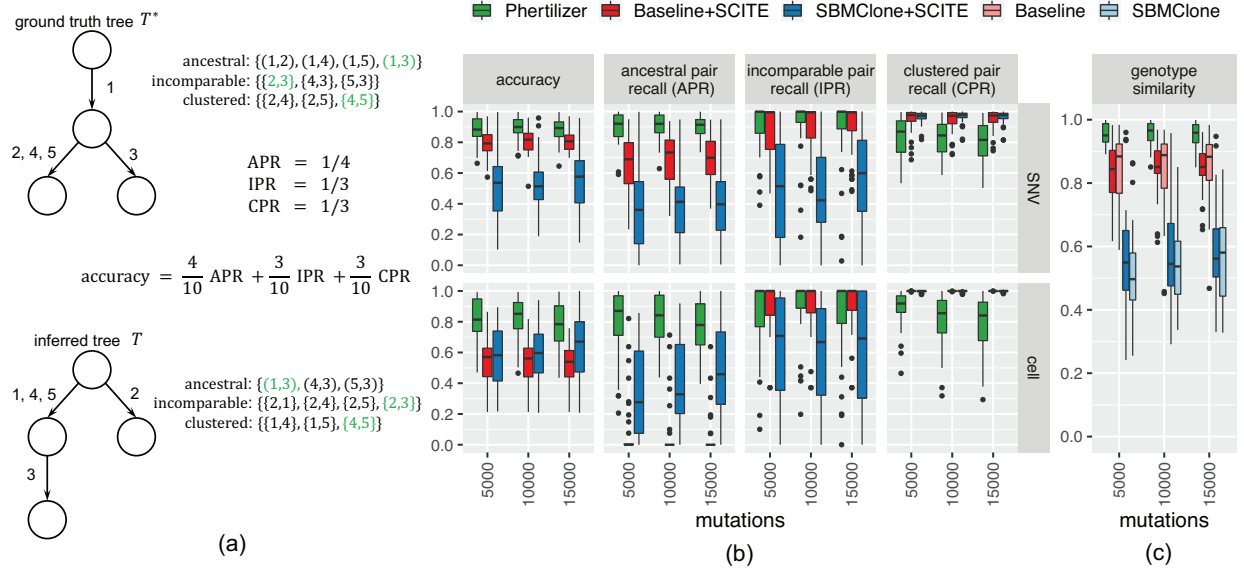


Figure 5.3: **PHERTILIZER outperforms BASELINE and SBMCLONE on simulated data.** We show aggregated results for $n \in \{1000, 2000\}$ cells, $k \in \{5, 9\}$ clones and coverage $g = 0.05\times$. (a) Example of SNV accuracy, ancestral pair recall (APR), clustered pair recall (CPR), and incomparable pair recall (IPR) metrics. See Appendix C.2.1 for the corresponding example for cell metrics and genotype similarity. (b) PHERTILIZER outperforms BASELINE + SCITE and SBMCLONE + SCITE in APR and IPR for both SNVs and cells. Although competing methods rank higher in CPR, overall PHERTILIZER performs the best considering the accuracy. (c) PHERTILIZER more accurately recovers clonal genotypes than competing methods.

typing and tree inference for ultra-low coverage SCDNA-SEQ, we paired SBMCLONE and BASELINE with SCITE [38]. We attempted to include SCARLET [122] and BiTSC2 [143] in our benchmarking as these SCDNA-SEQ methods incorporate copy number aberrations in their models. However both failed to run on our simulated data. Specifically, SCARLET, which was designed for medium-to-high coverage data, was unable to appropriately handle missing entries in the total read count matrix $\mathbf{D}$. BiTSC2, an MCMC method, was unable to scale to the size of our input datasets — the largest instances considered in the BiTSC2 paper [143] had $n = 500$ cells and $m = 200$ SNVs, significantly smaller than our smallest simulation instances with $n = 1000$ cells and $m = 5000$ SNVs.

For brevity, we focus our discussion on a coverage $g = 0.05\times$ and show aggregated results for $n \in \{1000, 2000\}$ cells and $k \in \{5, 9\}$ clones. We report the median for all performance metrics and include the interquartile range (IQR), i.e., the difference between the 75th and 25th percentiles of the data, when appropriate. We note deviations in trends where relevant, and refer to Appendix C.2 for remaining results.

Results. We first evaluated the accuracy of SNV placement on the inferred tree by assessing APR, CPR, IPR and accuracy for SNVs (Fig 5.3b, Fig. C.9). Overall, PHERTILIZER achieved the highest SNV accuracy (median: 0.90) in terms of SNV placement of all three methods (BASELINE+SCITE median: 0.81, SBMCLONE+SCITE median: 0.54).

For SNV APR, PHERTILIZER (median: 0.92) outperformed both BASELINE+SCITE (median: 0.71) and SBMCLONE+SCITE (median: 0.38). This implies that our **Linear** operation reliably partitions SNVs accurately and that our **Branching** operation performs well at identifying SNVs that should be placed at the parent node. Moving on to SNV IPR, PHERTILIZER (median: 1.0, IQR: 0.11) outperformed SBMCLONE+SCITE (median: 0.48, IQR: 0.5) with lower variability than BASELINE+SCITE (median: 1.0, IQR: 0.18). Thus, in addition to correctly identifying the SNVs in the parent node, the **Branching** operation also successfully partitions the SNVs in the children nodes. However, for SNV CPR, PHERTILIZER performed worse compared to SBMCLONE+SCITE (median: 0.98) and BASELINE+SCITE (median: 0.97) but still maintained good performance (median 0.84). It is important to note that an SNV CPR of 1 is achievable by clustering all SNVs into a single cluster. Thus, the cost of underfitting the data, or grouping the SNVs into a few very large clusters, will be reflected in decreased APR and IPR. Indeed, we observed this to be the case for both BASELINE+SCITE, which performed relatively poorly on APR, and SBMCLONE+SCITE which was the worst performing on both APR and IPR.

Next, we evaluated the accuracy of the cell clustering and placement on the inferred tree. We observed similar trends across accuracy, APR, IPR and CPR cell performance metrics to those identified for their SNV counterparts (Fig 5.3b, Fig. C.9). PHERTILIZER achieved the highest overall cell accuracy (median: 0.82) in comparison to SBMCLONE+SCITE (median: 0.60) and BASELINE+SCITE (median: 0.56). Likewise, PHERTILIZER (median: 0.84) outperformed all other methods on cell APR (BASELINE+SCITE median: 0.0, SBMCLONE+SCITE median: 0.32). On cell IPR, both PHERTILIZER (median: 1.0, IQR: 0.2) and BASELINE+SCITE (median: 1.0, IQR: 0.13) significantly outperformed SBMCLONE+SCITE (median: 0.67, IQR: 0.68) but BASELINE+SCITE had slightly lower variability (IQR) than PHERTILIZER. Similarly to SNV CPR, BASELINE+SCITE (median: 1.0) and SBMCLONE+SCITE (median: 1.0) outperformed PHERTILIZER on cell CPR (median: 0.87), but the corresponding decreased performance in cell APR and IPR was indicative of inferring too few cell clusters.

In addition to providing more supporting evidence for the validity of our elementary operations, these cell placement performance metrics also highlight the advantage of PHERTILIZER utilizing both copy number and SNV signal for tree inference. In contrast, BASELINE+SCITE prioritizes copy number signal and is unable to further refine a cluster of

cells with distinct clonal genotypes but the same copy number profile. Conversely, SBMCLONE+SCITE ignores copy number signal and struggles to infer clones with sparse SNV signal.

Finally, we assessed the genotype similarity and included SBMCLONE and BASELINE as this is obtained prior to tree inference. Since genotype similarity compares the inferred genotype of each simulated cell against its ground truth genotype, it captures the interplay between cell clustering and clonal genotyping. Given that PHERILIZER achieved the highest performance on the clonal tree inference and cell clustering metrics, we would expect PHERILIZER to have the highest genotype similarity. Indeed, Fig 5.3c demonstrates that to be true since PHERILIZER was the only method to have a median genotype similarity above 0.95. BASELINE was the next highest (median: 0.88), closely followed by BASELINE+SCITE (median: 0.85), while SBMCLONE+SCITE (median: 0.55) and SBMCLONE (median: 0.53) had the worst performance. When evaluating these metrics at the lowest coverage $g = 0.01\times$, PHERILIZER maintained top performance on both cell placement and genotype similarity but BASELINE+SCITE was competitive with PHERILIZER on SNV placement (Fig. C.8). For the highest coverage $g = 0.1\times$, PHERILIZER achieved the highest accuracy on SNV placement and cell placement and had a median genotype similarity of 0.97 while the next closest competitor (BASELINE) had a median similarity of 0.89 (Fig. C.10). In terms of running time for the case of $n = 1000$ cells, $m = 15000$ SNVs, and a coverage of $g = 0.01\times$, the median running time of PHERILIZER was 460 s, 45.9 s for BASELINE+SCITE and 101 s for SBMCLONE+SCITE (Fig. C.13).

To perform sensitivity analysis, we generated two additional sets of simulations. The first had the same parameters as above but excluded CNAs, such that every locus was heterozygous diploid. We also excluded BASELINE+SCITE from comparison as only a single clone was inferred after cell clustering. We found PHERILIZER still outperformed SBMCLONE+SCITE but that for coverage $g = 0.01\times$ our performance were slightly worse than simulations with CNAs (Fig. C.11), implying CNA features aid inference when sequencing coverage is extremely sparse. For the second, simulations were generated under a Dollo [150] evolutionary model with $k = 9$, $m = 15000$, coverage $g \in \{0.01\times, 0.05\times, 0.1\times\}$. We found that PHERILIZER still outperformed BASELINE+SCITE and SBMCLONE+SCITE, having maintained high scores on all performance metrics (Fig. C.12) with the exception of cell APR for the lowest coverage.

Hyperparameter selection PHERILIZER requires a number of hyperparameters for inference. To assess sensitivity to each of these hyperparameters, we performed an additional simulation study varying the base sequencing error probability $\alpha \in \{0.001^*, 0.01\}$, the maxi-

num number of copies $c \in \{3, 5^*, 9\}$, the detectability threshold $t \in \{3^*, 5, 7, 11\}$ in addition to runtime parameters such as the number of restarts ($\{5, 15^*, 30, 60\}$), maximum number of iterations ($\{25, 50^*, 100, 200\}$) per elementary tree operation and the quality check upper bound $qc \in \{0.025^*, 0.05\}$ for cells not harboring the specified set of SNVs — see Appendix C.1.4 for details on the quality check. Hyperparameter choices with * indicate the default PHERILIZER values used in the above simulation study. In all simulated instances, we fixed the number of cells at $n = 1000$, SNVs at $m = 15000$ and sequencing coverage at $g = 0.05\times$ — see Fig. C.14 and Fig. C.15 for these results.

Overall, we found that overestimating the base sequencing error α led to a small drop in performance in terms of genotype similarity and SNV and cell accuracy. For the remaining comparisons, we fixed $\alpha = 0.001$ and observed that increasing the maximum copies c to 9 slightly improved overall performance. This is likely attributed to the greater number of allele-specific copy number states that are considered for $c = 9$ than $c = 5$. We also found that performance remained similar for the detectability threshold until increased to $t = 11$. Due to the low-sequencing coverage, having such a high detectability threshold yields poorly resolved trees with fewer clones than ground truth. However, variation in the quality check threshold did not lead to significant differences in performance. Lastly, we found that 15 restarts with 50 maximum iterations per elementary tree operations was sufficient to maximize performance. However, it may be prudent to increase these values on real data at the cost of increasing runtime. Although this sensitivity analysis provides guidance for the setting of hyperparameter values, we recommend to perform a grid search over hyperparameters, such as α and c , when these values are difficult to estimate in practice. The posterior probability approximated by PHERILIZER can be utilized to discriminate between output clonal trees with different parameter settings.

In summary, we conclude that given the high level of accuracy obtained by PHERILIZER on these performance metrics, not only are the elementary operations successful in isolation but also the posterior probability is useful in discriminating between candidate clonal trees. Additionally, we find that utilizing copy number signal, whenever available, is necessary when working with ultra-low coverage SCDNA-SEQ data but not sufficient for accurate clonal tree reconstruction and/or SNV genotyping.

5.3.2 High-grade serous ovarian cancer patient

Utilizing a grid search over input parameters (Appendix C.2.2), we ran PHERILIZER on $n = 890$ DLP+ sequenced cells from three clonally-related cancer cells lines sourced from the same high-grade serous ovarian cancer patient [24]. We used the variant and total read counts

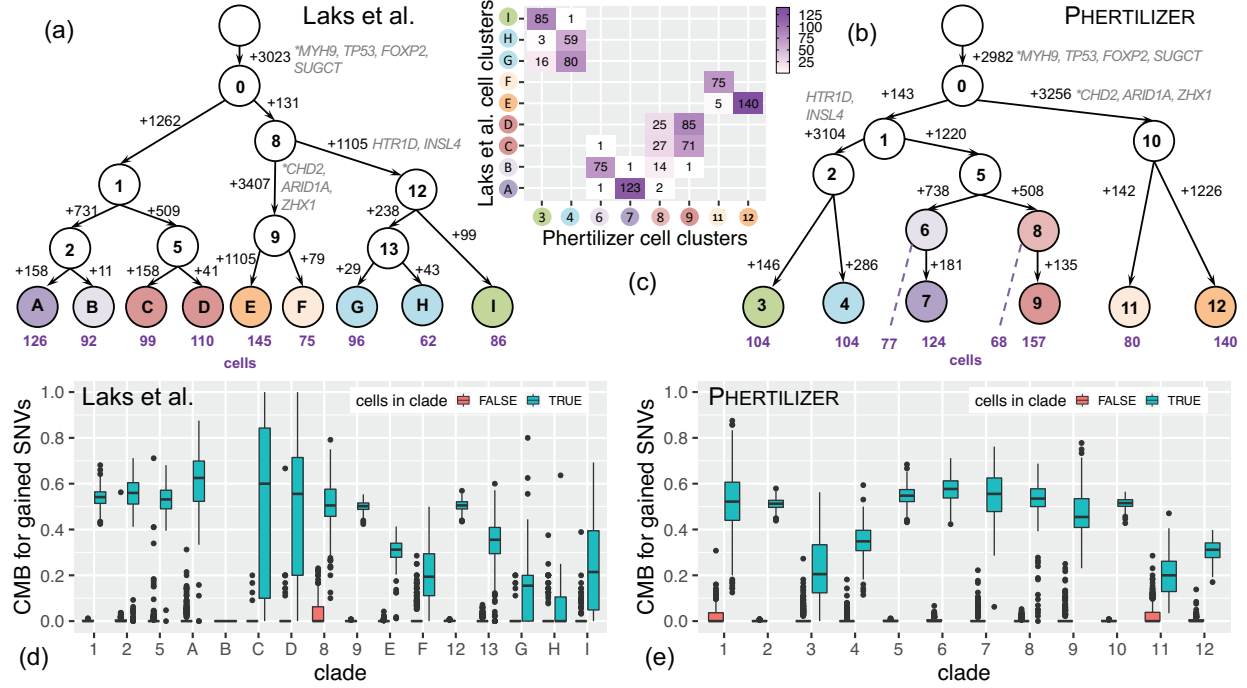


Figure 5.4: **PHERTILIZER improves upon the SNV phylogeny previously inferred by Laks et al.[24]** (a) The clonal tree inferred by Laks et al.[24] with edges labeled by SNV gains and cell numbers shown below leaf nodes. (b) The clonal tree inferred by PHERILIZER with edges labeled by SNV gains and cell numbers shown below leaf nodes. Cancer-related genes are labeled next to the SNVs in (a) and (b) with ‘*’ indicating a stop-gain variant. (c) Mapping between the Laks et al. [24] cell placement and PHERILIZER cell placement. (d-e) Cell mutational burden (CMB) comparison between cells within (blue) and outside (red) of each clade in the inferred Laks et al.[24] and PHERILIZER clonal tree.

A, D for $m = 14,068$ SNVs and derived binned read counts **R** for $b = 6,207$ bins (bin width of 500 KB) from data reported by Laks et al. [24]. The average sequencing coverage for these data was $0.25\times$. Utilizing an approach similar to the BASELINE method described above, Laks et al. [24] identified 9 copy number clones (labeled *A-I*) via dimensionality reduction and density-based clustering and reconciled them in a phylogeny with the copy number clones as leaves (Fig 5.4a). We annotated inferred trees with mutations in cancer-related genes in the cBioPortal [151, 152] and Cancer Gene Census (CGC) [153] from COSMIC v97 (see Appendix C.2.2).

As shown in Fig 5.4b, PHERILIZER inferred a clonal tree with 13 nodes and 8 clones. We found that PHERILIZER’s tree closely aligned with the Laks et al. [24] tree, with both approaches correctly identifying three major clades corresponding to the three distinct cell lines of origin (Fig 5.4c). Additionally, driver genes *TP53*, *SUGCT*, and *MYH9* are identified as clonal in both the Laks et.al [24] inferred tree (Fig 5.4a) and the PHERILIZER clonal tree

(Fig 5.4b). Similarly, subclonal SNVs in *CHD2*, *ARID1A*, *ZHX1*, *HTR1D*, and *INSL4* were placed in the corresponding clades of both trees. To further assess the quality of each inferred clade, we developed a performance metric called *cell mutational burden* (CMB) defined as $\text{CMB}(i, M) = \sum_{q \in M} \mathbf{1}\{a_{iq} > 0\} / \sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}$.

In words, $\text{CMB}(i, M)$ is the fraction of mapped SNV loci M with mapped variant reads in cell i . For a specified *clade* j or subtree rooted at node v_j , SNVs M_j are the SNVs gained at node v_j . For a cell i placed within clade j , we expect $\text{CMB}(i, M_j)$ to be high, although the value will depend on copy number. By contrast, for cells placed outside of clade j , we expect $\text{CMB}(i, M_j)$ to be low. More details on CMB are provided in Appendix C.2.2.

Fig 5.4d,e depicts the comparison of the distributions of CMB for all clades for cells placed within and outside that clade for Laks et al. [24] and PHERTILIZER, respectively. For cells placed outside of a specified clade, the reported tree by Laks et al. [24] as well as the tree inferred by PHERTILIZER have a median CMB of 0 for all clades. This is indicative of high specificity of both methods, i.e., SNVs are not assigned to a clade if there are observations of that SNV outside that clade. However, for cells placed within the clades, we observed greater variability for the Laks et al. [24] inferred clades than PHERTILIZER. This variability was most pronounced for the leaf nodes, especially *C*, *D*, *G*, *F*, *H* and *I*, where most have a small number of SNVs gained. We further analyzed clusters *G* and *H*, where the 25th percentile of the CMB for cells within the clade equals 0, and clusters *C* and *D* that had large variability (IQR of 0.74 and 0.51, respectively).

The location of clusters *G* and *H* in the embedding space suggests overfitting during density-based clustering, with an arbitrary split of a larger cohesive cluster containing *G* and *H* (Fig. C.17). In comparison, PHERTILIZER uses both copy number and SNV signal, resulting in the Laks et al. [24] *G* and *H* cells being clustered together to node 4 in the inferred PHERTILIZER clonal tree. Comparing CMB distributions for cells within clades *G* and *H* (Fig 5.4d) with PHERTILIZER's inferred clade 4 (Fig 5.4e), we observed a higher 25th percentile (0.31) for node 4 than for *G* (0.00) and *H* (0.00). This resulted in large separation between the CMB distributions of cells within and outside of clade 4 for PHERTILIZER but not for Laks et al. [24] *G* and *H* clades, implying better SNV placement in PHERTILIZER's clonal tree.

The last major difference of note between the two inferred trees is the clustering of the cells in Laks et al. [24] nodes *C* and *D* versus PHERTILIZER's nodes 8 and 9. Similarly to nodes *G* and *H*, we did not observe a clear separation of *C* and *D* in the embedding space (Fig. C.17), making it difficult to define clusters of these cells without more rigorous copy number profiling and clustering. However, as we saw in the simulation study, PHERTILIZER was able to detect further SNV evolution within a set of cells having the same copy number

profile. Cells in these clusters would not be split in half in the embedding space but instead should be scattered randomly throughout a single cluster in the embedding space. In addition to having observed cells in nodes 8 and 9 randomly scattered in a cluster in the embedding space (Fig. C.17), we also observed a clear separation between the CMB distributions for cells placed within and outside of clades 8 and 9 (Fig 5.4e). For these clades, we also observed low variability with IQRs of 0.08 and 0.13, respectively, whereas clusters *C* and *D* have very high variability in the inferred Laks et al.[24] clonal tree.

Overall, both inferred clonal trees are very similar but since PHERTILIZER simultaneously uses both CNA and SNV information, we obtained a slight improvement in terms of SNV phylogeny inference. Additionally, we note that the small number of SNVs gained at most of the leaf nodes is a direct result of the bottom up approach taken by Laks et al. [24], which performed pseudobulk SNV calling individually on each cell cluster. When SNVs are present in multiple cell clusters but at low prevalence in each cluster, they may not pass filtering of current somatic SNV callers for a cell cluster. This gives the appearance that SNVs are unique to a single clone, when in actuality they are present in multiple clones but have not been called. When sequencing coverage is closer to the $0.01\times$ as opposed to the $0.25\times$ that we have with these data, correctly genotyping a cell cluster becomes more challenging. In contrast, the top-down approach of PHERTILIZER is better suited to detect SNVs present in multiple cell clusters at low prevalence.

5.3.3 Eight triple negative breast tumors

We applied PHERTILIZER to eight triple negative breast tumors sequenced via ACT [23], labeled TN1 to TN8. After dimensionality reduction of normalized and GC-bias corrected binned read counts $\mathbf{R}$, Minussi et al. [23] identified for each tumor two sets of cell clusters with varying granularity, denoted as superclones and subclones. To obtain the input set of SNVs for each patient, we performed SNV calling of a pseudobulk sample of pooled sequenced cells using MuTect2 in tumor-only mode [154]. See Appendix C.2.2 for further details on data processing. Table. C.1 displays the breakdown of each tumor in terms of the number n of cells, the number m of SNVs and average coverage g , and depicts the number of inferred clones by PHERTILIZER, Minussi et al. [23] superclones and subclones, as well as the number of inferred clones by SBMCClone. Note that these data have a markedly lower coverage (ranging from $0.017\times$ to $0.039\times$) than the DLP+ data ($0.25\times$). We additionally ran BASELINE+SCITE with the cell clusters fixed to the Minussi et al. [23] subclones but all instances except TN3 and TN5 timed out after 10 hours. However, the CMB distribution for the inferred trees for TN3 and TN5 (Fig. C.19) provides no evidence in support of these

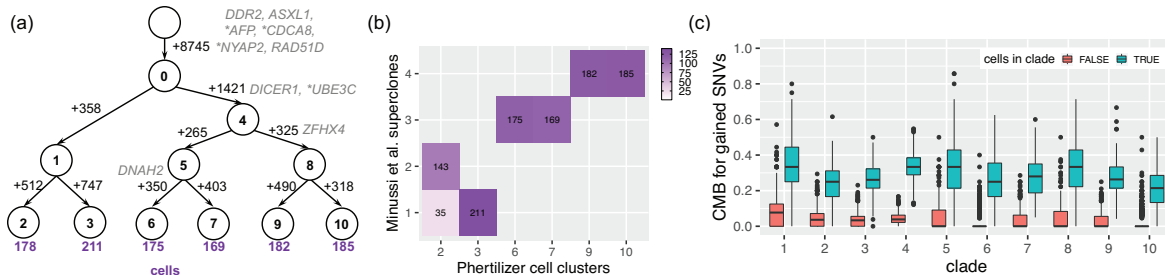


Figure 5.5: **PHERTILIZER infers clonal tree for breast cancer tumor TN1.** (a) The tree inferred by PHERTILIZER with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs (*): stop-gain variant). (b) A mapping between PHERTILIZER’s cell clusters and the Minussi et al. [23] superclones. (c) The cell mutational burden (CMB) comparison between cells within (blue) and outside of (red) each clade in the inferred clonal tree.

trees. SBMCClone only inferred a single clone for all patients, while PHERTILIZER inferred a tree with more than one clone for 4 out of the 8 tumors (TN1: 6, TN2: 6, TN4: 4 and TN8: 2). These four tumors have the highest average coverage of the eight patients. We will focus our discussion on the clonal trees inferred by PHERTILIZER for tumors TN1 (Fig 5.5) and TN2 (Fig. C.20).

For tumor TN1, PHERTILIZER inferred a branching tree with 11 nodes and 6 clones (Fig 5.5a). We also identified a subclonal missense SNV in driver gene *DICER1* associated with tumorigenesis and poor prognosis [155, 156]. We noted good concordance between the Minussi et al. [23] superclones and the PHERTILIZER cell clusters, with the exception of 35 cells that appear as outliers in superclone 1 (Fig 5.5b, Fig. C.18). This suggests these cells might fit better in superclone 2 based on SNV signal. In addition, we identified 8745 of the 13934 SNVs as truncal. This large truncal distance and branching structure were in alignment with the truncal distance in the clonal lineage tree inferred by Minussi et al. [23] using bulk whole exome sequencing. We used CMB to assess the performance of SNV and cell placement (Fig 5.5c). For clades 5 through 10, we observed that the median CMB for cells outside of the clade is 0. Clades 9 and 10 were particularly interesting because the embedding space depicts the occurrence of SNV evolution within Minussi et. al.’s [23] superclone 4 (Fig. C.18). For clades 2 through 4, we noted the median CMB for cells outside of the clade was around 0.035 for each of these clades, while clade 1 is the highest at 0.077. Upon further investigation of these 358 SNVs, we found that the median number of mapped reads was 5 when aggregating all cells, making these SNVs especially challenging to place. This drop in performance on the median CMB for cells outside of a clade when compared to the ovarian cancer patient analyzed above is expected due to the drop in sequencing cov-

erage from $0.25\times$ to $0.031\times$. However, we still observed a large separation between CMB distributions for the cells within the clade and cells outside the clade for all clades.

For tumor TN2, we inferred a branching clonal tree with 11 nodes and 6 clones (Fig. C.20). Two of PHERTILIZER’s cell clusters directly agreed with Minussi et al. [23] superclones. However, PHERTILIZER split the remaining two superclones into four cell clusters (3, 4, 5, 6) using SNV information. We observed low median CMB (0) for cells outside of the clade and a distinct separation between the cells within and outside of the clade distributions, providing evidence for this cell clustering and SNV placement. For tumor TN4, we identified an 8-node tree with five cell clusters, with trends similar to tumors TN1 and TN2 in terms of cell clustering concordance and CMB (Fig. C.21). Finally, for tumor TN8 at the lowest coverage $0.021\times$ of these four tumors, PHERTILIZER only inferred a 3-node branching tree with two cell clusters (Fig. C.22).

5.4 DISCUSSION

Ultra-low coverage SCDNA-SEQ has greatly enhanced our ability to study tumor evolution from a copy number perspective [23, 139]. Capitalizing on the strong copy number signal inherent in this data, we proposed a new method, PHERTILIZER, that grows an SNV phylogeny in a recursive fashion using elementary tree operations. We demonstrated the effectiveness of our approach relative to existing clustering methods on both simulated and real data. Importantly, we found that for the current number of cells (800 – 2000) used in practice, PHERTILIZER performs markedly better than these methods, yielding more accurate clonal trees, cell clusters, and clonal genotypes. As the first method to reconstruct the evolutionary history of SNVs from ultra-low coverage SCDNA-SEQ, PHERTILIZER helps advance the study of tumor evolution and makes progress towards the goal of joint SNV and CNA phylogeny inference at single-cell resolution.

There are several additional limitations and directions for future research. First, as sequencing coverage drops below $0.02\times$ as in the ACT data, PHERTILIZER does not infer clonal trees with more than one clone. Although inference is impacted by numerous factors, like copy number profile, it does perhaps suggest a coverage of approximately $0.02\times$ as the limit of detection for PHERTILIZER. Second, accurate variant calling also remains an open problem for ultra-low SCDNA-SEQ data, making it challenging to identify input subclonal variants. Beyond SCDNA-SEQ data, accurate SNV variant calling from scRNA-seq and ATAC-seq datasets is also challenging [157, 158]. But new methods, such as Monopogen [159], SComatic [160], VarCA [161], scAllele [162], reference-based methods [163] and the use of patient derived cell lines [164] are rapidly improving our ability to accurately conduct single-cell so-

matic mutational profiling from diverse technologies beyond SCDNA-SEQ. By accounting for different error profiles in variant calling, PHERILIZER could be extended to directly model the evolution of somatic variants from scRNA-seq and ATAC-seq datasets. Third, beyond SNVs and CNAs we plan to support structural variants and integrate other omics modalities such as methylation and transcription [165]. Fourth, the infinite sites model used in this work is often violated due to copy-number deletions. While we demonstrated robustness to such violations, a future direction is to use the Dollo evolutionary model [42, 150]. Finally, our model lacks an explicit placement of CNA events on the tree. Tree reconciliation methods, such as PACTION [166], can now be applied to integrate an SNV clonal tree generated by PHERILIZER and a CNA tree to obtain a joint tree.

CHAPTER 6: PHARMING: INFERRING A JOINT COPY NUMBER AND SNV CLONAL TREE FROM ULTRA-LOW COVERAGE SINGLE-CELL DNA SEQUENCING OF TUMORS

6.1 INTRODUCTION

As discussed in Chapter 6, high-throughput ultra-low coverage technologies, such as DLP+ [24] and ACT [23] are ideally suited for inferring copy number profiles due to their uniform coverage. But the sparsity of coverage makes it difficult to accurately infer the cellular profiles of point mutations, or single-nucleotide variants (SNVs) (Fig. 6.1a,b). The evolution of these two types of somatic alterations, copy number aberrations (CNAs) and SNVs are not independent (Fig. 6.1c,d). The dependencies between CNAs and SNVs has been exploited for bulk DNA sequencing data in order to validate inferred copy number profiles given a set of clonal SNVs [167]. With PHERILIZER (Chapter 6), we sought to exploit the dependencies between CNAs and SNVs indirectly by utilizing a low dimensional embedding of the binned read counts as a surrogate for copy number signal. While PHERILIZER was the first SNV tumor phylogeny inference for ultra-low coverage SCDNA-SEQ, it has two major shortcomings. First, PHERILIZER makes use of the infinite sites assumption and therefore does not allow SNV loss. However, copy number loss and copy neutral loss of heterozygosity (CN-LOH) is ubiquitous in cancer. Second, since the copy number information is compressed, PHERILIZER is not able to infer a joint CNA and SNV tree.

There are a number of methods that have considered these dependencies. For targeted sequencing and medium-to-high coverage SCDNA-SEQ, these methods include SCARLET [122], CONDoR [168], which COMPASS [169] and SCsnvcna [170]. In SCARLET, Satas et al. [122] infers an SNV phylogeny from a CNA tree which provides constraints on where losses can occur. However, this method first requires the accurate inference of a CNA tree. CONDoR [168] obviates the need for an input CNA tree by relying instead on a given clustering of cells as a proxy for copy-number profiles and proposes the constrained k -Dollo model that only allows SNV losses to occur when a shift in the cluster assignment of a node occurs. SCsnvcna [170] is a method to place SNVs on a CNA tree inferred from cells derived from an independent sequencing experiment, where sampling biases could occur. Additionally, SCsnvcna takes a binarized cell by SNV matrix as input and does not model read counts, which would better capture the interplay between an SNV and a CNA mutation. Lastly, COMPASS [169] infers a joint SNV and CNA phylogeny using maximum likelihood inference with MCMC but limits the inference of mutational events to be either SNV, gain, loss or CN-LOH and does not directly model the evolution of copy number profiles. Criti-

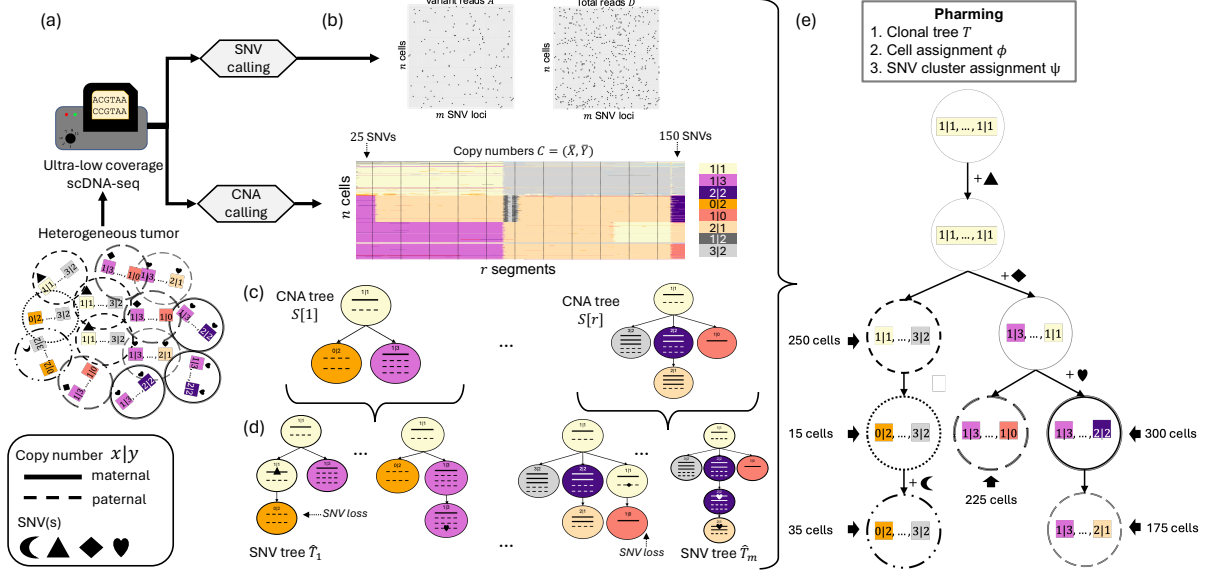


Figure 6.1: **PHARMING infer a clonal tree that jointly reconstructs the evolutionary history of CNAs and SNVs.** (a) A heterogeneous tumor comprising both CNAs and SNVs is sequenced with ultra-low coverage scDNA-SEQ technologies, such as DLP+ [24] and ACT [23]. (b) After SNV calling and CNA calling, the input to PHARMING is generated and consists of variant reads counts A , total read counts D for n cells and m SNV loci and segmented allele-specific copy number profiles $C = (\bar{X}, \bar{Y})$ for n cells and r segments. Additionally, the called SNVs are mapped to segment in which they occur. (c) The underlying evolutionary history of CNAs with respect to the individual genomic segments 1 through r of the heterogeneous tumor is represented by a CNA tree $S[\ell]$ for segment ℓ . (d) The underlying evolutionary history of individual SNVs, which co-occur in the identified segments is represented by an SNV tree $\hat{T}$, which relates the evolution of SNVs to CNAs. (e) The output of PHARMING is a clonal tree T , which models the shared evolutionary history of the heterogeneous tumor across all r segments, cell assignment ϕ , which assigns individual cells to nodes in the clonal tree and an SNV clustering ψ , that depicts a cluster of SNVs that are introduced along the same incoming edge of the clonal tree.

cally, these methods rely on either MCMC or integer linear programming formulations and do not scale to the number of cells and/or SNVs in present in ultra-low coverage datasets. Additionally, they are unable to handle the high missing data rate resulting from the sparsity of coverage.

In this work, we seek to overcome these shortcomings by developing a method named PHARMING that uses the interplay between CNAs and SNVs in order to yield a more comprehensive and accurate tumor phylogeny from ultra-low coverage scDNA-SEQ that need not adhere to the infinite sites model (Fig. 6.1e). The key idea is that reliable but noisy copy number profiles can be obtained from ultra-low coverage scDNA-SEQ [26, 171]. Therefore, PHARMING utilizes a divide and conquer approach to accomplish this challenging task. First,

symbol	domain	indices	meaning
n	$\mathbb{N}$	i, i'	the number of cells
m	$\mathbb{N}$	j, j'	the number of SNVs
r	$\mathbb{N}$	ℓ, ℓ'	the number of segments
k	$\mathbb{N}$	q, q'	the number of SNV clusters
$V(T)$	$\mathbb{N}$	u, v	the set of clones/nodes in tree T
$E(T)$	$\mathbb{N}$	(u, v)	the set of edges in tree T
x, y	$V(T) \times [k] \rightarrow \mathbb{N}$	$x(v, s)$	inferred allele-specific copy number of segment s for clone v
w, z	$V(T) \times [m] \rightarrow \mathbb{N}$	$w(v, j)$	inferred allele-specific mutation number of SNV j for clone v
(x, y, w, z)	$\mathbb{N}^4$	$g(v, j)$	inferred genotype for node v at SNV j
g	$V(T) \times [m] \rightarrow \mathbb{N}^4$	$g(v, j)$	inferred genotypes for clonal tree T
ϕ	$[n] \rightarrow V(T)$	$\phi(i)$	cell assignments
ϕ^{-1}	$V(T) \rightarrow 2^{[n]}$	$\phi^{-1}(v)$	mapping of nodes to a subset of cells
ψ	$[m] \rightarrow V(T)$	$\psi(j)$	mapping of SNV j to cluster q
ω	$[k] \rightarrow V(T)$	$\omega(q)$	mapping of SNV cluster q to node v
$C = (\bar{X}, \bar{Y})$	$(\mathbb{N}^2)^{n \times r}$	$x_{i,\ell}, y_{i,\ell}$	input allele-specific copy numbers
A	$\mathbb{N}^{n \times m}$	$a_{i,j}$	input variant read counts
D	$\mathbb{N}^{n \times m}$	$d_{i,j}$	input total read counts
σ	$[m] \rightarrow [r]$	$\sigma(j)$	SNV to segment mapping
σ^{-1}	$[r] \rightarrow 2^{[m]}$	$\sigma^{-1}(\ell)$	segment to SNV subset mapping

Table 6.1: **Summary of notation.**

PHARMING infers a set of candidate trees for each specified genomic segment or region. By focusing on a single segment, PHARMING is able to infer the gain and loss of SNVs with respect to the small set of copy number states that occurs with that segment. Then, then we use the joint cellular profiles across all segments to find an integrated tumor phylogeny that best fits the observed SCDNA-SEQ data.

6.2 MATERIALS AND METHODS

Suppose we have sequenced n cells and identified m SNVs from ultra-low coverage SCDNA-SEQ data. From this, we obtain the variant $A = [a_{i,j}]$ and total $D = [d_{i,j}]$ read counts for each cell $i \in [n]$, where $[n] = \{1, \dots, n\}$, and SNV $j \in [m]$. Using existing allele-specific copy number callers, such as DeepCopy [171], we obtain allele-specific copy number profiles $C = (\bar{X}, \bar{Y}) \in (\mathbb{N} \times \mathbb{N})^{n \times r}$, where r is the number of segments inferred during the genome segmentation by the copy number caller [171]. In words, segmentation ensures the copy number call (x_ℓ, y_ℓ) for an individual cell i is constant throughout all loci contained within segment ℓ and consequently segments may have varying size. We define a mapping $\sigma : [m] \rightarrow [r]$ that indicates the segment $\ell = \sigma(j)$ where SNV j occurs. Similarly, the set $\sigma^{-1}(\ell) \subseteq [m]$ contains the SNVs located in segment ℓ . We use the tuple (C, A, D) as a shorthand to refer to the observed copy number profiles C as well as the variant read counts

A and total read counts D . As SNVs can be affected by CNAs, we model the *genotype* of an SNV in an individual cell by the tuple (x, y, w, z) , where (x, y) are the allele-specific copy numbers and w and z are the number of mutated maternal and paternal copies. We summarize key notation in Table 6.1.

The joint evolution of SNVs and CNAs is represented by a clonal tree. While parallel gains of an SNV are rare [172], SNV loss due to CNAs or LOH events are common. Therefore, we make use of the Dollo evolutionary model [150] for SNVs. For CNAs, we utilize the infinite alleles assumption [173]. Under the infinite alleles assumption, each copy number state within a segment is introduced exactly once in the clonal tree. While violations of this assumption do occur in practice, it reasonably holds for larger segments and makes the challenging problem of joint clonal tree inference more tractable.

Next, we give the formal definition of a clonal tree (Fig. 6.1a, Fig. 6.2a) under the specified evolutionary models for SNVs (Dollo) and CNAs (infinite alleles).

Definition 6.1. A rooted, node-labeled tree T is a *clonal tree* with genotypes $g = (x, y, w, z)$ for r segments, consisting of allele-specific copy numbers $x : V(T) \times [r] \rightarrow \mathbb{N}$ and $y : V(T) \times [r] \rightarrow \mathbb{N}$ and allele-specific mutation numbers $w : V(T) \times [m] \rightarrow \mathbb{N}$ and $z : V(T) \times [m] \rightarrow \mathbb{N}$, such that

- (i) for each SNV j , there is exactly one edge $(u, v) \in E(T)$ where the SNV was gained, i.e., where either $z(u, j) = 0$ and $z(v, j) = 1$, or $w(u, j) = 0$ and $w(v, j) = 1$,
- (ii) and for each SNV j and for each node v it holds that $w(v, j) \leq x(v, \ell)$ and $z(v, j) \leq y(v, \ell)$ where ℓ is the segment where SNV j is located, and
- (iii) there is exactly one edge $(u, v) \in E(T)$ where the allele-specific copy number $(\hat{x}, \hat{y})$ of SNV j is attained, i.e., where $(x(u, j), y(u, j)) \neq (\hat{x}, \hat{y})$ and $(x(v, j), y(v, j)) = (\hat{x}, \hat{y})$.

The variant allele frequency (VAF) of an SNV j , i.e., the ratio of mutated to total copies, is a particularly informative for modeling observed variant read counts conditioned on its genotype $g(v, j)$ at node v . The formal definition is provided below.

Definition 6.2. Given a clonal tree T with genotypes g the *latent variant allele frequency (VAF)* of an SNV j in segment ℓ of node v equals

$$f(v, j) = \frac{w(v, j) + z(v, j)}{x(v, \ell) + y(v, \ell)}. \quad (6.1)$$

We also define a cell assignment $\phi : [n] \rightarrow V(T)$ that specifies the assignment of each cell i to a node v in clonal tree T . Given a clonal tree T and cell assignment ϕ , we derive a cost

function, comprised of two parts, to assess goodness of fit with respect to the observed data (C, A, D) . First, we define the likelihood for the observed variant read counts A given total read counts D , clonal tree T and cell assignment ϕ under a binomial model that accounts for the probability ϵ of misreading a base during sequencing.

$$P(a_{i,j} \mid d_{i,j}, T, \phi, \epsilon) = P(a_{i,j} \mid d_{i,j}, f(\phi(i), j), \epsilon) \quad (6.2)$$

$$= \binom{d_{i,j}}{a_{i,j}} p^{a_{i,j}} (1-p)^{d_{i,j}-a_{i,j}}, \quad (6.3)$$

where probability

$$p = f(\phi(i), j)(1-\epsilon) + (1-f(\phi(i), j))\frac{\epsilon}{3}, \quad (6.4)$$

is a function of the latent VAF $f(\phi(i), j)$ of SNV j in cell i and ϵ .

Second, while allele-specific copy number calling methods are rapidly improving, they are not error-free. As a result, we permit some deviation between the observed $(\bar{x}_{i,\ell}, \bar{y}_{i,\ell})$ and inferred $(x(\phi(i), \ell), y(\phi(i), \ell))$ allele-specific copy numbers. We consider a penalty term $\chi(i, \ell)$ that accounts for deviations for each cell i and segment ℓ as

$$\chi(i, \ell) = |x(\phi(i), \ell) - \bar{x}_{i,\ell}| + |y(\phi(i), \ell) - \bar{y}_{i,\ell}|. \quad (6.5)$$

The level of tolerance for deviation is controlled by parameter $\lambda \in \mathbb{R}_{\geq 0}$. Assuming independence between cells, segments and SNVs, we arrive at the following cost function for a given clonal tree T and cell assignment ϕ

$$J(T, \phi) = - \prod_{i \in [n]} \prod_{j \in [m]} P(a_{i,j} \mid d_{i,j}, T, \phi, \epsilon) + \lambda \sum_{i \in [n]} \sum_{\ell \in [r]} \chi(i, \ell), \quad (6.6)$$

This leads to the following problem.

Problem 6.1 (CLONAL TREE INFERENCE WITH COPY NUMBER (CTICN)). Given allele-specific copy-numbers $C = (\bar{X}, \bar{Y})$, variant read counts A , total read counts D and parameters $\lambda \in \mathbb{R}_{\geq 0}$ and $\epsilon \in [0, 1]$, find a clonal tree T with genotypes g and cell assignment ϕ such that cost function $J(T, \phi)$ is minimum.

6.2.1 The PHARMING algorithm

To solve the CTICN problem, we rely on three key insights. Firstly, similar to PHERTILIZER (Chapter 6) [69], we expect the SNVs to cluster into a small number k of clusters where

$k \ll m$. We define $\psi : [m] \rightarrow [k]$ to be a mapping of an SNV to one of k SNV clusters.

Secondly, SNVs in the same SNV cluster must share a descendant cell fraction (DCF) [122]. The DCF of an SNV is defined as the fraction of cells that are assigned to or are descendants of the node where an SNV was introduced. By construction, the DCFs account for cells assigned to nodes even where an SNV may have been lost. We define a mapping $\delta : [k] \rightarrow [0, 1]$ that yields the DCF $\delta(q)$ of an SNV cluster q . Although DCFs were originally intended to facilitate SNV clustering for bulk data, we infer them from a pseudobulk sample. Critically, the DCFs δ constrain the evolutionary relationships between SNV clusters and provide a means to significantly constrain the underlying space of clonal trees [127].

Thirdly, the individual problems of inferring a clonal tree individually for CNA or SNVs are both challenging. However, SNVs commonly co-occur in segments impacted by CNAs, yielding a valuable signal for the evolutionary relationships between these events. This allows us to utilize a divide and conquer approach to simplify the problem. First, we infer a set of candidate clonal trees for each segment. Then, we integrate these clonal trees across segments using both the cellular profiles and the DCFs.

Utilizing these three key ideas, we developed a three-phased algorithm named PHARMING, a divide and conquer algorithm to solve the CTICN problem. We describe this approach for a fixed number k of SNV clusters, however a sweep across reasonable values of this parameter in conjunction with model selection (Sec. 6.2.2) is used to find a heuristic solution to the CTICN problem. In the following, we define some additional terminology and notation helpful for describing the PHARMING algorithm (Sec. 6.2.1) and then we describe its three main phases: (i) initialization (Sec. 6.2.1), (ii) clonal tree inference for a single segment (Sec. 6.2.1) and (iii) integration of clonal trees (Sec. 6.2.1). We conclude with details on performing model selection (Sec. 6.2.2) for varying numbers of SNV clusters.

Preliminaries. Since PHARMING is a divide and conquer algorithm, it makes use of a variety of tree structures (Fig 6.2), some of which are special types of clonal trees, that efficiently capture the complex evolutionary history of both SNVs and CNAs in a heterogeneous tumor. We start by formally defining these structures and terminology that defines their relationships to one another in the inferred clonal tree T .

Next by defining the key building blocks of a clonal tree, a CNA tree S and an SNV tree $\hat{T}$. Informally, each segment ℓ of a clonal tree has an underlying CNA tree S that describes the evolution of the copy number states within segment S (Fig. 6.1c, Fig. 6.1d).

Definition 6.3. A rooted, node-labeled tree S is a *CNA tree* with allele-specific copy number states $\mu = (x, y)$ and root state (x_0, y_0) provided (i) the root v_0 of S has copy number state

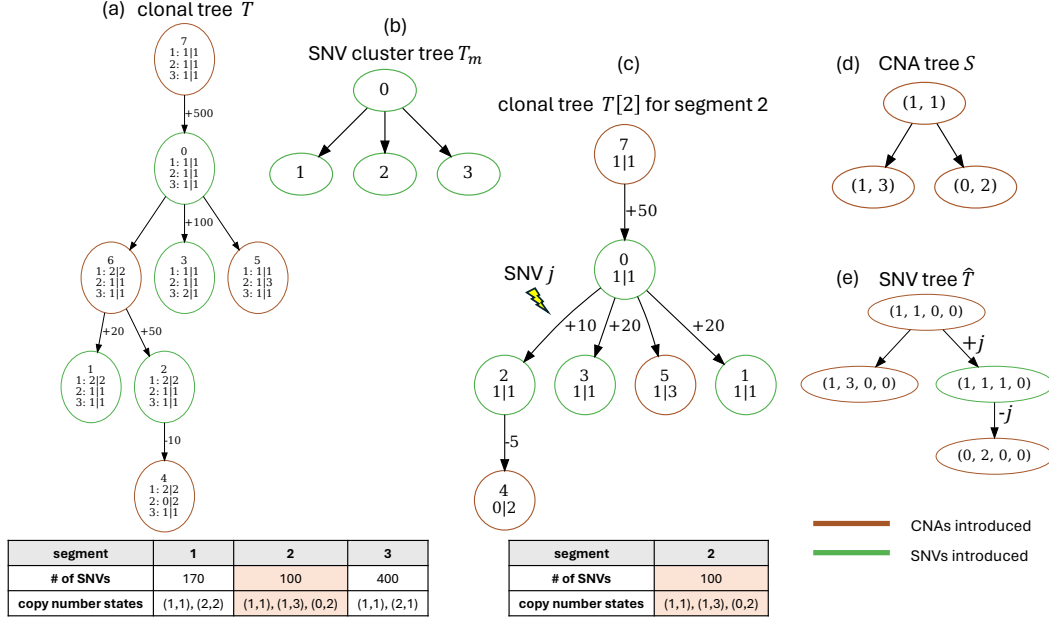


Figure 6.2: **Depiction of tree structures for PHARMING.** (a) A PHARMING clonal tree T for $r = 3$ segments with varying number of SNVs per segment and sets of copy number states. Newly introduced or lost SNVs are shown on the incoming edges. The copy number state for each node and each segment is labeled as segment: $x|y$. The root, node 7, is the normal clone. (b) Clonal tree T with $k = 4$ SNV clusters is a refinement of SNV cluster tree T_m . The nodes labeled 0 to 4 in clonal tree T correspond to the SNV clusters depicted in SNV cluster tree T_m . (c) The underlying clonal tree $T[2]$ for segment 2 corresponding to clonal tree T . Clonal tree T is also a refinement of clonal tree $T[2]$. (d) The underlying CNA tree S for clonal tree $T[2]$. Clonal tree $T[2]$ will be a refinement of CNA tree S . (e) The underlying SNV tree $\hat{T}$ for SNV j , with SNV cluster assignment $\psi(j) = 2$. SNV tree $\hat{T}$ is a refinement of CNA tree S .

$\mu(v_0) = (x_0, y_0)$ and there exists exactly one edge $(u, v) \in E(S)$ where copy-number state $(\hat{x}, \hat{y})$ is attained, i.e., $\mu(u) \neq (\hat{x}, \hat{y})$ and $\mu(v) = (\hat{x}, \hat{y})$.

To avoid confusion with shorthand to refer to individual components of a genotype $g(u, j)$ for a node u and SNV j , i.e., $x(u, j)$, we use the notation $\mu_x(u)$ to refer to the value of x , i.e., the number of maternal alleles, at node u in CNA tree S .

Similarly to a CNA tree, each SNV j that occurs within segment ℓ of a clonal tree has an underlying SNV tree that depicts the evolutionary history of a single SNV j within clonal tree T (Fig. 6.2e).

Definition 6.4. A rooted node-labeled tree $\hat{T}$ is an *SNV tree* with genotypes $\hat{g} = (x, y, w, z)$, $g : V(\hat{T}) \rightarrow \mathbb{N}^4$, for a specified root genotype $g_0 = (x_0, y_0, 0, 0)$ provided (i) the root v_0 of $\hat{T}$ has genotype $\hat{g}(v_0) = (x_0, y_0, 0, 0)$ and (ii) it adheres to the single-split copy number

assumption [174], i.e., there is exactly one copy number (x^*, y^*) with two distinct genotypes $\hat{g}(u), \hat{g}(v)$ having copy number (x^*, y^*) and mutation states $\max(w(u), z(u)) = 0$ and $\max(w(v), z(v)) > 0$.

Note that an SNV tree can be thought of as a clonal tree for a single but generic SNV. That is, the genotypes $\hat{g}(v)$ for SNV tree $\hat{T}$ are not indexed by an SNV as they are for the genotypes $g(v, j)$ for a clonal tree T . However, like the definition of clonal tree, an SNV tree adheres to the Dollo model for SNVs and the infinite alleles assumption for copy numbers.

If we consider segment ℓ and an SNV j that occurs in segment ℓ , intuitively the evolution of copy number states in an SNV tree $\hat{T}$ that describes the evolution of SNV j and the evolution of copy number states in CNA tree S for segment ℓ must be consistent. We formalize this intuitive notation below.

Definition 6.5. An SNV tree $\hat{T}$ is a *refinement* of a CNA tree S if for each edge $(u, v) \in E(S)$, there is exactly one edge $(u', v') \in E(\hat{T})$ where $\mu_x(u) = x(u'), \mu_y(u) = y(u'), \mu_x(v) = x(v')$ and $\mu_y(v) = y(v')$.

We also introduce an important tree structure for our algorithm referred to as a SNV cluster tree (Fig. 6.1b). Recall the key idea is that the SNVs should cluster in small number k of SNV clusters.

Definition 6.6. A rooted tree T_m is an *SNV cluster tree* provided T_m has exactly k nodes where each node represents an SNV cluster q .

A tree $T[\ell]$ or $T[L]$ is a clonal tree for segment ℓ or segments $L \subseteq [r]$ such that the genotypes $g(v, j)$ are restricted to the set $\sigma^{-1}(\ell) \subseteq [m]$ of SNVs in located in segment ℓ (Fig. 6.2c). This same indexing $[\ell]$ is also used where appropriate for other tree structures.

Initialization. We begin by initializing the DCFs δ for a fixed value of k . These DCFs may be obtained via an existing algorithm called DECiFER [174] on pseudobulk sample generated from merging the SCDNA-SEQ data. Briefly, starting from a random initialization of k DCFs values, DECiFER uses a coordinate ascent approach to alternate between (i) optimizing an assignment of each SNV j to a cluster q and an SNV tree $\hat{T}$ given the DCF cluster centers δ and (ii) optimizing the DCF cluster centers δ given a cluster assignment q and SNV tree $\hat{T}$ for each SNV j . One shortcoming of DECiFER is that it does not consider the location of SNVs and therefore may imply inconsistent CNA trees for SNVs that co-occur in the same segment. To remedy this, we also provide a modified implementation of DECiFER called DCF Clustering that enforces consistent underlying CNA trees for a segment. See Appendix D for additional details.

Given the inferred DCFs δ for k clusters, we next enumerate a set $\mathcal{T}_m$ of valid SNV cluster trees. The formal definition of valid is defined below.

Definition 6.7. A SNV cluster tree T_m is *valid* for DCFs δ if it holds that for each node $u \in V(T_m)$, the sum condition is met [127], i.e., $\delta(u) \geq \sum_{v \in \text{children}(u)} \delta(v)$, where $\text{children}(u)$ is the set of children of node u in T_m .

Additional details on enumeration of the set $\mathcal{T}_m$ of valid SNV cluster trees are available in El-Kebir et al. [127]. Starting from the initial DCFs δ and the set $\mathcal{T}_m$ of SNV cluster trees, we next infer a set $\mathcal{T}[\ell]$ of candidate clonal trees for each segment ℓ .

Clonal tree inference for a single segment. The problem of inferring tree $T[\ell]$ is a special case of the CITCN problem where only a single segment ℓ is considered. We focus our discussion on a fixed segment ℓ and a fixed SNV cluster tree T_m . Recall from Sec. 6.2.1 that clonal tree $T[\ell]$ implies an underlying CNA tree S for segment ℓ . Another way to think of clonal tree $T[\ell]$ is that it specifies one possible refinement of CNA tree S and SNV cluster tree T_m [166]. Let $\omega(q)$ be a mapping of SNV cluster q to node v in clonal tree $T[\ell]$ indicating where in the clonal tree SNV cluster q is gained. Recall also that a clonal tree $T[\ell]$ implies an underlying SNV tree $\hat{T}$ for each SNV j in segment ℓ . Consider the set $\hat{\mathcal{T}}$ of SNV trees such that each SNV tree $\hat{T} \in \hat{\mathcal{T}}$ is a refinement of CNA tree S . Then a clonal tree $T[\ell]$ that is a refinement of CNA tree S and SNV cluster tree T_m implies a mapping $\rho : [k] \rightarrow 2^{|\hat{\mathcal{T}}|}$ of an SNV clusters q to a subset of $\hat{\mathcal{T}}$ SNV trees. In words $\rho[q]$ is the set of valid evolutionary histories for each SNV j that is assigned to SNV cluster $\psi(j)$ and introduced at node $\omega(\psi(j))$ in clonal tree $T[\ell]$. The power of this mapping ρ is that it greatly simplifies the problem of solving this special case of the CITCN. For a fixed refinement of CNA tree S and SNV cluster tree T_m , the problem reduces to finding an optimal SNV clustering ψ and an SNV tree assignment $\alpha(j) \in \rho(\psi(j))$ for each SNV j .

We accomplish this with a coordinate descent algorithm. We first initialize the clonal tree $T[\ell]$ by finding an SNV clustering ψ and SNV tree assignment α (See Appendix D.1.2 for details). Then we alternately (i) find a optimal cell assignment ϕ^* given $T[\ell]$ such that cost $J(T[\ell], \phi^*)$ is minimum and then find an optimal SNV clustering ψ^* and SNV tree assignment α^* given a fixed cell assignment ϕ such that cost $J(T[\ell]^*, \phi)$ is minimum. We perform this alternating optimization until the cost $J(T[\ell], \phi)$ converges or we reach a maximum number of iterations. See Appendix D.1.2 for more details on this coordinate descent algorithm.

We infer a clonal tree $T[\ell]$ for each CNA tree S in the enumerated set of all valid CNA trees for segment ℓ and for each valid refinement of CNA tree S and SNV cluster tree T_m .

We rank the inferred clonal trees using cost $J(T[\ell], \phi)$ for each mutation cluster T_m in order to prioritize trees for integration in the next phase of PHARMING.

Integrating clonal trees. Given two clonal trees T_1 and T_2 constructed from the same underlying SNV cluster tree T_m on disjoint sets L and L' of segments, the *integration* problem is to find a clonal tree T' and cell assignment ϕ such T' is a refinement of clonal trees T_1 and T_2 and $J(T', \phi)$ is minimum.

The key idea to solving the integration problem is that both clonal trees are in agreement with the same underlying SNV cluster tree T_m . Thus, we only need to resolve the evolutionary relationships of copy number events with respect to the SNV clusters. Conceptually, a node in a clonal tree T may indicate the gain of an SNV cluster or the introduction of a copy number state change in one or more segments. More formally, we say that node u in clonal tree T is a *copy number node* if there does not exist an SNV cluster q such that $\omega(q) = u$. The ordered pair (q, u) of an SNV cluster q and a copy number node u has one of three possible evolutionary relationships, $\omega(q)$ is ancestral to u , $\omega(q)$ is a descendant of u or $\omega(q)$ and u are incomparable, meaning they occur on distinct branches of the clonal tree. Therefore, we say clonal tree T' is a *refinement* of T_1 (T_2) if for each SNV cluster q and copy number node u in T' , the ordered pair (q, u) has the same evolutionary relationship, i.e., ancestral, descendant, incomparable, as the pair (q, v) , where v is the node in T_1 (T_2) indicating the introduction of the same copy number event(s). Adhering to these evolutionary constraints with each SNV cluster allows us to easily enumerate a set $\mathcal{T}'$ of clonal trees. These refinements are then ranked by cost $J(T', \phi')$ after finding the optimal cell assignment ϕ' for each clonal tree T' .

To integrate a set $\{T[1], \dots, T[r]\}$ of clonal trees and with the same underlying SNV cluster tree T_m , we progressively solve the integration problem. First, we obtain a random permutation π of the r segments and initialize the optimal clonal tree $T^* := T[\pi(1)]$ with the candidate clonal trees inferred for segment $\pi(1)$, i.e., $T[\pi(1)]$. Then, for each $p \in \{2, \dots, r\}$, we solve the *integration* problem for T^* and $T[\pi(p)]$. This progressive integration may be repeated for multiple random permutations of the segments.

In practice, we integrate a collection $\{\mathcal{T}[1], \dots, \mathcal{T}[r]\}$ of clonal trees. This is because a clonal tree $T[\ell]$ may be optimal when looking myopically a segment but may fit poorly into integrated tree T^* on many segments. Therefore, at each progressive integration step, we integrate $(T^*, T[\pi(p)]) \in \mathcal{T}^* \times \mathcal{T}[\pi(p)]$ and retain a user-specified number of the minimum cost clonal trees. After completing the user-specified number of progressive integrations, PHARMING will return the clonal tree $T^* \in \mathcal{T}^*$ and cell assignment ϕ^* where cost $J(T^*, \phi^*)$ is minimum.

6.2.2 Model selection

The key hyperparameter for PHARMING is the number k of mutation clusters in the underlying SNV cluster tree T_m . This parameter choice also influences the estimates of the input DCFs δ . To avoid overfitting the data, we propose a two-stage model selection process. In stage one, the DCF clustering algorithm of choice is run for wide range of reasonable values of k . Initial model selection utilizing standard approaches for clustering, such as the elbow method or silhouette score, are then applied to narrow down the range of k values and corresponding DCFs δ given to PHARMING for clonal tree inference. Set K denotes these prioritized choices for the number of mutation clusters used as a hyperparameter for PHARMING.

In the second stage, we make use of the Bayesian information criterion (BIC), i.e., $BIC(k)$, defined as a function of the number of SNV clusters k , the number of SNVs m and the likelihood of the variant reads A given total read counts D , a clonal tree T with k SNV clusters, cell assignment ϕ and a sequencing error probability ϵ .

$$BIC(k) = k \log m - 2 \log \prod_{i=1}^n \prod_{j=1}^m P(a_{i,j} \mid d_{i,j}, T, \phi, \epsilon) \quad (6.7)$$

$$= k \log m - 2 \sum_{i=1}^n \sum_{j=1}^m \log P(a_{i,j} \mid d_{i,j}, T, \phi, \epsilon) \quad (6.8)$$

PHARMING outputs the clonal tree T and cell assignment ϕ for the number $k^* = \operatorname{argmin}_{k \in K} BIC(k)$ of SNV clusters.

6.3 RESULTS

6.3.1 *In silico* experiments

To evaluate PHARMING, we generated *in silico* experiments with known ground truth clonal trees and cell clusterings. We set simulation parameters $n = 1000$ cells, $m = 5000$ SNVs, $r = 25$ segments, $k = 5$ mutation clusters, coverage varying in $\{0.01, 0.05, 0.25\}$ and the copy number error rate varying in $\{0, 0.035\}$ with 5 replicates for each coverage for a total of 30 experiments. For all experiments, the per base sequencing error rate ϵ was set to 0.001. Since in Chapter 6, we demonstrated the PHERILIZER outperforms all comparable methods for ultra-low coverage SCDNA-SEQ data, we only benchmarked PHARMING against PHERILIZER.

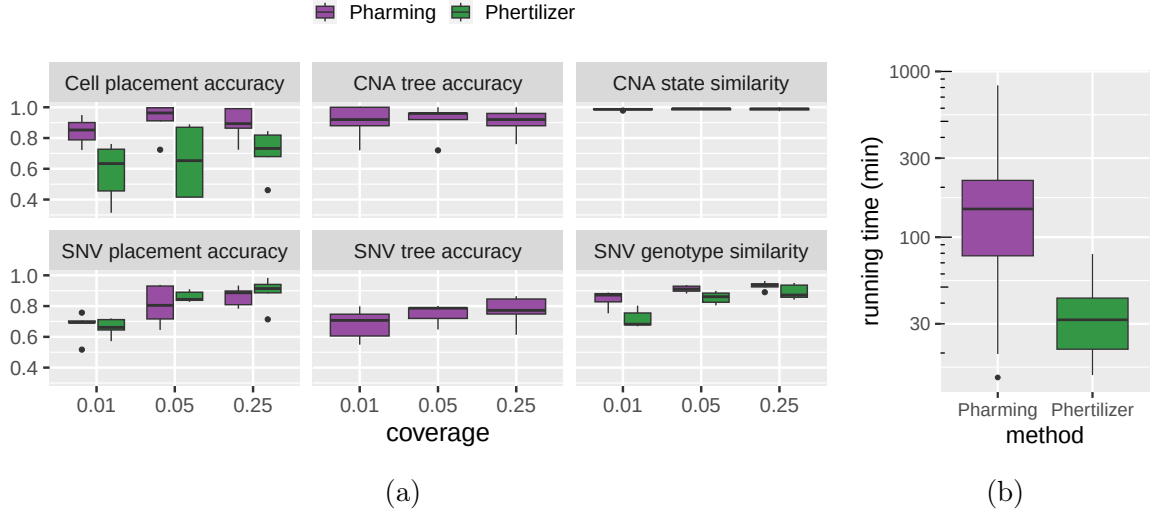


Figure 6.3: **PHARMING outperforms PHERTILIZER on *in silico* experiments.** (a) Comparison of PHARMING versus PHERTILIZER on *in silico* experiments with known ground truth clonal tree. PHERTILIZER is excluded from metrics that utilize inferred copy numbers. (b) Comparison of running time between PHARMING and PHERTILIZER.

PHERTILIZER was run with default parameters. For the binned read count embedding required by PHERTILIZER, we ran UMAP [175] on the copy number profiles $\bar{C}$. For PHARMING, we first ran our DCF clustering algorithm for values of $k \in \{3, \dots, 6\}$ to obtain estimated DCFs. Then, we ran PHARMING for each of the inferred DCFs at various values of k . During the integration stage, we maintained a set of the top 5 solutions and utilized only one random ordering for progressive integration. We utilized a penalty of $\lambda = 1000$ in order to compute our cost function. Finally, we used BIC as described above in order to select an inferred clonal tree T^* and cell assignment ϕ^* . To assess performance with respect to inference of SNVs, CNAs and cell assignment, we utilized a number of performance metrics. We focus our discussion on simulations with a copy number error rate of 0.035 as these instances are more reflective of real data (Fig. 6.3a).

As in Chapter 6, we computed the pairwise ancestral, incomparable and clustered recall for both cells and SNVs. Taking the weighted average of these recall metrics yields a *cell (SNV) placement accuracy* score. See Chapter 6 and Appendix 13 for additional details on these metrics. For cell placement accuracy, PHARMING (median: 0.90) significantly outperformed PHERTILIZER (median: 0.68) for all coverages. This is likely due to the fact that PHARMING utilizes the copy number information in a much more effective way than PHERTILIZER, which compresses it. Similarly to cell placement accuracy, PHARMING outperforms PHERTILIZER for all three recall metrics (Fig. D.1). For SNV placement accuracy, we observed

that PHARMING (median: 0.78) performed comparably to PHERTILIZER (median: 0.84) or slightly worse. Of the three recall metrics, PHARMING had the lowest performance for incomparable pair recall (Fig. D.2, suggesting an opportunity for further improvement.

We next assessed accuracy of the inferred CNA trees and SNV trees. *CNA tree accuracy* is the proportion of segments where the inferred CNA tree is identical to the ground truth CNA tree. Note that since PHERTILIZER does not infer CNAs, we exclude PHERTILIZER from analysis on CNA related metrics. Impressively, PHARMING is able to obtain a median CNA tree accuracy of 0.92, meaning for median of 23 out of 25 segments, PHARMING infers the correct underlying CNA tree. This demonstrates the CNA signal is strong enough to support our divide and conquer approach, even in coverages as low as $0.01\times$.

Similarly, *SNV tree accuracy* is the proportion of SNVs where the inferred underlying SNV tree matches ground truth modulo allele-specificity. In other words, when comparing SNV trees, we collapsed the four tuple genotypes (x, y, w, z) to a three tuple genotype $(x, y, \max(w, z))$. On SNV tree accuracy, PHARMING performed well with a median score of 0.75. Unlike CNA tree accuracy, we did observe a drop in accuracy as the coverage decreased. This is somewhat expected since SNVs introduced lower in the tree may only have a few variant reads, making it more difficult to distinguish signal from sequencing error.

Next, we assessed two similarity metrics, *CNA state similarity* and *SNV genotype similarity*, that simultaneously capture the accuracy of the clonal tree, genotypes and cell assignment. The *CNA state similarity* score equals 1 minus the normalized Hamming distance between ground truth copy number states and inferred copy number states of cells. On this metric, PHARMING obtained an impressive median 0.99 on all instances.

Similarly, the *SNV genotype similarity* score equals 1 minus the normalized Hamming distance between the ground truth number of mutated copies, i.e., $\max(w, z)$ and inferred number of mutated copies of cells. We included PHERTILIZER in this comparison since it is related to the the presence or absence of SNVs PHARMING (median: 0.90) did outperform PHERTILIZER (median 0.84) even though SNV placement accuracy was worse. Critically, we see the largest performance gap between PHARMING (median: 0.87) and PHERTILIZER (median: 0.68) at the lowest coverage of $0.01\times$.

Lastly, we compared the running time of PHARMING and PHERTILIZER (Fig. 6.3b). Overall, we found that PHERTILIZER was significantly faster (median: 31.7 minutes) than PHARMING (median: 148 minutes). This is somewhat expected as PHARMING is solving a significantly more complicated inference problem than PHERTILIZER. But the scalability of PHARMING remains a challenge, especially as the the number k of SNV clusters and the number r of segments increases.

Combined, these performance metrics highlight the capability of PHARMING to infer an

cell line	n cells	m SNVs (med. per segment)	m' SNVs sampled (med. per segment)
SA921	244	6243 (12)	814 (27)
SA922	220	7143 (13)	865 (30)
SA1090	426	5408 (10)	865 (29)

Table 6.2: **Summary of the ovarian cancer cell line dataset.**

accurate joint CNA and SNV clonal tree, even at a coverage as low as $0.01\times$. We observe similar trends and performance for instances with a copy number error rate of 0 (Fig. D.3), suggesting that PHARMING is robust to error in the inference of the copy number profiles.

6.3.2 Ovarian cancer cell lines

We assessed the performance of PHARMING on a dataset comprised of three clonally related ovarian cancer cell lines sequenced with DLP+ [24]. This dataset contained 890 cells and 13628 SNVs. We removed 200 SNVs that were suspected sequencing artifacts since they appeared to be subclonal in two or more of the cell lines and therefore in violated the Dollo evolutionary model for SNVs. This resulted in a total of 13428 SNVs for analysis. We utilized DeepCopy [171] for allele-specific copy number calling and genome segmentation, resulting in 409 segments with a median of 26 SNVs per segment. Only two segments do not have any mapped SNVs.

To demonstrate the features of our algorithm while overcoming scalability challenges, we opted to run PHARMING individually for each of the three cell lines. For each cell line, we first filtered out SNVs that did not have at least two variant reads across all cells in the sample. Next, we ran DECIPHER [174] independently on each cell line for values of $k \in \{3, \dots, 6\}$. After this, we randomly sampled $r = 25$ segments from each cell line among segments that had a minimum of 20 SNVs per segment— see Table 6.2 for a summary of the data. Finally, we ran PHARMING on each cell line for each value of k , using BIC to select the best value of k and corresponding PHARMING clonal tree and cell assignment (Fig 6.4e).

We focus our discussion on cell line SA922, which had $n = 220$ cells and $m' = 865$ SNVs after filtering. PHARMING inferred a clonal tree with $k = 5$ SNV clusters and 8 clones, i.e., nodes that contained more than 5 mapped cells. Three of the $k = 5$ SNV clusters were introduced on the trunk of the PHARMING clonal tree (nodes 10, 9 and 1). The two remaining SNV clusters, nodes 8 and 0 were subclonal containing 136 and 16 newly introduced SNVs respectively (Fig. 6.4e). We utilized our cell mutational burden (CMB) metric introduced in Chapter 6 for each of $k = 5$ nodes where SNVs were introduced to assess the placement of SNVs in our clonal tree — see Appendix 13 and Fig. C.16 for more details on this metric.

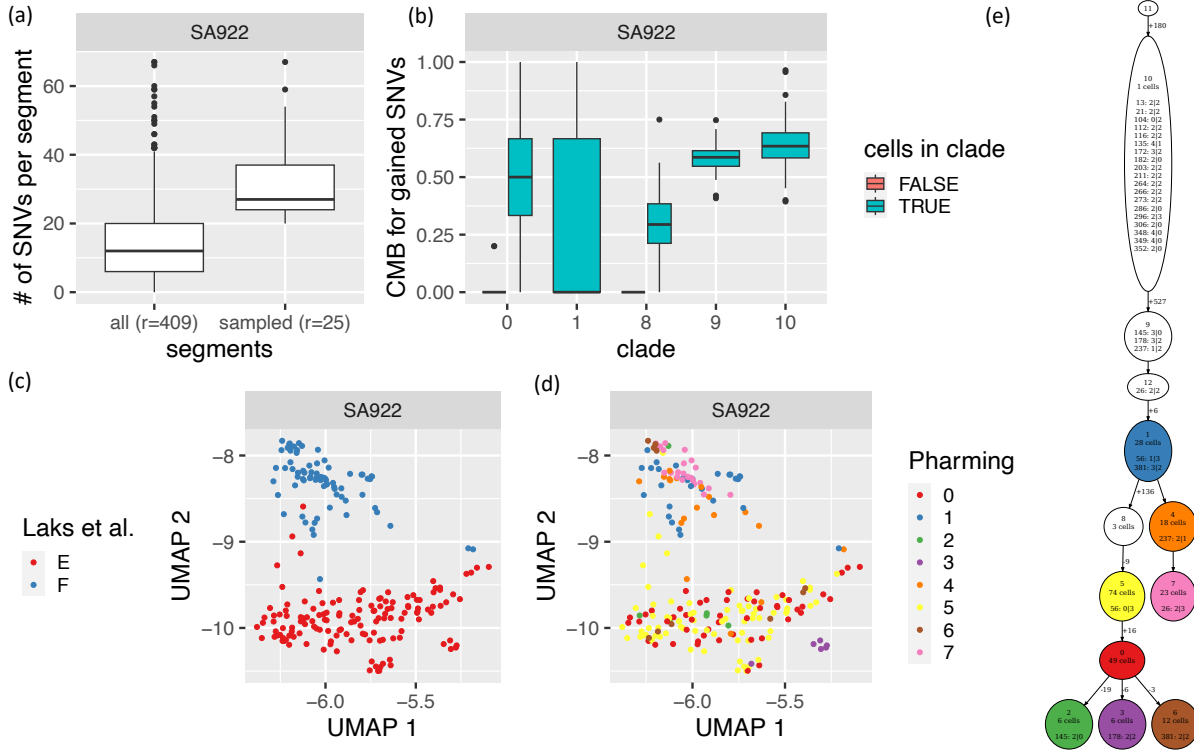


Figure 6.4: **PHARMING** analysis of the SA922 ovarian cancer cell lines sequenced with DLP+ [24]. (a) Distributions of the number of SNVs in all segments ($r = 409$) and the number of SNVs in the sampled segments ($r = 25$). (b) Distribution of the cell mutational burden (CMB) for the clades in inferred PHARMING clonal tree. (c) UMAP projection of the cells in SA922 colored by Laks et al. [24] cluster. (d) UMAP projection of the cells in SA922 colored by PHARMING cell assignment ϕ . (e) PHARMING inferred clonal tree for the 25 sampled segments of SA922. Nodes with more 5 cells assigned are colored to be consistent with (d) and labeled by the number of assigned cells as well as the introduction of a copy number state change, i.e., segment: $x|y$. The incoming edges are labeled by the number of newly introduced SNVs and the number of lost of SNVs when applicable.

Briefly, $\text{CMB}(i, M)$ is the fraction of mapped SNV loci M with mapped variant reads in cell i . For a specified *clade* q or subtree rooted at node $\omega(q)$, SNVs M_q are the SNVs gained at node $\omega(q)$ and never lost. For the three truncal clusters (nodes 1, 9 and 10), we were only able to assess the CMB for cells in the clade since these SNVs were present in almost all placed cells, except one. For clades 9 (median: 0.59, 527 SNVs) and 10 (median: 0.63, 180), we observed a high CMB for cells in the clade, with low variance, of this metric, providing supporting evidence that these SNVs are indeed truncal. For Clade 9, which only has 6 newly introduced SNVs, we observed a low CMB (median: 0), with high variance, suggesting slight overfitting in regards to the number k of SNV clusters. For the two subclonal clusters, we observed low CMB (node 0 median: 0, node 8 median 0) for cells outside of the clade and

a high CMB for cells within the clade (node 0 median: 0.5, node 8 median: 0.29), lending credence to the placement of these SNVs in our PHARMING clonal tree.

Interestingly, cell lines SA922 and SA921 were both previously hypothesized to have undergone a whole genome duplication [24, 27] and this is indeed reflected in the inferred PHARMING trees (Fig. 6.4e and Fig. D.5). In particular, we see that 19 of the 25 sampled segments underwent a copy number state change after the introduction of the first SNV truncal cluster, with 8 segments transitioning to copy number state (2, 2), 14 segments having 4 or more total copies and 4 segments indicating a copy neutral LOH event of either (2, 0) or (0, 2).

Next, we plotted the UMAP coordinates obtained from the binned read count embedding of this same dataset in Chapter 6, with projected cells colored by the Laks et al. [24] clustering (Fig. 6.4c) and by the Pharming clone assignment (Fig. 6.4d). Laks et al. [24] previously inferred two distinct clusters for these cells using a density based clustering approach. However, there are two drawbacks of this clustering approach. First, the inferred cell clusters do not account for the potential of differentiation by SNVs only. Second, by definition, the embedding of the cells into a low-dimensional space will naturally obscure small copy number differences between the clones. In contrast, PHARMING is able to provide a much higher resolution view into the evolutionary history of these cells (Fig. 6.4d,e).

First, we noted that the PHARMING clonal tree clearly distinguishes nodes 1, 4 and 7 corresponding to Laks et al. cluster F from nodes 8, 5, 0, 2, 3 and 6, which correspond to Laks et al. cluster E. These two sets of nodes are clearly differentiated in the PHARMING tree by the introduction of 136 SNVs at node 8 as well as a copy number state change to (0, 3) at segment 56. Second, we observed that the two largest PHARMING clones 5 and 0 comprising Laks et al. [24] cluster E are only differentiated by SNVs, whose placement is supported by our CMB metric (Fig. 6.4b). Thus, we expected to see cells from clone 5 and 0 scattered randomly throughout the Laks et al. [24] cluster E, which is indeed the case. Lastly, while it is difficult to assess if PHARMING is overfitting the copy number profiles since there is no lower bound on the number of cells that may be mapped to a node in the PHARMING tree, we do observe that the 6 cells assigned to clone 3 are clearly differentiated in UMAP space, suggesting that PHARMING is sensitive enough to detect the presence of very small clones.

The trends discussed for SA922 were also observed in SA921 (Fig. D.5) and SA1090 (Fig. D.4). Taken together, these analyses demonstrate that PHARMING holds the potential to offer an unparalleled level of detail in uncovering the evolutionary history of a tumor.

6.4 DISCUSSION

In this work, we introduced the CITCN problem and proposed a heuristic algorithm, PHARMING for inferring a joint CNA and SNV clonal tree from ultra-low coverage SCDNA-SEQ. We demonstrated the accuracy of our algorithm on both *in silico* experiments and a dataset consisting of three clonally related cell lines. On *in silico* experiments, we showed that PHARMING faithfully reconstructs the evolutionary history of CNAs and SNVs with respect to a known ground truth, as well as placement of cells onto the clonal tree. On experimental data, we should that PHARMING provides accurate and high resolution joint reconstruction of CNA and SNV evolution.

There are several directions for future work. First, we plan to overcome the scalability issues in a number of ways, including reimplementing PHARMING in C++ instead of Python as well as parallelizing our implementation. We could also overcome some of the scalability challenges by first using PHERTILIZER to identify the major clades of the clonal tree and then using PHARMING to get a high-resolution joint CNA and SNV view of each clade. Second, we plan to support parallel evolution of copy number states, i.e., violations of the infinite alleles assumption. Potential violations may be identified by looking at segments that do not integrate well with the other segments during progressive integration. Once identified, segments could be subjected to a more detailed inference by doubling one or more copy states before enumerating candidate CNA trees. Third, our progressive integration can be made more aware of dependencies between segments. For example, copy number events that span multiple segments should imply a consistent CNA evolutionary history within each segment. This could be overcome by first integrating such segments and applying additional consistency constraints. Fourth, our algorithmic approach is easily extendable to bulk DNA sequencing data or even joint bulk and single-cell data with only minor adjustments required. Finally, we plan to implement a web-based visualization tool to allow researchers to more easily interrogate and analyze these high dimensional clonal trees. For example, users will be able to filter by segment or a set of SNVs in order to drill down to specific evolutionary events(s) of interest. Additionally, we plan to support algorithmic steering of the progressive integration stage by allowing users to select an ordering and construct the complete clonal tree in real time.

CHAPTER 7: TRIBAL: CLONAL TREE INFERENCE OF B CELL CLONAL LINEAGES

7.1 INTRODUCTION

The generation of naive B cell receptor (BCR) stems from a process known as V(D)J recombination, where DNA segments are rearranged to ensure a wide spectrum of antibodies to counter various pathogens [178]. To enhance their effectiveness, B cells undergo affinity maturation (Fig. 7.1b) [56], a micro-evolutionary process involving repeated cycles of *somatic hypermutation* (SHM) and cellular divisions. SHM introduces mutations in the BCR genes, selecting for B cells expressing high-affinity BCRs, while eliminating those with low affinity. Concurrently, B cells have the ability for *class switch recombination* (CSR) (Fig. 7.1b) [57], which diversifies their response by altering the antibody’s functional class or *isotype*.

However, the selection pressures applied to B cells during the affinity maturation process necessitate more specialized analytical approaches than those utilized for species phylogeny inference [5, 58, 59, 60]. Specifically, Hoehn et al. developed HLP17 [61] and HLP19 [62], which are specialized codon substitution models for use with maximum likelihood inference via IgPhyML. Another important difference between B cell and species evolution is the lower mutation rate and the relatively short length of the BCR sequence (≈ 600 bp). These properties imply that maximum parsimony inference methods are viable [63, 64, 65] but typically result in a large solution space of many plausible phylogenies for the same data. In addition, these solutions exhibit additional tree uncertainty in the form of *polytomies*, i.e., multifurcating nodes with more than two children. This is counter to the underlying cell lineage tree, which is bifurcating as cell division results in exactly two daughter cells.

One approach to resolve phylogenetic uncertainty is the inclusion of additional data. This approach has proven effective in other areas, such as the use of physical location for studying cancer migration and metastasis [72], or geographical location for the inference of gene flow [73]. More related to B cell lineage inference, sequence abundance has been utilized by both GCTree [63] and ClonalTree [70] to discriminate between candidate solutions. However, both methods were originally developed for bulk RNA sequencing data and consequently were

⁷This chapter contains identical material with an adapted introduction previously published as the bioRxiv preprint “TRIBAL: Tree inference of B cell clonal lineages” [176], which is an extended version of the proceedings for RECOMB 2024 [177]. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with M. El-Kebir, A.A. Khan, D. Reiman, M.S. Roddur and Y. Qi. MEK and AAK conceived the project; LLW, MEK established theory and the method; MSR and YQ provided a proof of NP hardness. DR prepared the experimental data. LLW performed experiments; LLW, MEK, AAK contributed to the writing of this chapter.

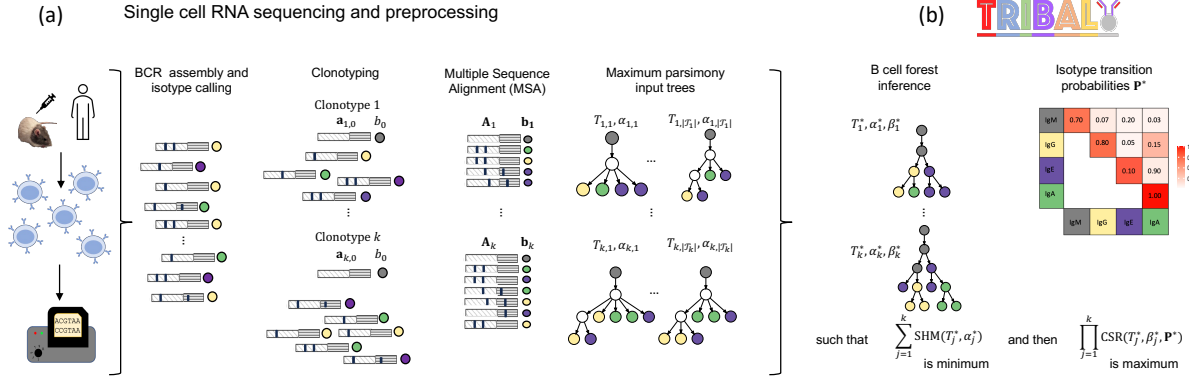


Figure 7.1: **TRIBAL infers B cell lineage trees and isotype transition probabilities for scRNA-seq data.** (a) After scRNA-seq, the variable regions for the light and heavy immunoglobulin alleles are assembled, the isotypes are called and the B cells are clustered into k clonotypes. A multiple sequence alignment A_j is found for each clonotype j and used to infer a set of input trees with maximum parsimony. The leaves of each input tree are labeled by isotypes b_j . (b) TRIBAL jointly infers a B cell lineage tree T_j^* for each clonotype j and population-specific isotype transition probabilities P^* with maximum parsimony for MSA A_j and maximum likelihood for isotypes b_j .

restricted to analysis of only the heavy chain sequences. With single cell RNA-sequencing (scRNA-seq), it is now possible to efficiently assemble BCR sequences that includes both heavy and light chain from a population of B cells [71] (Fig. 7.1c). As a result, the evolution of BCRs during affinity maturation can now be tracked with scRNA-seq with higher fidelity [17].

In addition, scRNA-seq yields another valuable data source in the form of the expressed isotype of the constant region of the heavy chain (Fig. 7.1c) [17]. Isotype data extracted from scRNA-seq has been helpful in related immunological domains, such as inferring dynamic cellular trajectories[179]. However, isotype expression is an especially useful marker of B cell evolution. When a B cell undergoes class switching from its current isotype to a new isotype, any heavy chain constant region locus between the current isotype and the new isotype in the genome is cut out or removed via a recombination process (Fig. 7.1b). Consequently, CSR is an irreversible process and the isotype state of a B cell offers a distinct milestone in its evolutionary history. Therefore, the inclusion of isotype information into the problem of B cell lineage inference has the potential to help minimize phylogenetic uncertainty by both reducing the size of the solution space and yielding more refined B cell lineage trees with fewer polytomies.

In this work, we present TRIBAL, which stands for Tree Inference of B cell Clonal Lin-

eages (Fig. 7.1d). TRIBAL utilizes both the BCR sequence and isotype information from sequenced cells to infer a B cell lineage tree that jointly models the evolutionary processes of SHM and CSR. Additionally, TRIBAL infers the underlying isotype transition probabilities providing valuable insight into the dynamics of CSR (Fig. 7.1d). We demonstrate the accuracy of TRIBAL on simulated data and show that it is effective on experimental single cell data generated from the 5' 10x Genomics platform. TRIBAL is available open source and has the potential to improve understanding of vaccine responses, track disease progression, and identify therapeutic antibodies.

7.2 MATERIALS AND METHODS

7.2.1 Problem Statement

Suppose that we have sequenced and subsequently aligned the variable regions of the heavy and light chain of the B cell receptor (BCR) of n B cells that descend from the same naive B cell post V(D)J recombination, resulting in a multiple sequence alignment (MSA) $\mathbf{A}$ composed of m columns — the typical alignment length is $m \approx 650$ (Fig. 7.1c). That is, for each B cell i , we are given the concatenated sequence $\mathbf{a}_i \in \Sigma^m$ where $\Sigma = \{A, G, C, T, -\}$. In addition, we are given the isotype $b_i \in [r] = \{1, \dots, r\}$, determined using tools such as Cell Ranger [180]. For humans, there are $r = 8$ isotypes ordered as IgM/D, IgG3, IgG1, IgA1, IgG2, IgG4, IgE and IgA2, whereas for mice there are $r = 7$ isotypes ordered as IgM/D, IgG3, IgG1, IgG2b, IgG2c (2a), IgE, IgGA. Finally, we are given sequence $\mathbf{a}_0$ and isotype $b_0 = 1$ of the ancestral naive B cell of the n B cells — note that $\mathbf{a}_0$ and b_0 is post V(D)J recombination but prior to somatic hypermutation (SHM) and class switch recombination (CSR) and thus $b_0 = 1$. (Fig. 7.1a).

To study the evolutionary history of these n cells, we aim to construct a B cell lineage tree T that jointly describes the evolution of the DNA sequences $\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n]^\top$ and their isotypes $\mathbf{b} = [b_0, b_1, \dots, b_n]^\top$. As such, each node v of T will be labeled by a sequence $\alpha(v) \in \Sigma^m$ and isotype $\beta(v) \in [r]$. In particular, the root v_0 will be labeled by $\alpha(v_0) = \mathbf{a}_0$ and $\beta(v_0) = b_0 = 1$ while the n leaves $L(T) = \{v_1, \dots, v_n\}$ of T will be labeled by sequence $\alpha(v_i) = \mathbf{a}_i$ and isotype $\beta(b_i) = b_i$ for each $i \in [n]$. A key property of isotype evolution is that it is *irreversible*. As such, the isotype $\beta(u)$ of an ancestral cell u must be less than or equal to the isotype $\beta(v)$ of its descendants v . More formally, we have the following definition of a B cell lineage tree.

Definition 7.1. A rooted tree T whose nodes are labeled by sequences $\alpha : V(T) \rightarrow \Sigma^m$ and

isotypes $\beta : V(T) \rightarrow [r]$ is a *B cell lineage tree* for MSA $\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n]^\top$ and isotypes $\mathbf{b} = [b_0, b_1, \dots, b_n]^\top$ provided (i) T has n leaves $L(T) = \{v_1, \dots, v_n\}$ such that each leaf $v_i \in L(T)$ is labeled by sequence $\alpha(v_i) = \mathbf{a}_i$ and isotype $\beta(v_i) = b_i$, (ii) the root node v_0 of T is labeled by sequence $\alpha(v_0) = \mathbf{a}_0$ and isotype $\beta(v_0) = b_0$, and (iii) for all nodes $u, v \in V(T)$ such that u is ancestral to v it holds that $\beta(u) \leq \beta(v)$.

In the following, we will refer to B cell lineage trees as *lineage trees*. Lineage trees typically have shallow depth due to the limited number of mutations introduced during SHM, making parsimony a reasonable optimization criterion [63, 64, 65]. Given a lineage tree T , the SHM parsimony score is computed as

$$\text{SHM}(T, \alpha) = \sum_{(u,v) \in E(T)} D(\alpha(u), \alpha(v)), \quad (7.1)$$

where $D(\alpha(u), \alpha(v))$ is the Hamming distance [181] between sequences $\alpha(u)$ and $\alpha(v)$. However, one common challenge of using parsimony to model SHM is that it often results in a large number of candidate lineage trees with equal optimal parsimony score. In addition, many inferred lineage trees contain *polytomies*, or internal nodes with out-degree greater than 2. To overcome these two challenges, we propose to infer lineage trees that optimize both sequence evolution (SHM) and isotype evolution (CSR).

Similarly to SHM, one could model the evolution of CSR using unweighted parsimony. That is, one would prefer lineage trees T with isotypes $\beta : V(T) \rightarrow [r]$ that minimize the number of isotype changes, i.e., $\sum_{(u,v) \in E(T)} D(\beta(u), \beta(v))$. However, there are two issues with this approach. First, it does not appropriately penalize lineage trees that violate the irreversible property of isotype evolution [64]. Second, it does not account for the fact that given an isotype starting state the probability of transitioning to each of the possible isotype states is not necessarily equal. In fact, knowing these probability distributions is useful for researchers looking to gain basic insight into the patterns and casual factors of class switch recombination [182]. Therefore, we seek to develop an appropriate evolutionary model for CSR that captures the irreversible property of class switching and models preferential isotype class transitions.

We propose a state or tree dependence model [183, 184] evolutionary model for CSR, which models the joint probability distribution of a random variable vector under Markov-like assumptions on a given tree (Appendix E.1). Here, the random variables of interest in this state tree model are the isotypes $\beta(v)$ of each node v in lineage tree T . This model is parameterized by a probability distribution over the isotype of the root and isotype transition probabilities. As the root v_0 of a lineage tree T is a naive B cell post V(D)J recombination,

the isotype $\beta(v_0)$ is always 1 (IgM) and the probability distribution of $\beta(v_0)$ is defined as $\Pr(\beta(v_0) = 1) = 1$ and 0 otherwise. Intuitively, isotype transition probabilities captures the conditional probability of a descendant isotype given the isotype of its parent subject to irreversible isotype evolution. Next, we give a formal definition of isotype transition probabilities.

Definition 7.2. An $r \times r$ matrix $\mathbf{P} = [p_{s,t}]$ is an *isotype transition probability matrix* provided for all isotypes $s, t \in [r]$ it holds that (i) $p_{s,t} \geq 0$, (ii) $p_{s,t} = 0$ if $s > t$, and (iii) $\sum_{t=1}^r p_{s,t} = 1$ for all isotypes $s \in [r]$.

We define the joint likelihood $\text{CSR}(T, \beta, \mathbf{P})$ of the observed isotypes $\mathbf{b}$ for isotype transition probabilities $\mathbf{P}$ and any lineage tree T whose leaves have isotypes $\mathbf{b}$ as

$$\text{CSR}(T, \beta, \mathbf{P}) = \Pr(\mathbf{b} \mid T, \alpha, \beta, \mathbf{P}) = \Pr(\mathbf{b} \mid T, \beta, \mathbf{P}) = \prod_{(u,v) \in E(T)} p_{\beta(u), \beta(v)}. \quad (7.2)$$

We consider the problem of simultaneously inferring a lineage tree T with nodes labeled by sequences $\alpha(v)$ and isotypes $\beta(v)$ given MSA $\mathbf{A}$ and isotypes $\mathbf{b}$ that optimizes both $\text{SHM}(T, \alpha)$ and $\text{CSR}(T, \beta, \mathbf{P})$. However, a significant barrier to solving this problem is that isotype transition probabilities $\mathbf{P}$ are unknown and need to be inferred. While there have been experimental studies that estimate these quantities under specific biological conditions [182], there currently exists no computational methods to infer these probabilities directly from a sequencing experiment. Here, we will leverage that typical single-cell sequencing experiments yield data from a set of diverse B cell lineages, where each lineage is commonly referred to as a *clonotype*. A common pre-processing step is to first cluster the n sequenced cells into k clonotypes, where each clonotype j contains n_j cells (Fig. 7.1c). We follow convention and group B cells based on shared V(D)J alleles in heavy and light chains into a clonotype. Moreover, we reason that under many experimental conditions, the transition probabilities between isotypes will be similar to those within clonotypes. Thus, inferring these isotype transition probabilities for k clonotypes will yield higher accuracy than inferring isotype transition probabilities for a single lineage in isolation. This leads to the following problem statement.

Problem 7.1 (B CELL LINEAGE FOREST INFERENCE (BLFI)). Given MSAs $\mathbf{A}_1, \dots, \mathbf{A}_k$ and isotypes $\mathbf{b}_1, \dots, \mathbf{b}_k$ for k clonotypes, find isotype transition probabilities $\mathbf{P}^*$ for r isotypes and lineage trees $T_1^*, \dots, T_k^*$ for $(\mathbf{A}_1, \mathbf{b}_1), \dots, (\mathbf{A}_k, \mathbf{b}_k)$ whose nodes are labeled by sequences $\alpha_1^*, \dots, \alpha_k^*$ and isotypes $\beta_1^*, \dots, \beta_k^*$, respectively, such that $\sum_{j=1}^k \text{SHM}(T_j^*, \alpha_j^*)$ is minimum and then $\prod_{j=1}^k \text{CSR}(T_j^*, \beta_j^*, \mathbf{P}^*)$ is maximum.

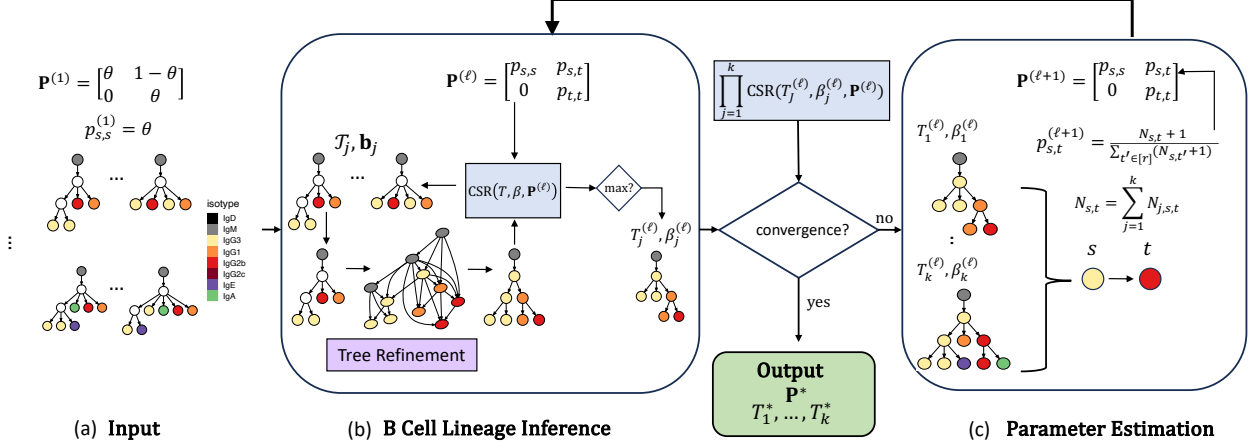


Figure 7.2: **TRIBAL** infers B cell lineage forest $T_1^*, \dots, T_k^*$ and isotype transitions $\mathbf{P}^*$ for k clonotypes utilizing coordinate ascent. (a) The inputs to TRIBAL are isotype transition probabilities $\mathbf{P}^{(1)}$, which are initialized given a parameter $\theta \in [0.5, 1]$, and a tuples $(T_1, \mathbf{b}_1) \dots, (T_k, \mathbf{b}_k)$, where set T_j are maximum parsimony trees for MSA $\mathbf{A}_j$ and $\mathbf{b}_j$ are the observed isotypes of the n_j cells of clonotype j . (b) Conditioning on isotype transition probabilities $\mathbf{P}^{(\ell)}$, a B cell lineage tree $T_j^{(\ell)}$ with nodes labeled by isotypes $\beta_j^{(\ell)}$ is inferred for each clonotype j by solving the MPTR problem for each tree in the input set T_j . (c) Convergence between $\prod_{j=1}^k \text{CSR}(T_j, \beta_j, \mathbf{P})$ for iterations ℓ and $\ell - 1$ is checked. If the difference has not converged, isotype transition probabilities $\mathbf{P}^{(\ell+1)}$ are updated using maximum likelihood estimation. If the difference has converged, the current inferred B cell lineage forest and isotype transition probabilities $\mathbf{P}$ are output. Multiple restarts may be performed for different values of θ .

In other words, we prioritize the SHM objective and then among all solutions with minimum SHM score we prefer those that additionally maximize the CSR objective. It is easy to see that the BLFI problem is NP-hard via a simple reduction from the Large Parsimony problem — see Appendix E.2.1 for more details.

We introduce Tree Inference of B cell Clonal Lineages, or TRIBAL, a method to solve the BLFI problem. This method is based on two key ideas that allow us to effectively solve the BLFI problem.

First, the lexicographical ordering of our two objectives — optimizing for SHM followed by CSR — enables one to use the following two-stage approach (Fig. 7.1c). In the first stage, we use existing maximum parsimony methods to generate a set $\mathcal{T}$ of input trees — also called a *maximum parsimony forest* — for each clonotype such that each tree $T \in \mathcal{T}$ minimizes the objective $\text{SHM}(T, \alpha)$. To do so, we provide these methods only the sequence information $\mathbf{A}$ to enumerate a solution space $\mathcal{T}$ of trees whose nodes are labeled by sequences $\alpha_1, \dots, \alpha_{|\mathcal{T}|}$. In the second stage, we incorporate isotype information $\mathbf{b}$ to further operate on the set $\mathcal{T}$

and additionally optimize $\text{CSR}(T, \beta, \mathbf{P})$ in such a manner that maintains optimality of the SHM objective. We note that a lexicographically optimal lineage tree T^* does not necessarily need to be an element of $\mathcal{T}$, but instead it suffices that the evolutionary relationships in tree T^* are a refinement of the evolutionary relationships described by some tree T among the set $\mathcal{T}$ of input trees. More specifically, a *refinement* T' of tree T is obtained by a series of EXPAND operations such that an EXPAND operation of node v consists of splitting node v into v and v' , joining them with an edge (v, v') and then redistributing the children of v to be descendants of either v or v' . We have the following key proposition and corollary.

Proposition 7.1. For any tree T labeled by sequences α and refinement T' of T , there exists a sequence labeling α' for T' such that $\text{SHM}(T, \alpha) = \text{SHM}(T', \alpha')$.

Proof. The sequencing labeling α' is found by setting $\alpha'(v) = \alpha(v)$ and $\alpha'(v') = \alpha(v)$ during each EXPAND operation. By construction, the new edge (v, v') has $D(\alpha'(v), \alpha'(v')) = 0$ and every original edge maintains its original Hamming distance in T' . Therefore, $\text{SHM}(T, \alpha) = \text{SHM}(T', \alpha')$. QED.

Corollary 7.1. Any lineage tree T' that lexicographically optimizes $\text{SHM}(T', \alpha')$ and then $\text{CSR}(T', \beta', \mathbf{P})$ must be a refinement of some tree T optimizing only $\text{SHM}(T, \alpha)$.

Therefore, our sought lineage tree T^* that lexicographically optimizes both objectives must be a refinement of some tree T in the set $\mathcal{T}$.

The second key idea is that the inference of optimal lineage trees $T_1^*, \dots, T_k^*$ is conditionally independent when given isotype transition probabilities $\mathbf{P}$. This motivates the use of a coordinate ascent algorithm where we randomly initialize isotype transition probabilities $\mathbf{P}^{(1)}$ (Sec. 7.2.2 and Fig. 7.2a). Then, at each iteration ℓ , we use isotype transition probabilities $\mathbf{P}^{(\ell)}$ and the input set $\mathcal{T}_j$ of trees to independently infer an optimal lineage tree $T_j^{(\ell)}$ for each clonotype j (Sec. 7.2.3 and Fig. 7.2b). This is then followed by estimating updated isotype transition probabilities $\mathbf{P}^{(\ell+1)}$ given trees $T_1^{(\ell)}, \dots, T_k^{(\ell)}$ (Sec. 7.2.4 and Fig. 7.2c). We terminate upon convergence of our CSR objective or when exceeding a specified number of maximum iterations.

TRIBAL is implemented in Python 3, is open source (BSD-3-Clause license), and is available at <https://github.com/elkebir-group/tribal>.

7.2.2 Input

The input to TRIBAL is a set of k clonotypes with corresponding maximum parsimony forest $\mathcal{T}_j$ for each clonotype j . In addition, we are given isotypes $\mathbf{b}_j$ labeling the leaves of

trees $\mathcal{T}_j$ for each clonotype j (Fig. 7.1c, Fig. 7.2a). Obtaining this input requires a number of preprocessing steps of a scRNA-seq dataset (Fig. 7.1), including (i) BCR assembly and isotype calling of each sequenced cell, (ii) clonotyping or clustering the cells based on a shared germline alleles for both the heavy and light chains, (iii) obtaining an MSA for sequences within a clonotype and (iv) finding a parsimony forest for each MSA of a clonotype. These preprocessing steps are not part of TRIBAL.

For our coordinate ascent approach, we also require an initialization for the isotype transition probabilities (Fig. 7.2a). We set the initial transition probabilities to reflect the observation that under baseline conditions, the probability of a B cell undergoing class switching is lower than the probability of it maintaining its original antibody class. [56, 182]. Thus, we initialize $\mathbf{P}^{(1)}$ such that $p_{s,s} > p_{s,t}$ for all isotypes s and t . Let $\theta \in [0.5, 1]$ be the probability that a B cell does not class switch, i.e., $p_{s,s} = \theta$ for each isotype $s < r$ and $p_{s,s} = 1$ if $s = r$. We enforce irreversibility such that $p_{s,t} = 0$, if $s > t$. We then initialize the remaining parameters uniformly, i.e., $p_{s,t} = (1 - p_{s,s})/(r - s)$ where r is the total of number isotypes. We conduct multiple restarts, varying $\theta \in [0.5, 1]$ in each restart.

7.2.3 B cell lineage tree inference via refinement

As described above, the inference of optimal lineage trees $T_1^{(\ell)}, \dots, T_k^{(\ell)}$ is conditionally independent given isotype transition probabilities $\mathbf{P}^{(\ell)}$. We therefore focus our discussion on how TRIBAL infers a B cell lineage tree $T_j^{(\ell)}$ for a single clonotype j during iteration ℓ given isotype transition probabilities $\mathbf{P}^{(\ell)}$. By Corollary 7.1, we solve this problem by finding an optimal refinement T' and corresponding isotype labeling β' for each tree T in the input set $\mathcal{T}_j^{(\ell)}$ and select the one that maximizes our CSR objective (Fig. 7.2b). Maximizing the log-likelihood of $\text{CSR}(T, \beta, \mathbf{P})$ is equivalent to maximizing a weighted parsimony criterion. This leads to the following problem statement.

Problem 7.2 (MOST PARSIMONIOUS TREE REFINEMENT (MPTR)). Given a tree T on n leaves, isotypes $\mathbf{b} = [b_0, \dots, b_n]$ and isotype transition probabilities $\mathbf{P}$, find a tree T' with root v'_0 and isotype labels $\beta' : V(T') \rightarrow [r]$ such that (i) T' is a refinement of T , (ii) $\beta'(v'_0) = b_0 = 1$, (iii) $\beta'(v'_i) = b_i$ for each leaf $v'_i \in \{v'_1, \dots, v'_n\}$ and (iv) $\log \text{CSR}(T', \beta', \mathbf{P})$ is maximum.

It was shown by Mrinmoy S. Roddur and Yuanyuan Qi in Weber et al. [176] that the MPTR is NP-hard. Therefore, we convert this problem to a graph problem (Fig. E.1). For each node u in tree T , multiple copies (u, s) are added to the expansion graph $G_{T, \mathbf{b}}$, one for each possible isotype $s \in [r]$ subject to additional constraints depending on the

observed isotypes of the leaves descendant from u . The edges $((u, s), (v, t))$ in this graph have weight $\log p_{s,t}$. We then solve the MPTR problem by finding a valid, maximum weight subtree (T', β') in the expansion graph $G_{T, \mathbf{b}}$ such that T' with isotype labels β' is a most parsimonious refinement of T . This is achieved via a mixed-integer linear programming (MILP) formulation, similar to that used to solve the Steiner minimal tree problem [185]. See Appendix E.3.1 for more details on the construction of the expansion graph from tree T and leaf isotypes $\mathbf{b}$, proof of correctness for this algorithm and the MILP formulation.

To obtain outputs $(T_1^{(\ell)}, \beta_1^{(\ell)}), \dots, (T_k^{(\ell)}, \beta_k^{(\ell)})$, we set $(T_j^{(\ell)}, \beta_j^{(\ell)})$ to the tree T' and corresponding isotype labeling β' that maximize $\text{CSR}(T', \beta', \mathbf{P}^{(\ell)})$ among all input trees $\mathcal{T}_j^{(\ell)}$ for each clonotype $j \in [k]$.

7.2.4 Parameter estimation

In the next step, given tuples $(T_1^{(\ell)}, \beta_1^{(\ell)}), \dots, (T_k^{(\ell)}, \beta_k^{(\ell)})$ of lineage trees and isotype labels for iteration ℓ , we seek updated isotype transition probabilities $\mathbf{P}^{(\ell+1)}$ for iteration $\ell + 1$ such that $\prod_{j=1}^k \text{CSR}(T_j^{(\ell)}, \beta_j^{(\ell)}, \mathbf{P}^{(\ell+1)})$ is maximum (Fig. 7.2c). This is achieved via maximum likelihood estimation. For fixed lineage trees $T_1, \dots, T_k$ and isotypes $\beta_1, \dots, \beta_k$, our CSR objective can be rewritten as

$$\prod_{j=1}^k \text{CSR}(T_j, \beta_j, \mathbf{P}) = \prod_{j=1}^k \prod_{(u,v) \in E(T_j)} p_{\beta_j(u), \beta_j(v)} = \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{N_{s,t}} \quad (7.3)$$

where $N_{s,t} = \sum_{j=1}^k \sum_{(u,v) \in E(T_j)} \mathbf{1}(\beta_j(u) = s, \beta_j(v) = t)$ is the number of transition from s to t across all selected trees and clonotypes. Thus, we seek isotype transition probabilities $\mathbf{P}$ that maximize (7.3) subject to the constraints that $\sum_{t \in [r]} p_{s,t} = 1$ for every isotype s . This constrained optimization problem is easily solved with the aid of Lagrange multipliers (see Appendix E.3.2 for details). Additionally, we add pseudocounts of 1 to avoid overfitting the data in the event of unobserved transitions, yielding the following updated probabilities $p_{s,t}^{(\ell+1)}$ for iteration $\ell + 1$.

$$p_{s,t}^{(\ell+1)} = \frac{N_{s,t} + 1}{\sum_{t' \in [r]} (N_{s,t'} + 1)}. \quad (7.4)$$

After estimating $\mathbf{P}^{(\ell+1)}$, we then infer $(T_1^{(\ell+1)}, \beta_1^{(\ell+1)}), \dots, (T_k^{(\ell+1)}, \beta_k^{(\ell+1)})$, as discussed in Section 7.2.3.

7.3 RESULTS

We evaluated TRIBAL on both *in silico* experiments (Sec. 7.3.1) and two sets of experimental scRNA-seq data. The first assessed performance on a model immunological system for affinity maturation (Sec. 7.3.2) and the second on a study of the relationship between age-associated B cells and autoimmune disorders (Sec. 7.3.3).

7.3.1 *In silico* experiments

We designed *in silico* experiments to evaluate TRIBAL with known ground-truth isotype transition probabilities $\mathbf{P}$ and lineage trees T labeled by sequences α and isotypes β . Specifically, we used an existing BCR phylogenetic simulator [64] that models SHM but not CSR. We generated isotype transition probabilities $\mathbf{P}$ with $r = 7$ isotypes (as in mice) under two different models of CSR. Briefly, both CSR models assume the probability of not transitioning is higher than the probability of transitioning, but in the *sequential model* there is clear preference for transitions to the next contiguous isotype, while in the *direct model* the probabilities of contiguous and non-contiguous class are similar. Given $\mathbf{P}$, we evolved isotype characters down each ground truth lineage tree T . We generated 5 replications of each CSR model for $k = 75$ clonotypes and $n \in \{35, 65\}$ cells per clonotype, resulting in 20 *in silico* experiments, yielding a total of 1500 ground truth lineage trees. In addition to comparing TRIBAL to existing methods including dnapars [5], dnaml [5] and IgPhyML [61], we also compared to a version of TRIBAL without tree refinement, denoted as TRIBAL-NO REFINEMENT (TRIBAL-NR). To obtain the input set $\mathcal{T}_j$ of trees with maximum parsimony for each clonotype j , we utilized dnapars [5]. We refer to Appendix E.4 for additional details on the simulations. In the following we focus our discussion on *in silico* experiments with $n = 35$ cells per clonotype (see Fig. E.4 for $n = 65$).

To evaluate accuracy of isotype transition probability inference, we used *Kullback–Leibler (KL) divergence* [186] to compare the inferred transition probability distribution $\hat{\mathbf{p}}_s$ of each isotype s to the simulated ground truth distribution $\mathbf{p}_s$. KL divergence is defined as $D_{\text{KL}}(\hat{\mathbf{p}}_s || \mathbf{p}_s) = \sum_{q \in [r]} \hat{p}_{s,t} \log(\hat{p}_{s,t}/p_{s,t})$; the lower the KL divergence, the more similar the two distributions. Since no existing methods infer isotype transition probabilities, we restricted this analysis to TRIBAL and TRIBAL-NR. Overall, we observed good concordance between simulated and TRIBAL inferred isotype transition probabilities (Fig. 7.3a). Specifically, TRIBAL had lower median KL divergence than TRIBAL-NR for all isotype starting states, except IgA, which is trivially 0, under both direct and sequential CSR models (direct: median of 0.15 vs. 0.73; sequential: median of 0.099 vs. 0.55). We observed improved perfor-

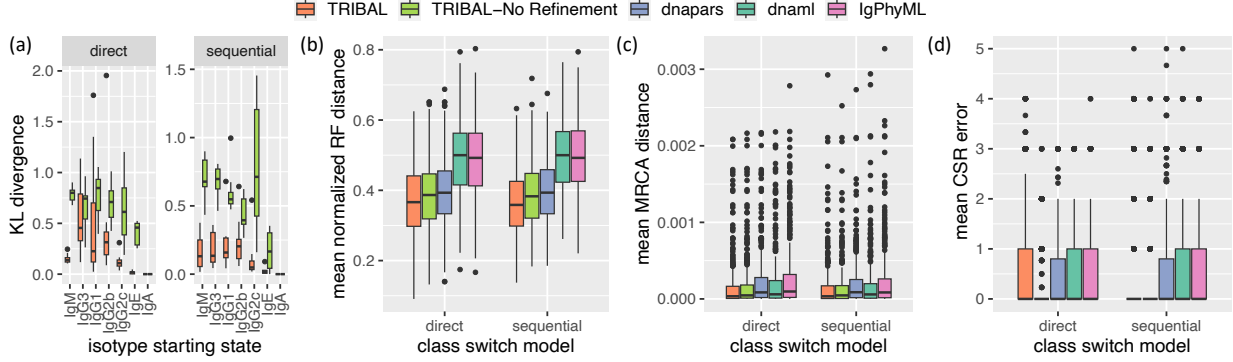


Figure 7.3: **TRIBAL accurately infers isotype transition probabilities on simulated data while outperforming existing methods on lineage tree inference.** Simulation results shown are for 5 replications with $k = 75$ clonotypes per replication and $n = 35$ cells per clonotype. (a) KL divergence between inferred isotype transition probabilities and the reference ground truth distribution. (b) mean Robinson-Foulds distance between ground truth and inferred lineages tree per clonotype. (c) mean MRCA distance (E.36) between ground truth and inferred lineage trees per clonotype. (d) mean CSR error between ground truth and inferred B cells.

mance of TRIBAL (but not for TRIBAL-NR) for $n = 65$ cells per clonotype (Fig. E.4a and Fig. E.5). To assess the sensitivity of TRIBAL to infer isotype transition probabilities with fewer than $k = 75$ clonotypes, we downsampled the 75 clonotypes to 25 and 50 clonotypes per experiment. We observed similar trends for experiments with $k \in \{25, 50\}$ clonotypes, with TRIBAL continuing to outperform TRIBAL-NR while still achieving small KL divergences even as k decreases (Fig. E.6). These findings demonstrate that tree refinement is key to accurately estimate isotype transition probabilities.

Next, we assessed the accuracy of lineage tree inference using the *Robinson-Foulds (RF) distance* [187] normalized by the total number of bipartitions in the ground truth T and inferred lineage tree $\hat{T}$ (E.35). Since TRIBAL, TRIBAL-NR and dnaphars return multiple optimal solutions, we report the mean of the lineage tree inference metrics over all optimal solutions. We found TRIBAL had the lowest mean normalized RF distance for both direct and sequential CSR models (Fig. 7.3b). Overall, dnaml (median: 0.5) and IgPhyML (median: 0.49) had the worst performance on normalized RF. Interestingly, even though the starting trees of dnaphars are used by TRIBAL, both TRIBAL (median: 0.36) and TRIBAL-NR (median: 0.38) outperformed dnaphars (median: 0.39), showing the importance of using isotype information to resolve phylogenetic uncertainty.

While normalized RF distance only assesses the accuracy of the tree topology, it is important to also assess the accuracy of the ancestral sequence reconstruction. To that end, we used a metric called *Most Recent Common Ancestor (MRCA) distance* (E.36) introduced

by Davidsen and Matsen [64]. For any two simulated B cells (leaves), the MRCA distance is the Hamming distance between the MRCA sequences of these two B cells in both the ground truth and inferred lineage trees. This distance is then averaged over all pairs of simulated B cells — see Appendix E.4.4 and Fig. E.3a for additional details. Again, we report the mean of over all optimal solutions for TRIBAL, TRIBAL-NR and dnapars. We found TRIBAL outperformed all other methods (Fig. 7.3c), achieving the lowest overall median MRCA distance (3.46×10^{-5}), followed by TRIBAL-NR (4.63×10^{-5}). IgPhyML had the worst performance with a median of 8.78×10^{-5} . Performance trends were consistent between methods across both CSR models.

Lastly, we assessed the accuracy of isotype inference by a new metric called *CSR error*, which is computed for each B cell i and clonotype j and is the absolute difference between the number of ground-truth class switches and inferred number of class switches that occurred along its evolutionary path from the root — see Appendix E.4 and Fig E.3b for additional details. Since dnaml, dnapars and IgPhyML do not infer isotypes for internal nodes, we pair these methods with the Sankoff algorithm [188] using $w_{s,t}$ equals 1 if $s = t$, 0 if $s < t$ and ∞ otherwise. We account for the presence of multiple solutions by taking the mean across solutions. All methods had a median CSR error of 0 for both the direct and sequential models (Fig. 7.3d). Therefore, we utilized the third quartile for a more robust comparison. We found that under the direct model TRIBAL-NR (third quartile: 0) was the best performing method, while dnapars was second best (third quartile 0.8) and all other methods, including TRIBAL had a third quartile of 1. We observed a slight tendency of TRIBAL to overestimate the number of transitions due to the tree refinement step, while other methods tended to underestimate the number of transitions. However, under a sequential model, where refinement is helpful in accurately capturing sequential state transitions, we found that TRIBAL was tied with TRIBAL-NR for the best performance (third quartile: 0). All other methods had similar performance between both CSR models for this metric.

We observed similar trends on these metrics for *in silico* experiments containing $n = 65$ cells per clonotype for $k = 75$ clonotypes (Fig. E.4). In summary, these results suggests that the incorporation of isotype data and the use of tree refinement are beneficial for both lineage tree inference and ancestral sequence and class switch reconstruction.

7.3.2 Keyhole limpet haemocyanin (NP-KLH) antigen immune response studies

We applied TRIBAL as well as IgPhyML to $10 \times 5'$ scRNA-seq data of B cells extracted from mice immunized with nucleoprotein keyhole limpet haemocyanin (NP-KLH), a commonly used antigen in the study of antibody affinity maturation [189]. Our goal was to

determine whether these methods recapitulate known patterns of B cell lineage evolution for this well-studied antigen using data from two studies and to compare the lineage trees inferred by each method. The first dataset (NP-KLH-1) was generated from C57BL/6 mice that were immunized with NP-KLH and total germinal center B cells were extracted 14 days after immunization [179]. The other two datasets came from a single study in which C57BL/6 mice were immunized with NP-KLH (NP-KLH-2a and NP-KLH-2b) and NP-specific germinal center B cells were extracted 13 days after immunization [190]. We utilized the standard 10× Cell Ranger [180] single-cell bioinformatics pipeline to generate sequence $\mathbf{a}_i$ and isotype b_i for each cell i . We used Dandelion [191] to remove doublets, reassign alleles, and cluster the cells into clonotypes. We identified clonotype MSAs $\mathbf{A}_1, \dots, \mathbf{A}_k$ based on shared V(D)J alleles for the heavy chain using the dowser package [65]. Finally, we excluded clonotypes with fewer than 5 cells. This yielded a total of $n = 2670$ sequenced B cells clustered into $k = 295$ clonotypes. We exclude methods that rely on sequence abundance as a key signal, such as GCTree [63] and ClonalTree [70] as we observed very few duplicated sequences within each clonotype. Fig. 7.4a shows the distribution of isotypes by dataset and Table E.1 includes a more detailed summary of each dataset.

We used dnapars [5] to infer TRIBAL’s input set $\mathcal{T}_j$ for each clonotype j . We found that TRIBAL’s use of isotype information significantly reduced the number of optimal solutions identified by dnapars (mean: 31.5 vs. 1.3, max: 4310 vs. 8) — see Fig 7.4b. While IgPhyML, a maximum likelihood method using the HLP19 codon-substitution model [61, 62], infers only a single tree per clonotype, it is important to note that there might be multiple trees with maximum likelihood in the solution space. Indeed, we found high concordance of HLP19 likelihoods between the TRIBAL and IgPhyML inferred lineage trees, with a small overall mean absolute deviation of 0.97 (Fig. 7.4c, Fig. E.7). We even observed that TRIBAL had a greater likelihood than IgPhyML in 59.3% of the clonotypes. Thus, TRIBAL resulted in a significant reduction in the size of the solution space compared to the maximum parsimony method dnapars with similar (and sometimes better) HLP19 likelihood as IgPhyML, illustrating how isotype information can be used to effectively reduce phylogenetic uncertainty.

Next, we assessed whether the lineage trees inferred by TRIBAL recapitulated expected biological trends for the NP-KLH model system. Previous research has indicated that the IGHV1-72*01 (VH186.2) variable gene in the heavy chain is preferentially used in the anti-NP response in C57BL/6 mice [192] via two distinct mutation paths: one in which a W33L mutation greatly increases affinity to NP, and another in which high affinity is achieved through K59R and the accumulation of several other mutations, such as S66N[193, 194]. Consequently, we expect that for clonotypes containing both W33L and K59R, these muta-

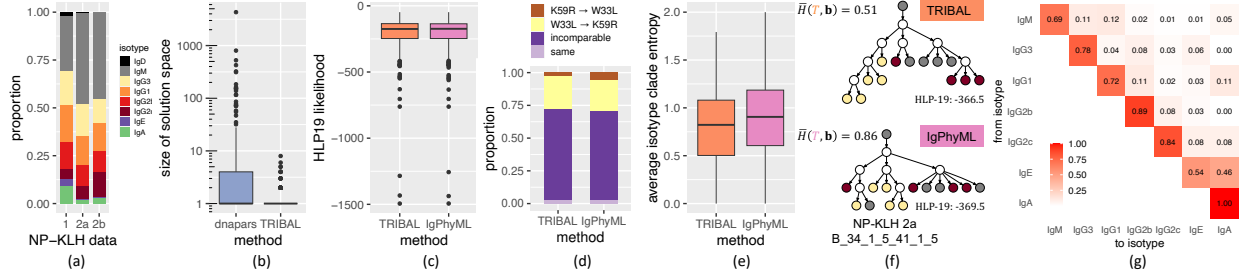


Figure 7.4: **Comparison between TRIBAL and IgPhyML on the NP-KLH data.** (a) The distribution of isotypes **b** in each dataset. (b) A comparison of the solution space of dnaphars versus TRIBAL. (c) The distribution of the HLP19 codon-substitution likelihood [62] for lineage trees inferred by TRIBAL and IgPhyML. (d) Observed distribution of evolutionary relationships between the W33L and K59 in clonotypes where both mutations are present. (e) The distribution of the average clade entropy with respect to an isotype labeling **b** of the leafset. (f) A comparison of lineage trees inferred for clonotype NP-KLH-2a B_34_1_5_41_1_5 with the average isotype clade entropy $\bar{H}(T, \mathbf{b})$ reported for each inferred tree. (g) TRIBAL inferred isotype transition probabilities **P** for NP-KLH-1.

tions would tend towards occurring in distinct lineages of the tree. We analyzed the inferred pairwise relationships between W33L & K59R in the 26 lineage trees that contained both mutations. Specifically, we categorized the relationship as $K59R \rightarrow W33L$ if K59R was ancestral to W33L, $W33L \rightarrow K59R$ if W33L was ancestral to K59R, *incomparable* if W33L and K59R occurred on distinct lineages of the tree and *same* if they were introduced on the same edge of the lineage tree. Indeed, we confirmed the tendency for mutual exclusivity of W33L and K59R by finding that the proportion of pairwise introductions categorized as *incomparable* was 0.69 and 0.67 for TRIBAL and IgPhyML, respectively (Fig. 7.4d). Additionally, it has been suggested that W33L mutations appear relatively early during the anti-NP response, whereas the K59R and S66N mutations typically appear later in the evolutionary history [195]. Defining level as the length of the shortest path from the MRCA of all B cells, we observed that W33L occurred at a median level of 1 for both TRIBAL and IgPhyML while both the K59R and S66N mutation occurred at a median level 2 for both methods (Fig. E.8). This indicated that W33L was typically introduced earlier in the evolutionary history of a clonotype than K59R and S66N. Thus, both TRIBAL and IgPhyML trees recapitulate expected mutation patterns for this model system.

We next assessed the extent of agreement with isotype information. While TRIBAL infers isotype labels of ancestral nodes, IgPhyML does not have this capability. Therefore, we developed a new metric called *average isotype clade entropy*, which is computed with respect to the isotype labeling of the leaf set. For this metric, we compute the entropy of clade u in tree T with respect to all isotype leaf labels that are descendants of node u ,

taking the average entropy over all non-trivial clades, which excludes the root and the leaves (Appendix E.4.5). As IgPhyML returns bifurcating trees, we collapse edges with zero branch length for a more fair comparison of this metric. We observed lower average isotype clade entropy for TRIBAL (median: 0.82) versus IgPhyML (median 0.91) 7.4f. Fig. 7.4e depicts the lineage tree inferred by TRIBAL and IgPhyML for the NP-KLH-2a dataset (clonotype B_34_1_5_41_1_5). The TRIBAL inferred tree for this clonotype had lower isotype clade entropy than IgPhyML (TRIBAL: 0.51 vs IgPhyML: 0.86) while also resulting in a greater HLP19 likelihood (TRIBAL: -366.5 vs IgPhyML: -369.5). Thus, we find that the trees identified by TRIBAL are in better agreement with the leaf isotypes than IgPhyML.

In addition to the inferred B cell lineage trees, TRIBAL also inferred isotype transition probabilities $\mathbf{P}$ for each dataset (Fig. 7.4g, Fig E.9). All three inferred isotype transition probability matrices more closely matched a CSR model of direct switching as opposed to a strictly sequential model. To compare the consistency of these estimates across datasets, we computed the Jensen-Shannon divergence (JSD) between the distribution of isotype transition probabilities for each isotype starting state IgM through IgG2c for each dataset pair. We observed low JSD (median: 0.029) across a total of 15 pairwise comparisons, suggesting consistent estimates between isotype transition probabilities.

In summary, these analyses show that the inclusion of isotype information and tree refinement has the potential to yield high quality lineage tree inference, even under a simpler model of SHM, i.e., parsimony. Moreover, the TRIBAL inferred lineage trees additionally optimize an evolutionary model of CSR, yielding lower isotype entropy partitions of the leaf set than IgPhyML. Finally, the additional inference of isotype transitions probabilities $\mathbf{P}$ has the potential to distinguish between direct versus sequential switching events.

7.3.3 Age-associated B cell (ABC) datasets

We evaluated TRIBAL on three scRNA-seq datasets with V region sequencing that investigated the relationship between age-associated B cells (ABCs) and autoimmune disorders [196]. For each dataset, B cells were extracted from the spleen of a MRL/lpr female mouse and sequenced using $10\times 5'$ scRNA-seq.

The data was processed by the $10\times$ Cell Ranger [180] single-cell bioinformatics pipeline to generate sequence $\mathbf{a}_i$ and isotype b_i for each cell i . Nickerson et al. [196] identified clonotype MSAs $\mathbf{A}_1, \dots, \mathbf{A}_k$ based on shared V(D)J alleles for the heavy chain using the dowser package [65] and inferred B cell lineage trees using IgPhyML for each clonotype. After filtering out clonotypes with fewer than 5 sequences, we retained 599 B cells and 54 clonotypes across the three datasets (Table E.2). Fig. 7.5a shows the proportion of isotypes

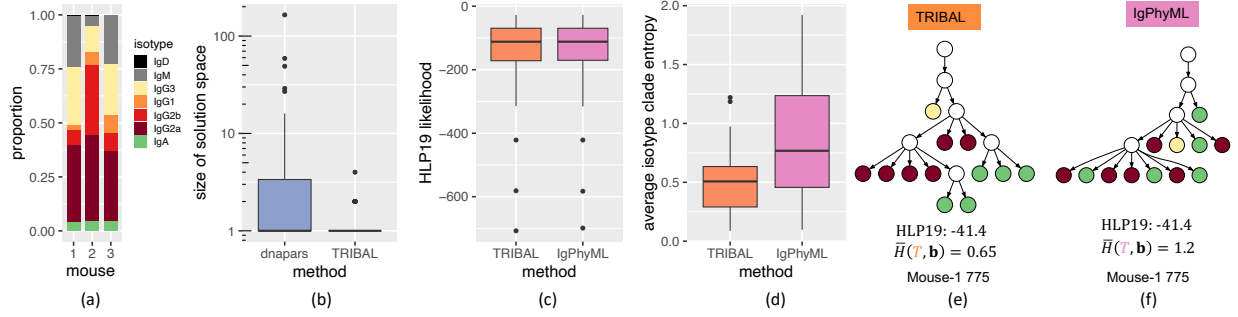


Figure 7.5: **Comparison between TRIBAL and IgPhyML on ABC data.** (a) Distribution of B cell isotypes. (b) A comparison of the solution space of dnaphars versus TRIBAL. (c) The distribution of the HLP19 codon-substitution likelihood [62] for lineage trees inferred by TRIBAL and IgPhyML. (d) Comparison of average isotype clade entropy for TRIBAL versus IgPhyML. (e,f) Comparison of inferred B cell lineage trees by TRIBAL (e) and IgPhyML (f) for clonotype Mouse-1 775 — see (a) for color legend.

and annotations by mouse for the retained B cells. Of these 54 clonotypes, 35 had more than one distinct isotype across the sequenced B cells, with a median of 3 distinct isotypes per clonotype.

We ran TRIBAL separately on each of the three mouse datasets, obtaining a maximum parsimony forest $\mathcal{T}_j$ for each clonotype j via dnaphars. Similar to our NP-KLH analysis, we found that TRIBAL effectively utilized the additional isotype data to reduce the number of optimal solutions identified by dnaphars (mean: 8.1 vs. 1.3, max: 165 vs. 4) — see Fig. 7.5b. The HLP19 likelihood of the TRIBAL inferred lineage trees had high concordance with the IgPhyML inferred trees (mean absolute deviation: 0.97), with TRIBAL yielding a higher likelihood for 53% of the clonotypes (Fig. 7.5c, Fig. E.10). The average isotype clade entropy for the 35 clonotypes with more than one distinct isotype was significantly lower for TRIBAL than for IgPhyML (median: 0.49 vs. 0.77) — see Fig. 7.5d. An example comparison is shown in Fig. 7.5e,f for clonotype Mouse-1 775. The tree refinement step of TRIBAL yielded a tree with a significantly lower average isotype clade entropy when compared to IgPhyML (0.65 vs. 1.2) while both trees had identical HLP19 likelihoods (-41.4). Finally, we observed that the isotype transition probabilities reveal evidence for both direct and sequential switching of isotypes (Fig. E.11).

In summary, both TRIBAL and IgPhyML yield lineage trees with very similar HLP19 likelihoods, giving support for the validity of the TRIBAL inferred lineage trees in terms of sequence evolution. However, TRIBAL jointly optimizes evolutionary models for both SHM and CSR, yielding trees with lower average isotype clade entropy.

7.4 DISCUSSION

The development and application of methods for inferring B cell lineage trees and isotype transition probabilities from scRNA-seq data is crucial for advancing our understanding of the immune system. In this work, we introduced TRIBAL, a method to infer B cell lineage trees and isotype transition probabilities from scRNA-seq data. TRIBAL makes use of existing maximum parsimony methods to optimize an evolutionary model for SHM, then incorporates isotype data to find the most parsimonious refinement, i.e., maximizing the CSR likelihood, among the input set of trees. We proved that the subproblem of finding a refinement maximizing the CSR likelihood is NP-hard. We demonstrated the effectiveness of TRIBAL via *in silico* experiments and on experimental data. On *in silico* experiments, we showed the importance of tree refinement for both accurately estimating isotype transition probabilities and lineage tree inference. Furthermore, we demonstrated on experimental data that TRIBAL returns lineage trees that have similar HLP19 likelihoods, despite utilizing a less complex model for sequence evolution but yield a reduction in the entropy of the isotype leaf labelings.

There are several directions for future research. First, integration of germline “sterile” transcripts may offer a way to initialize the TRIBAL inferred isotype transition probabilities [197]. Second, many existing B cell lineage inference methods, such as IgPhyML, yield multifurcating trees when zero length branches are collapsed. There exists an opportunity to combine likelihood or distance based inference methods with the tree refinement step of TRIBAL. Third, the MPTR problem has a more general formulation with the potential for wider applications beyond the problem of B cell lineage inference. For example, sample location is useful in refining tumor phylogeny with polytomies [72]. On a related note, we hypothesize that there are special cases of the MPTR problem and its more general formulation that are in P. Such special cases may include a weight matrix with unit costs and an upper triangular weight matrix that adheres to the triangle inequality. Fourth, the assumption that a single isotype transition probability matrix is shared by all clonotypes could be relaxed to allow the inference of multiple matrices per experiment and an assignment of clonotypes to an inferred matrix. Fifth, TRIBAL could be extended to allow for the correction of inaccurately clonotyped B cells. Finally, more robust evolutionary models for SHM are needed to capture the presence of complex mutations, such as indels, introduced during affinity maturation [198, 199].

CHAPTER 8: CONCLUSIONS

In this dissertation, we introduced the need for new algorithms that accurately model microevolutionary processes of cellular populations by optimizing the use of high resolution but error prone single-cell sequencing technologies. These algorithms are essential for extracting meaningful insights from complex biological data and have the potential to transform our understanding of microevolutionary dynamics. By overcoming the limitations of single-cell sequencing technologies and leveraging advanced computational techniques, this thesis lays the foundation for more precise and informative phylogenetic analyses.

To advance the field of tumor phylogenetics, we introduced five new algorithms addressing questions in optimal experimental design, the presence of common technical errors and tumor phylogeny inference from the latest generation of high throughput but ultra-low coverage SCDNA-SEQ technologies. In Chapter 2, we demonstrated that our proposed algorithm for phylogenetic design of single-cell sequencing experiments (PHYDOSE) significantly reduces the number of cells to sequence required to resolve phylogenetic uncertainty. This potential cost savings could help pave the way for more widespread use of single-cell tumor phylogenetics within the clinic. In Chapter 2, we introduced the first standalone method of doublet detection for medium to high coverage SCDNA-SEQ technologies. By integrating the microevolutionary process of clonal evolution and single-cell sequencing into a single probabilistic model, our method, DOUBLET-D, identifies and remove multiple types of doublets prior to downstream analysis, thereby increasing the accuracy and efficiency of existing phylogenetic inference methods.

Next, in Chapter 4, we addressed the prevalent issue of allelic dropout and the complexities of tumor phylogeny inference by first focusing on distinguishing linear from branched evolution from a set of sequenced cells. We introduced the Linear Perfect Phylogeny Flipping problem and proved it to be NP-hard. To solve this problem, we developed PHYOLIN, which determines the minimum number of inferred allelic dropout errors needed to yield a linear perfect phylogeny and then compares this rate to the expected allelic dropout rate of the sequencing technology. Finally, in Chapters 6 and 6, we addressed the lack of methods for inferring the evolutionary history of SNVs from ultra-low coverage SCDNA-SEQ technologies by proposing two new algorithms that utilize the strong copy number signal inherent in the data in two different ways. The first algorithm, PHERILIZER, requires minimal preprocessing of the data and scales well with the number of cells and SNVs but compresses the copy number signal in order to guide inference of the evolutionary history of SNVs. The second algorithm, PHARMING, requires allele-specific copy number calling as a preprocessing step

but yields a more comprehensive tumor phylogeny that captures the coevolution of SNVs and CNAs. Both algorithms yield accurate reconstructions of the evolutionary history of a tumor, facilitating new insights into the complexities of cancer and paving the way for better treatments.

But tumor cells undergoing clonal evolution are only one example of cellular populations undergoing microevolutionary processes. Another critical application for advancing human health involves B cells undergoing an adaptive immune response. In Chapter 7, we addressed a critical gap in the literature by proposing a new algorithm for B cell lineage inference (TRIBAL) from single-cell RNA sequencing data that jointly models the evolutionary processes of somatic hypermutation and class switch recombination. Our approach not only yields more accurate inference of B cell lineage trees and reconstruction of ancestral sequences, but also provides new biological insights into class switch dynamics during the immune response.

Overall, the work presented in this dissertation advances the field of computational biology by presenting new algorithms for studying the microevolutionary processes of cellular populations. In particular, this thesis highlighted the utility of designing tailored algorithms for microevolutionary processes from single-cell sequencing data and presents several high-level themes, including: (i) utilizing additional data to resolve phylogenetic uncertainty, (ii) the value of preprocessing data to mitigate the impact of technical errors on downstream analysis, and (iii) comprehensively reconstructing multiple underlying evolutionary processes that co-occur. These algorithms not only enhance our understanding of biological systems but also facilitate the development of more informed treatments and immunization strategies.

By focusing on tumor and B cell populations, it has addressed only a small subset of the microevolutionary processes that are ubiquitous in nature. Looking ahead, there remain numerous opportunities to apply these high-level themes to other biological entities, such as bacterial and viral populations. Furthermore, as single-cell sequencing technology continues to advance, new algorithms will be needed to capitalize on these advantages while simultaneously overcoming their disadvantages. For example, emerging technologies such as joint DNA/RNA single-cell sequencing provide both new technical challenges and opportunities for more directly linking clonal genotypes and phenotypes during tumor evolution. For B cell sequencing, technological advances are rapidly increasing throughput and improving whole transcriptomic sequencing. These advances pave the way to further enhance the comprehensiveness of our evolutionary models by incorporating cell fate and facilitating comparison of evolutionary trajectories across and within different disease cohorts. Ultimately, the continued development and application of these algorithms for microevolutionary processes will play a critical role in advancing personalized medicine and improving public health outcomes.

APPENDIX A: SUPPLEMENTARY METHODS AND RESULTS FOR PHYDOSE

A.1 SUPPLEMENTARY TEXT

A.1.1 Complexity

Theorem A.1. T -SCS-PC is NP-hard.

We prove the theorem using a polynomial-time reduction from the SET COVER problem, a known NP-hard problem [200].

Problem A.1 (SET COVER). Given a family $\mathcal{F}$ of subsets $\{S_1, \dots, S_{|\mathcal{F}|}\}$ over a universe $U = \{1, \dots, n\}$, find a cover $C \subseteq \mathcal{F}$ such that $\bigcup_{S \in C} S = U$ and C has minimum cardinality.

Specifically, we reduce a SET COVER instance $(\mathcal{F}, U)$ to an T -SCS-PC instance $(\mathcal{T}, T, \mathbf{f}, \gamma)$ as follows. The set $\mathcal{T} = \{T_0, \dots, T_n\}$ includes one tree T_i for each element i in the universe U and an additional tree T_0 . All trees in $\mathcal{T}$ have $|\mathcal{F}| + 2$ vertices, corresponding to subsets $\{S_1, \dots, S_{|\mathcal{F}|}\}$ and two additional mutations $\{\top, \perp\}$. Each tree in $\mathcal{T}$ includes the edge $(\top, \perp)$. Additionally, if element $i \in U$ is absent from subset S_j then there is an edge $(\top, S_j)$ in tree T_i , otherwise T_i includes an edge $(\perp, S_j)$. Tree T_0 includes edges $(\top, S_j)$ for all subsets S_j . As for the frequencies $\mathbf{f}$, we set $f_{\top} = 1$, $f_{\perp} = 0.5$ and the remaining frequencies $f_{S_j} = \varepsilon$ for all subsets $S_j \in \mathcal{F}$. Moreover, we set the confidence level γ to ε as well. In the corresponding T -SCS-PC instance $(\mathcal{T}, T_0, \mathbf{f}, \varepsilon)$, the tree of interest is T_0 . Fig. A.1 shows an example.

The key idea is that as $\gamma = \varepsilon > 0$ is a small positive infinitesimal constant, this T -SCS-PC instance seeks the smallest number k^* of cells such that $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f}))$ is non-zero. In particular, this number k^* of cells will only be achieved if there is a distinguishing feature Π of the same size k^* . By our reduction, there is a 1-1 correspondence between set covers of U and distinguishing features Π of T_0 with respect to $\{T_1, \dots, T_n\}$. Specifically, a set cover C of size k corresponds to a distinguishing feature $\Pi(C)$ of the same size k , and vice versa. As such, we have the following lemma whose proof is in the supplement.

⁸This chapter contains identical material with an adapted introduction previously published in “PhyDOSE: Design of follow-up single-cell sequencing experiments of tumors” [74], which is an extended version of the paper from RECOMB-CCB 2020 by the same title. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with N. Aguse and M. El-Kebir and N. Chia, MEK conceived the project; LLW, MEK and NA established theory; N. Chia provided biological and sequencing knowledge. LLW, NA performed experiments; LLW, MEK and NA contributed to the writing of this chapter.

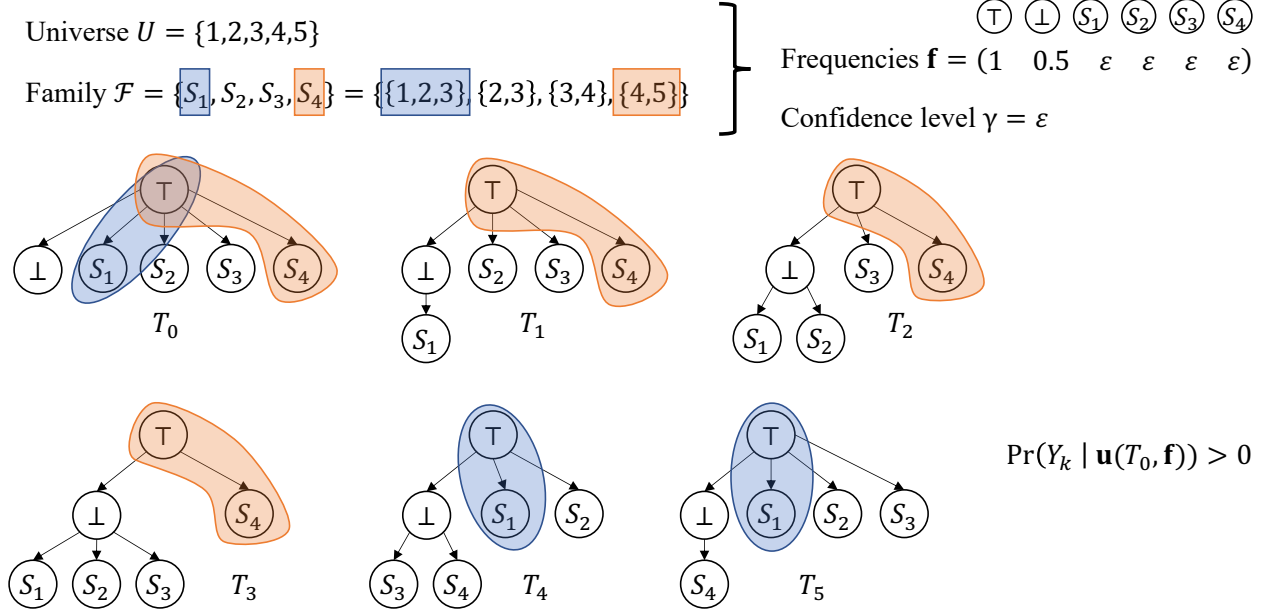


Figure A.1: **Reduction from SET COVER to T-SCS-PC.** Given a family $\mathcal{F}$ of subsets $\{S_1, \dots, S_{|\mathcal{F}|}\}$ on a universe $U = \{1, \dots, n\}$, we construct $n + 1$ trees $\mathcal{T} = \{T_0, \dots, T_n\}$ with mutations $\{\top, \perp, S_1, \dots, S_{|\mathcal{F}|}\}$. We seek to distinguish T_0 from the remaining trees $\{T_1, \dots, T_n\}$. The key concept captured by the reduction is that there is a cover of size k if and only if $\Pr(Y_k \mid \mathbf{u}(T_0, \mathbf{f}))$ is greater than 0. Here, S_1 and S_4 form a cover of size $k = 2$ of the universe U and the corresponding probability $\Pr(Y_k \mid \mathbf{u}(T_0, \mathbf{f}))$ is greater than 0.

Lemma A.1. Let $(\mathcal{T}, T_0, \mathbf{f}, \gamma = \varepsilon)$ be the T-SCS-PC instance corresponding to SET COVER instance $(U, \mathcal{F})$. A minimum cover has size k^* if and only if k^* is the smallest integer such that $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma$.

Proof. ($\Rightarrow$) Let C be a minimum cover of the SET COVER instance $(U, \mathcal{F})$. By the premise, we have that $|C| = k^*$. We start by showing that $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma$ by constructing a distinguishing feature $\Pi(C)$ of T_0 where $|\Pi| = k^*$. Observe that for each subset S_j in C we have that $\{\top, S_j\}$ is a featurette of T_0 . We define $\Pi(C)$ to be composed of featurettes $\{\top, S_j\}$ for all subsets $S_j \in C$. Thus, $|\Pi(C)| = k^*$. To show that $\Pi(C)$ is a distinguishing feature of T_0 , it remains to show that at least one featurette $\tau \in \Pi(C)$ is absent in each tree in $\mathcal{T} \setminus T_0 = \{T_1, \dots, T_n\}$. Consider any tree $T_i \neq T_0$. Since C is a cover, the element i of the universe U corresponding to tree T_i must be covered by some subset $S_j \in C$. This means that tree T_i contains the edge $(\perp, S_j)$, which means that the featurette $\{\top, S_j\}$ in $\Pi(C)$ is absent from T_i . Hence, $\Pi(C)$ is a distinguishing feature of T_0 .

We now must show that $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma$. We do so by focusing on distinguishing feature $\Pi(C)$. By construction of T_0 and $\mathbf{f}$, it follows from the sum condition (see Main Text) that each featurette $\{\top, S_j\}$ in $\Pi(C)$ has a clonal prevalence $u_j = \varepsilon$. This means that

a SCDNA-SEQ experiment of k^* cells where we only observe the k^* featurettes/clones has a probability that is strictly greater than 0. Therefore, $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) > 0$. Since ε is a small positive infinitesimal constant, we have that $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma = \varepsilon$.

It remains to show that k^* is the smallest integer where $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \varepsilon$. Assume for a contradiction that the smallest integer k' where $\Pr(Y_{k'} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \varepsilon$ is strictly smaller than k^* . This means that there exists a minimal distinguishing feature Π' of size at most k' . By definition Π' is composed of featurettes corresponding to root-to-vertex paths in T_0 . Since Π' is minimal, it will not contain the featurette $\{\top, \perp\}$ as this featurette is present in all remaining trees $\{T_1, \dots, T_n\}$. Thus, Π' is composed of featurettes of the form $\{\top, S_j\}$ where $S_j \in \mathcal{F}$. Since Π' is a distinguishing feature, no tree $T_i \in \{T_1, \dots, T_n\}$ contains all featurettes of Π' . By construction of $\{T_1, \dots, T_n\}$, this means that the subsets encoded in Π' form a cover of the universe U . Thus, there exists a cover with size strictly smaller than k^* , contradicting the premise. Therefore, k^* is indeed the smallest integer where $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma = \varepsilon$.

($\Leftarrow$) Let k^* be the smallest integer such that $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma = \varepsilon$. We start by showing that the size of a minimum distinguishing feature Π of T_0 has to be exactly k^* . Clearly, if $|\Pi| > k^*$ then $\Pr(Y_{k^*} \mid \mathbf{u}(T_0, \mathbf{f})) = 0$ as there exists no successful SCDNA-SEQ experiment with k^* cells. On the other hand, if $|\Pi| < k^*$ then there exists a successful SCDNA-SEQ experiment with $|\Pi|$ cells. In other words, $\Pr(Y_{|\Pi|} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \varepsilon$. This contradicts that k^* is the smallest integer where $\Pr(Y_{|\Pi|} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \varepsilon$. Hence, $|\Pi| = k^*$.

Consider a minimum distinguishing feature Π of T_0 . By the previous argument, we know that $|\Pi| = k^*$. We will show that Π encodes a cover $C(\Pi)$ of U of size k^* . Since Π is minimal, it will not contain the featurette $\{\top, \perp\}$ of T_0 as this featurette is present in all remaining trees $\{T_1, \dots, T_n\}$. Thus, Π is composed of k featurettes of the form $\{\top, S_j\}$ where $S_j \in \mathcal{F}$. Let $C(\Pi)$ be defined as the collection of subsets $S_j \in \mathcal{F}$ where $\{\top, S_j\}$ in Π . Since Π is a distinguishing feature, no tree $T_i \in \{T_1, \dots, T_n\}$ contains all featurettes of Π . By construction of $\{T_1, \dots, T_n\}$, this means that $C(\Pi)$ is a cover of size k of the universe U .

Finally, we must show that there exists no cover C' of U with size $|C'|$ strictly smaller than k^* . Suppose for a contradiction that such a cover C' exists. By construction, C' encodes a distinguishing feature $\Pi(C')$ composed of featurettes $\{\top, S_j\}$ for all subsets $S_j \in C'$. Thus, $|\Pi(C')| = |C'|$. To show that $\Pi(C')$ is a distinguishing feature of T_0 , we must show that (i) all features $\tau \in \Pi(C')$ are present in T_0 , and (ii) at least one featurette $\tau \in \Pi(C')$ is absent in each tree in $\mathcal{T} \setminus T_0 = \{T_1, \dots, T_n\}$. Condition (i) holds by construction of $\Pi(C')$ and T_0 , i.e. for each subset S_j in C' we have that $\{\top, S_j\}$ is a featurette of T_0 . As for condition (ii), consider any tree $T_i \neq T_0$. Since C' is a cover, the element i of the universe U corresponding to tree T_i must be covered by some subset $S_j \in C'$. This means that tree T_i contains the edge $(\perp, S_j)$, which means that the featurette $\{\top, S_j\}$ in $\Pi(C')$ is absent from T_i . Hence, $\Pi(C')$

is a distinguishing feature of T_0 . This in turn means that $\Pr(Y_{|\Pi(C')|} \mid \mathbf{u}(T_0, \mathbf{f})) > 0$. In other words, $\Pr(Y_{|\Pi(C')|} \mid \mathbf{u}(T_0, \mathbf{f})) \geq \gamma = \varepsilon$, thus contradicting the premise. Hence, minimum set covers of $(U, \mathcal{F})$ have cardinality k^* . QED.

The theorem follows from the above lemma, as the reduction to obtain $(\mathcal{T}, T_0, \mathbf{f}, \gamma = \varepsilon)$ from $(U, \mathcal{F})$ takes only polynomial time.

A.2 SUPPLEMENTARY METHODS

A.2.1 Finding the minimal distinguishing feature family

To perform the calculation, it is necessary to first find the minimal distinguishing feature family Φ^* . Using similar ideas as in our hardness proof (Section A.1.1), we consider the reverse reduction from the problem of finding a minimal distinguishing feature to that of finding a minimum size set cover (Problem A.1).

We define the universe $U = \{1, \dots, m\}$ to be the set $\mathcal{T} \setminus \{T\} = \{T_1, \dots, T_m\}$ of trees excluding the tree T for which we want to solve the T -SCS-PC problem. We define the family $\mathcal{F} = \{S_1, \dots, S_n\}$ of subsets to correspond to the n featurettes present in T . Specifically, the subset S_j corresponding to featurette τ_j that is present in T is composed of elements $i \in U$ corresponding to trees T_i where τ_j is absent. The key idea is that S_j is indicating in which input trees featurette τ_j of T is absent. We note that $\mathcal{F}$ is a multi-set as distinct featurettes τ_j and $\tau_{j'}$ may be absent in the same set of trees, thus leading to $S_j = S_{j'}$ (Fig. 2.3). There is a bijection between set covers of $(U, \mathcal{F})$ and distinguishing features of T . That is, each distinguishing feature $\Pi = \{\tau_1, \dots, \tau_{|\Pi|}\}$ corresponds to the same-sized cover $\Pi(C)$ composed of subsets $\{S_1, \dots, S_{|\Pi(C)|}\}$, and vice versa. In particular, a minimal distinguishing feature corresponds to a minimal set cover.

We used the following combinatorial algorithm to exhaustively enumerate all minimal distinguishing features of a given tree T . The algorithm generates all possible features Π of T that are of sizes $i = 1, \dots, m$. Since we enumerate features Π from size smallest to largest, we ensure that every feature $\Pi' \subsetneq \Pi$ was visited before it. Thus, if Π is minimal w.r.t Φ^* , it means that all features $\Pi' \subsetneq \Pi$ that were visited before it were not distinguishing. If Π is also a distinguishing feature of T , we add it to Φ^* . Hence, Φ^* contains all minimal distinguishing features of T . Fig. 2.3 shows an example.

Algorithm A.1: MINIMALDFF(T)

Input: Tree T
Output: Minimal distinguishing feature family Φ^*
 $m \leftarrow$ number of featurettes in T
 $\Phi^* \leftarrow \emptyset$
for $i \leftarrow 1$ **to** m **do**
 for each feature Π containing i featurettes **do**
 if $\Pi(T)$ and Π minimal w.r.t Φ^* **then**
 $\Phi^* \leftarrow \Phi^* \cup \{\Pi\}$
 end
 end
end
return Φ^*

A.2.2 Confidence Interval for k^*

The cancer cell fractions (CCFs) $\mathbf{f}$ of mutations are a key input to PhyDOSE. Specifically, the CCF f_i of a mutation i is estimated from the numbers of variant and reference reads spanning the mutation locus and additional copy number information. Uncertainty in these frequency estimates will impact the design of the follow-up single-cell sequencing experiment. Following a standard approach [34, 72, 75], we construct a confidence interval $[f_i^-, f_i^+]$ of the CCF of mutation i by modeling the number of variants reads as a binomial or beta-binomial distribution using a specified level of confidence $\alpha \in [0, 1]$. Here α is the probability that the true parameter estimate f_i falls outside the constructed confidence interval $[f_i^-, f_i^+]$. Thus, the input to PhyDOSE is composed of two frequency vectors $\mathbf{f}^-$ and $\mathbf{f}^+$.

Recall that PhyDOSE performs a power calculation on the number of cells to sequence such that the probability of successfully observing a distinguishing feature of a given tree $T \in \mathcal{T}$ is at least a given confidence level γ . This probability is modeled as the tail probability of the multinomial distribution where the probabilities are obtained from the clonal prevalence $\mathbf{u}(T, \mathbf{f}) = [u_i]$ of each featurette/clone i in the distinguishing features. These clonal prevalences are uniquely determined by the given tree T and frequencies f under the infinite sites assumption as

$$u_i = f_i - \sum_{j \in \delta_T(i)} f_j \quad \forall i \in [n], \quad (\text{A.1})$$

where $\delta_T(i)$ is the set of children of the node where mutation i was introduced [75]. In other words, to obtain non-negative clonal prevalences, it must hold that $f_i \geq \sum_{j \in \delta_T(i)} f_j$ for all mutations i . Thus, given a candidate set $\mathcal{T}$, a tree $T \in \mathcal{T}$ and a range $[\mathbf{f}^-, \mathbf{f}^+]$ for the cancer

cell fractions, we reduce the problem of finding the k^* confidence interval to one of finding two assignments $[\hat{\mathbf{f}}^-, \hat{\mathbf{f}}^+]$ of frequencies within the given range $[\mathbf{f}^-, \mathbf{f}^+]$ of frequencies that can be utilized directly in the PhyDOSE power calculation. Specifically, $\hat{\mathbf{f}}^-$ corresponds to the theoretical minimum number $k_-^*(T)$ and $\hat{\mathbf{f}}^+$ corresponds to theoretical maximum number $k_+^*(T)$ of single cells for each tree T .

Here we formally state two problems to be solved in order to obtain the $[k_-^*(T), k_+^*(T)]$ confidence interval.

Problem A.2. For a given tree T in candidate set $\mathcal{T}$, frequencies $[\mathbf{f}^-, \mathbf{f}^+]$ and confidence level γ , find frequencies $\hat{\mathbf{f}}^-$ such that (i) $f_i^- \leq \hat{f}_i^- \leq f_i^+$ for each mutation i and (ii) $k_*^- = \operatorname{argmin}_k \Pr(Y_k \mid \mathbf{u}(T, \hat{\mathbf{f}}^-)) \geq \gamma$ is minimum.

Problem A.3. For a given tree T in candidate set $\mathcal{T}$, frequencies $[\mathbf{f}^-, \mathbf{f}^+]$ and confidence level γ , find frequencies $\hat{\mathbf{f}}^+$ such that (i) $f_i^- \leq \hat{f}_i^+ \leq f_i^+$ for each mutation i and (ii) $k_*^+ = \operatorname{argmin}_k \Pr(Y_k \mid \mathbf{u}(T, \hat{\mathbf{f}}^+)) \geq \gamma$ is maximum.

We solve both problems heuristically using the following two simplifications. First, for the sake of a clear exposition, we assume that the distinguishing feature family $\Phi(T, \mathcal{T} \setminus \{T\})$ is composed of a single distinguishing feature Π — we will describe below how this assumption can be relaxed. Second, rather than jointly considering all featurettes, we optimize the featurette in the distinguishing feature that will yield the smallest clonal prevalence from the assigned frequency values $\hat{\mathbf{f}}^-$ and $\hat{\mathbf{f}}^+$. The rationale is that the multinomial tail probability is closely approximated by the binomial tail probability utilizing the minimum probability featurette in the distinguishing feature [50]. In the following, we develop two linear programs to heuristically find the frequencies $\hat{\mathbf{f}}^-$ and $\hat{\mathbf{f}}^+$.

Recall that trees under the infinite sites assumption exhibit a one-to-one correspondence between mutations and clones. In addition, recall that the distinguishing feature Π is composed of a set of distinguishing featurettes/clones. Let $X(\Pi)$ be the set of mutations that identify these clones. To find values $\hat{\mathbf{f}}^-$, we use a linear program with an objective to maxi-

mize the minimum clonal prevalence among all featurettes in the distinguishing feature.

$$\max z \tag{A.2}$$

$$\text{s.t. } \hat{f}_i^- \geq \sum_{v_j \in \delta(v_i)} \hat{f}_j^-, \quad \forall i \in [n], \tag{A.3}$$

$$z \leq \hat{f}_i^- - \sum_{j \in \delta_T(i)} \hat{f}_j^-, \quad \forall i \in X(\Pi), \tag{A.4}$$

$$f_i^- \leq \hat{f}_i^- \leq f_i^+, \quad \forall i \in [n], \tag{A.5}$$

$$z \geq 0, \tag{A.6}$$

where $\delta_T(i)$ is the set of children of the node where mutation i was introduced. The first constraint ensures the frequencies $\hat{\mathbf{f}}^-$ yields a non-negative clonal prevalence for all clones in tree T . The second constraint is to facilitate the objective to maximize the minimum clonal prevalence of the featurettes in the distinguishing feature. The last constraint ensures the assigned frequencies $\hat{\mathbf{f}}^-$ are within the specified frequency confidence interval.

In the second linear program, we find values $\hat{\mathbf{f}}^+$ to heuristically solve Problem A.3 with an objective to maximize the minimum clone prevalence among all featurettes in the distinguishing feature.

$$\min z \tag{A.7}$$

$$\text{s.t. } \hat{f}_i^+ \geq \sum_{v_j \in \delta(v_i)} \hat{f}_j^-, \quad \forall i \in [n], \tag{A.8}$$

$$z \geq \sum_{j \in \delta_T(k)} \hat{f}_j^+ - \hat{f}_k^+, \quad \forall k \in \Pi_j, \tag{A.9}$$

$$f_i^- \leq \hat{f}_i^+ \leq f_i^+, \quad \forall i \in [n], \tag{A.10}$$

$$z \geq 0. \tag{A.11}$$

The first and last constraints are the same as in the previous linear program. In the second constraint, we ensure the problem remains convex by switching the sign of the clonal prevalence values and setting z to be greater than the maximum negative clonal prevalence of all featurettes in the distinguishing feature. We use the `lpSolve`⁹ R package solve these linear programs within the `phydoser` R package.

When a tree has multiple distinguishing features, we solve the two linear programs for each distinguishing feature $\Pi \in \Phi^*$ returning frequencies $\hat{\mathbf{f}}^-$ and $\hat{\mathbf{f}}^+$ that achieve the maximum objective value of the respective linear programs. This is because it is only necessary

⁹<http://lpsolve.sourceforge.net/5.5/>

to observe a single distinguishing feature for an experiment to be successful. Thus, the probability calculation will be dominated by the distinguishing feature where the smallest featureette clonal prevalence is maximum over the smallest featureette clonal prevalence rates of all distinguishing features.

The goal of the confidence interval is to consider uncertainty in the frequency parameter estimates when determining k^* . Therefore, we opt to construct the k^* confidence interval conservatively. Specifically, we first construct a confidence interval for each tree $T \in \mathcal{T}$. Then, we take the confidence interval that corresponds to the tree with the maximum upper end of the confidence interval as the confidence interval for k^* .

A.2.3 Heuristic power calculation for multiple biopsies

Given multiple biopsies from the same tumor, the T -MUL-SCS-PC seeks the numbers $\mathbf{k}^* = [k_1^*, \dots, k_b^*]$ of cells to sequence from each biopsy such that $\|\mathbf{k}^*\|_1$ is minimum.

Here, we develop a heuristic because the exact approach presented in the main text does not scale. Given a frequency matrix $F \in [0, 1]^{b \times n}$, confidence level γ and the minimal distinguishing feature family Φ^* of a tree T with respect to candidate trees $\mathcal{T}$, the heuristic proceeds in two steps in order to select a subset of clones to detect in each biopsy that together form a distinguishing feature in Φ^* and achieve the smallest total number of cells with confidence level γ .

First, we perform a power calculation at confidence level $\gamma^{1/b}$ for each biopsy and subset of featurettes present in Φ^* . Due to independence, a confidence of $\gamma^{1/b}$ in each biopsy will achieve the overall confidence of γ , i.e. $\prod_{p=1}^b \gamma^{1/b} = \gamma$. To define the desired family of subsets of featurettes, let $N = \bigcup_{\Pi \in \Phi^*} \Pi$ be the set of featurettes present in the distinguishing features of Φ^* , and let $\{\Psi(1), \dots, \Psi(2^{|N|})\}$ be the powerset of featurettes N (Fig. A.2). We define $Z_{k, \Psi(i), p}$ as the event of observing the set $\Psi(i)$ of featurettes/clones in an SCDNA-SEQ experiment with k cells from biopsy p . Using the multinomial power calculation outlined previously, we compute the minimum number k_{ip} of cells required to observe the set $\Psi(i)$ from biopsy p such that the probability of success is at least $\gamma^{1/|b|}$. That is,

$$k_{ip} = \underset{k \in \mathbb{N}}{\operatorname{argmin}} \Pr(Z_{k, \Psi(i), p} \mid u(T, \mathbf{f}_p)) \geq \gamma^{1/|b|}. \quad (\text{A.12})$$

Note that for the index i where the corresponding featurette subset $\Psi(i)$ equals the empty set $\emptyset$, we have that $k_{ip} = 0$ as $\Pr(Z_{0, \emptyset, p} \mid u(T, \mathbf{f}_p)) = 1$ for all biopsies p .

Second, given the number k_{ip} for each featurette subset $i \in [2^{|N|}]$ and biopsy p , we now want to pick a subset of featurettes $\Psi(\sigma(1)), \dots, \Psi(\sigma(b))$ for each biopsy p that together cover

at least one distinguishing feature in Φ^* and have the minimum total number $\sum_{p=1}^b k_{\sigma(p),p}$ of cells. In Section A.2.3, we provide a mixed integer linear program (MILP) to solve this problem. To solve the MUL-SCS-PC problem, we take the per biopsy maximum across all solutions to T -MUL-SCS-PC for each $T \in \mathcal{T}$.

This approach is a heuristic in two ways. First, the heuristic does not account for events where we observe a distinguishing feature in ways other than the selected featurettes per biopsy. As such, the heuristic might overestimate the total number of cells that are needed for the specified confidence γ , as we show in Fig. A.2. Second, when solving the MUL-SCS-PC, we only consider a single optimal solution for each $T \in \mathcal{T}$. Indeed, it may be possible to find a solution $\mathbf{k}^*$ with a smaller L_1 norm if we consider the entire set of feasible solutions for each tree.

Integer linear program for selecting featurettes per biopsy. Let the minimal distinguishing feature family Φ^* be composed of distinguishing features $\{\Pi_1, \dots, \Pi_{|\Phi^*|}\}$. The featurettes in Φ^* correspond to $N = \{\tau_1, \dots, \tau_{|N|}\}$. The powerset of featurettes N equals $\{\Psi(1), \dots, \Psi(2^{|N|})\}$. Recall that k_{ip} indicates the minimum number of cells required to observe the featurette set $\Psi(i)$ from biopsy p such that the probability of success is at least $\gamma^{1/|b|}$. Given these values, we develop an integer linear program (ILP) whose objective is to select a subset $\Psi(\sigma(p))$ of featurettes for each biopsy p such that the total number $\sum_{p=1}^b k_{\sigma(p),p}$ of cells is minimum and the selected featurettes cover at least one distinguishing feature in Φ^* . To accomplish this, we need variables and constraints to model the following.

- Assignment of a featurette subset to each biopsy.

Binary variables $\mathbf{x} \in \{0, 1\}^{2^{|N|} \times b}$ indicate the assignment of featurette subsets to biopsies, modeled by (A.14).

- Coverage of a distinguishing feature.

The key constraint is that the selected featurette subsets cover at least one distinguishing feature in Φ^* . To model this constraint, we need several helper variables and constraints.

We begin with binary variables $\mathbf{r} \in \{0, 1\}^{2^{|N|}}$, which indicate that feature subset $\Psi(i)$ is assigned to at least one biopsy. This is modeled by constraints (A.15) and (A.16).

We introduce binary variables $\mathbf{z} \in \{0, 1\}^{|N|}$, which indicate the coverage of featurettes τ_ℓ present in Φ^* . To that end, for a featurette $\tau_\ell \in N$, let $\Psi^{-1}(\ell)$ be the set of indices i corresponding to featurette subsets $\Psi(i)$ that contain featurette τ_ℓ . Constraints (A.17)

and (A.18) use variables $\mathbf{r}$ to model that $z_\ell = 1$ if and only if a featurette subset containing τ_ℓ has been selected.

Next, binary variables $\mathbf{y} \in \{0, 1\}^{|\Phi^*|}$ indicate the coverage of distinguishing features by $\mathbf{x}$. To that end, we use variables $\mathbf{z}$ in constraints (A.19) and (A.20).

Finally, constraint (A.21) uses binary variables $\mathbf{y}$ to require that at least one distinguishing feature of Φ^* is covered.

$$\min \sum_{i=1}^{2^{|N|}} \sum_{p=1}^b k_{ip} x_{ip} \quad (\text{A.13})$$

$$\text{s.t. } \sum_{i=1}^{2^{|N|}} x_{ip} = 1, \quad \forall p \in [b], \quad (\text{A.14})$$

$$r_i \leq \sum_{p=1}^b x_{ip}, \quad \forall i \in [2^{|N|}], \quad (\text{A.15})$$

$$r_i \geq x_{ip}, \quad \forall i \in [2^{|N|}], p \in [b], \quad (\text{A.16})$$

$$z_\ell \leq \sum_{i \in \Psi^{-1}(\ell)} r_i, \quad \forall \ell \in [N], \quad (\text{A.17})$$

$$z_\ell \geq r_i, \quad \forall \ell \in [N], i \in \Psi^{-1}(\ell), \quad (\text{A.18})$$

$$y_j \geq \sum_{\tau_\ell \in \Pi_j} z_\ell - |\Pi_j| + 1, \quad \forall j \in [|\Phi^*|], \quad (\text{A.19})$$

$$y_j \leq z_\ell, \quad \forall j \in [|\Phi^*|], \tau_\ell \in \Pi_j, \quad (\text{A.20})$$

$$\sum_{j=1}^{|\Phi^*|} y_j \geq 1, \quad (\text{A.21})$$

$$x_{ip} \in \{0, 1\}, \quad \forall i \in [2^{|N|}], p \in [b], \quad (\text{A.22})$$

$$r_i \in \{0, 1\}, \quad \forall i \in [2^{|N|}], \quad (\text{A.23})$$

$$z_\ell \in \{0, 1\}, \quad \forall \ell \in [N], \quad (\text{A.24})$$

$$y_j \in \{0, 1\}, \quad \forall j \in [|\Phi^*|]. \quad (\text{A.25})$$

Note that we can relax the integrality constraints (A.23), (A.24) and (A.25) as this will be enforced by integrality of $\mathbf{x}$. Doing so will lead to a speed up. Using the `lpSolveAPI`¹⁰ R package solve to this ILP for each tree T within the `phydoser` R package.

¹⁰<http://lpsolve.sourceforge.net/5.5/>

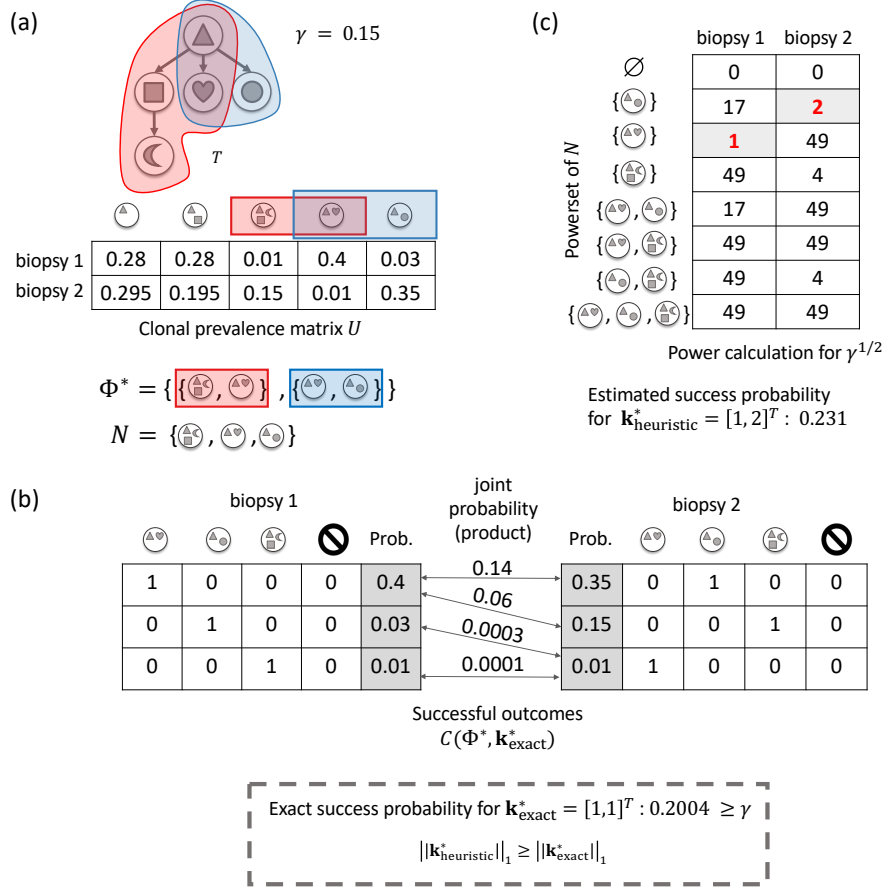


Figure A.2: **Solving the T -MUL-SCS-PC problem** (A) An example instance of the T -MUL-SCS-PC problem with $b = 2$ biopsies, where we are given a confidence level γ , a clonal prevalence matrix U and the minimal distinguishing feature family Φ^* . The set N of featurettes is the union of the distinguishing features in Φ^* . (B) For $\mathbf{k}_{\text{exact}}^* = [1, 1]^T$ cells, there are four outcomes in the set $\mathcal{C}(\Phi^*, \mathbf{k}_{\text{exact}}^*)$ of successful outcomes. The probability of each successful outcome is the joint probability of observing the required number of cells in each biopsy. Here, the probability $\Pr(Y_{\mathbf{k}_{\text{exact}}^*} | U)$ of success equals 0.2004, which is greater than $\gamma = 0.15$. Moreover, $\Pr(Y_{[1,0]^T} | U) = \Pr(Y_{[0,1]^T} | U) = 0$ as each distinguishing feature has at least two featurettes. Hence, $\mathbf{k}_{\text{exact}}^* = [1, 1]^T$ is the optimal solution to this T -MUL-SCS-PC problem instance. (C) The heuristic performs a power calculation at confidence level $\gamma^{1/b} = \gamma^{1/2}$ for each element of the powerset of N and each biopsy. Then, an ILP assigns a featurette set to each biopsy minimizing the total number of cells while covering at least one distinguishing feature. Here, the heuristic solution is $\mathbf{k}_{\text{heuristic}}^* = [1, 2]^T$, which is worse than the exact calculation.

A.3 SUPPLEMENTARY RESULTS

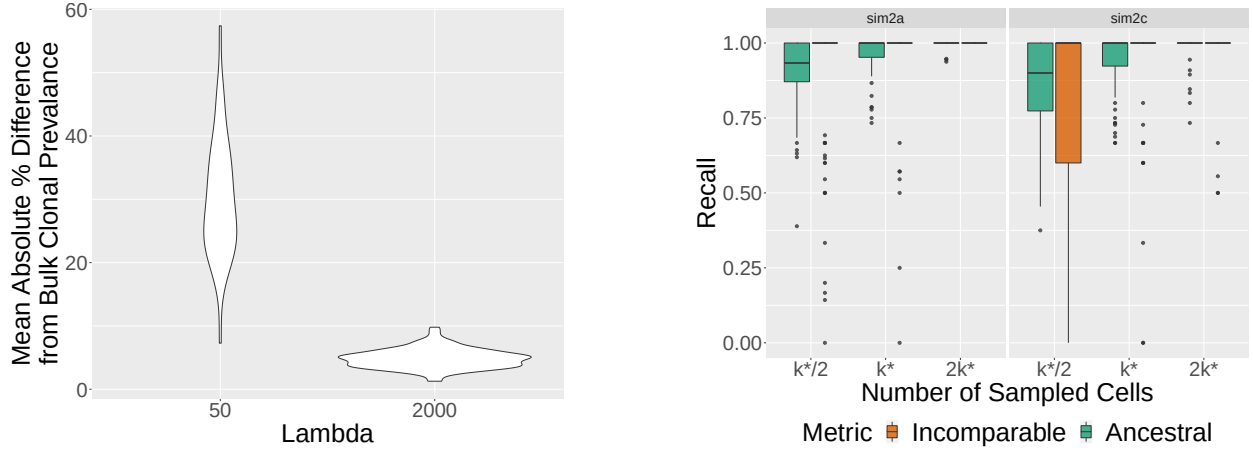


Figure A.3: **PhyDOSE sensitivity analysis of clonal prevalence distortion** (A) The mean absolute percentage difference of the single cell clonal prevalence per replication from simulated bulk clonal prevalence at values of $\lambda \in \{50, 2000\}$. These values of λ resulted in mean absolute percentage difference of 5% and 20% respectively. (B) $\lambda = 2000$ was selected for sim2a/b but further sensitivity analysis was performed with $\lambda = 50$. The recall metrics are compared between sim2a at $\lambda = 2000$ and sim2c at $\lambda = 50$. (C) Recall metrics when inferring T^* with SPhyR [42] by randomly sampling $k^*/2, k^*, 2k^*$ simulated single cells.

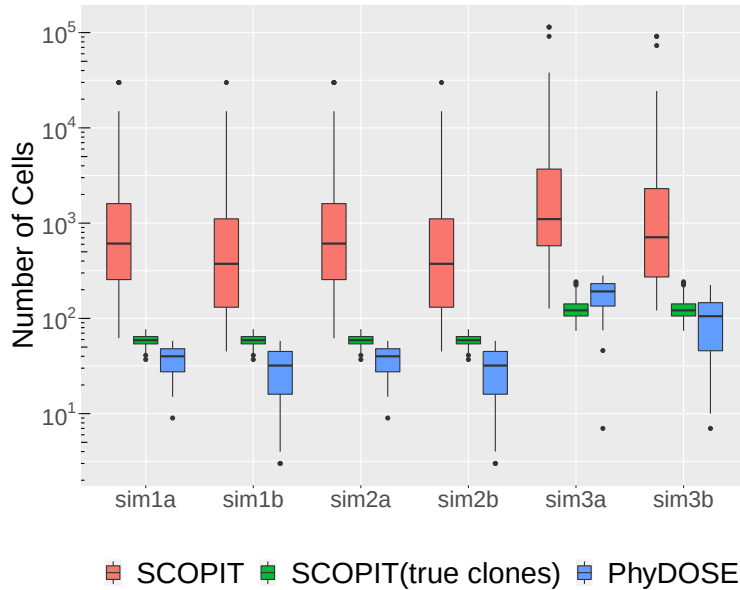


Figure A.4: **PhyDOSE comparison with SCOPIT** The distribution of the recommended number of cells to sequence per simulated replication determined by SCOPIT, SCOPIT using the true clones, and PhyDOSE.

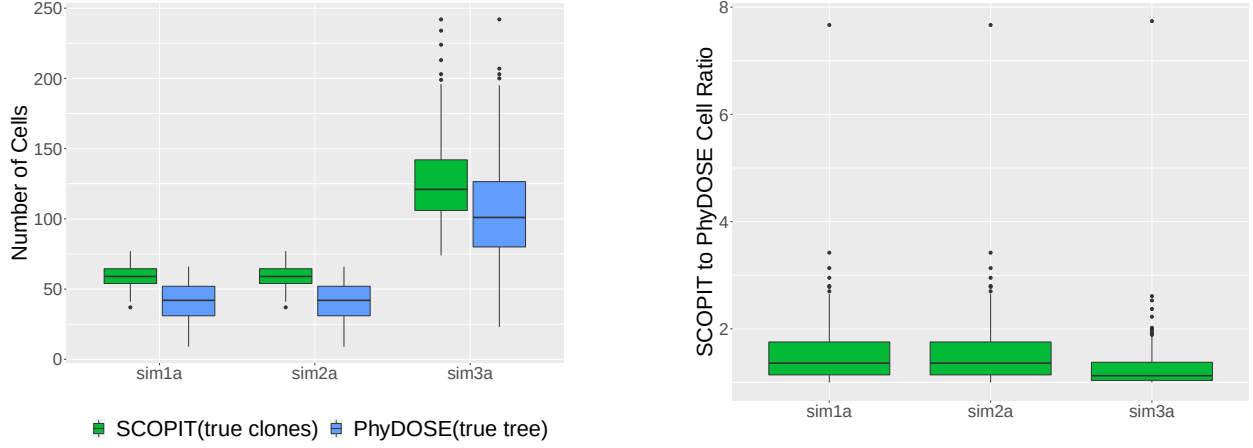


Figure A.5: **PhyDOSE $k^*(T^*)$ comparison with SCOPIT (true clones)** (A) The distribution of the recommended number of cells to sequence per simulated replication determined by SCOPIT when given the ground truth clones and PhyDOSE $k^*(T^*)$. (B) The cell ratio per simulation between SCOPIT (true clones) and PhyDOSE $k^*(T^*)$.

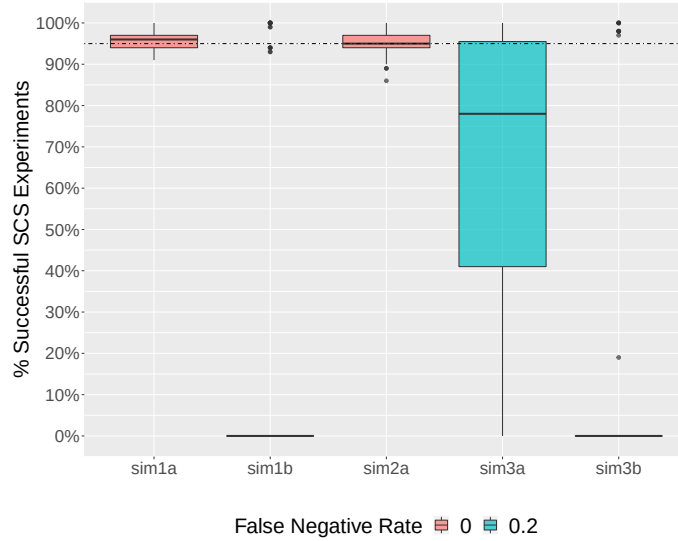


Figure A.6: **Percentage of *in silico* single-cell sequencing experiments correctly identifying T^* as the true tree utilizing PhyDOSE's k^* at $\gamma = 0.95$ and $\beta \in \{0.0, 0.2\}$.** PhyDOSE has a high success rate in alignment with $\gamma = 0.95$ (dashed horizontal line) when utilizing the entire set $\mathcal{T}$ in condition a . However, the introduction of false negatives results in a reduction in success rate and greater variance. In the cases where $T^* \in \mathcal{T}$, the support metric was still able to successfully identify T^* as the true tree.

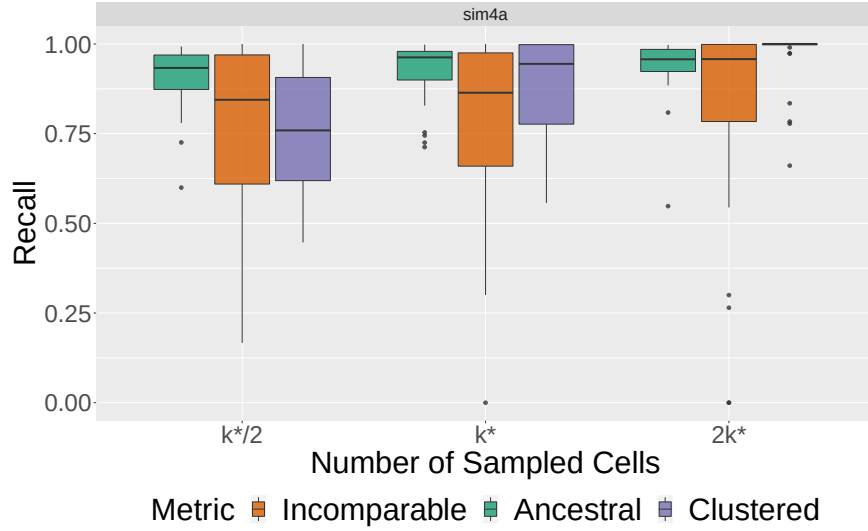


Figure A.7: **PhyDOSE performance with mutation clustering** Recall metrics when inferring T^* with SPhyR [42] by randomly sampling $k^*/2, k^*, 2k^*$ simulated single cells.

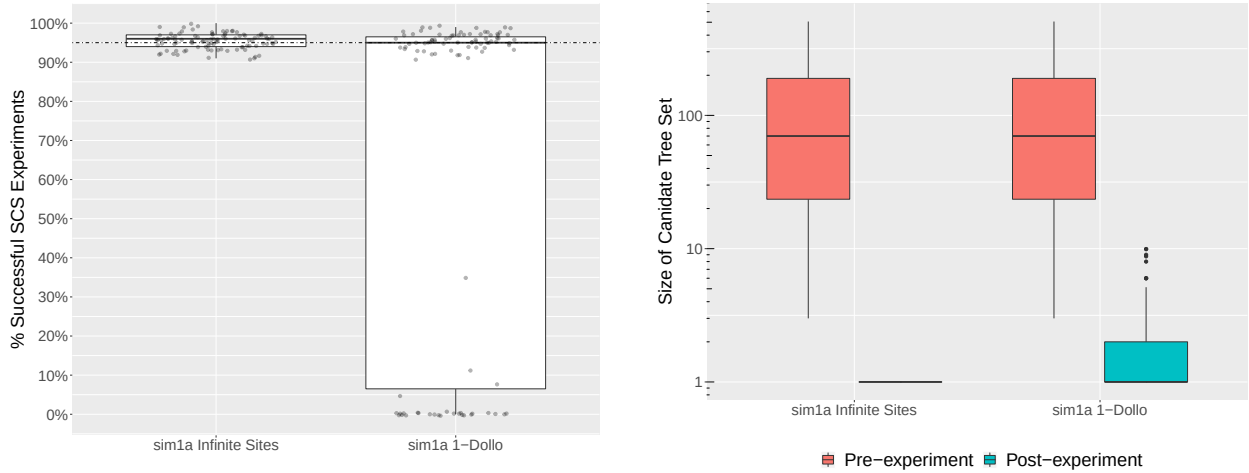


Figure A.8: **Assessment of PhyDOSE with respect to violations of the infinite sites assumption.** (A) Percentage of *in silico* single-cell sequencing experiments correctly identifying T^* as the true tree utilizing PhyDOSE's k^* at confidence level $\gamma = 0.95$ and false negative rate $\beta = 0$ under the infinite sites model and the 1-Dollo model. PhyDOSE has a high success rate in alignment with $\gamma = 0.95$ under both models of evolution. The variance in the success rate in the 1-Dollo model can be attributed to whether or not the distinguishing features are impacted by the mutation loss. When the distinguishing features are no longer distinguishing, then the experiment will be unsuccessful. (B) The average number of trees in the candidate set $\mathcal{T}$ pre and post execution of an experiment designed by PhyDOSE. The post-experiment candidate set $\mathcal{T}$ contains all trees T that yielded $\text{support}(T) > 0$ during the experiment. Even in cases where the experiment does not uniquely identify T^* as the true tree, the size of the candidate set is still significantly reduced as compared to the input candidate set.

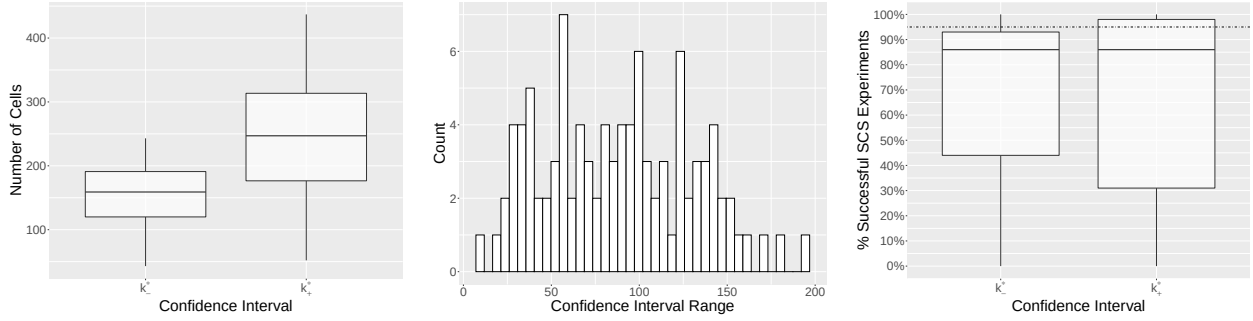


Figure A.9: **Computation and evaluation of the k^* confidence interval** (A) Distribution of the number of cells computed by PhyDOSE at confidence level $\gamma = 0.95$ for the k^* confidence interval $[k_-^*, k_+^*]$ in 100 simulations. (B) The distribution of the k^* 95% confidence interval range ($k_+^* - k_-^*$) over 100 simulations. (C) Distribution of the percentage of successful *in silico* SCDNA-SEQ experiments (100 experiments per simulation) when randomly sampling $k \in \{k_-^*, k_+^*\}$ cells computed by PhyDOSE at confidence level $\gamma = 0.95$ (dashed line) in 100 simulations.

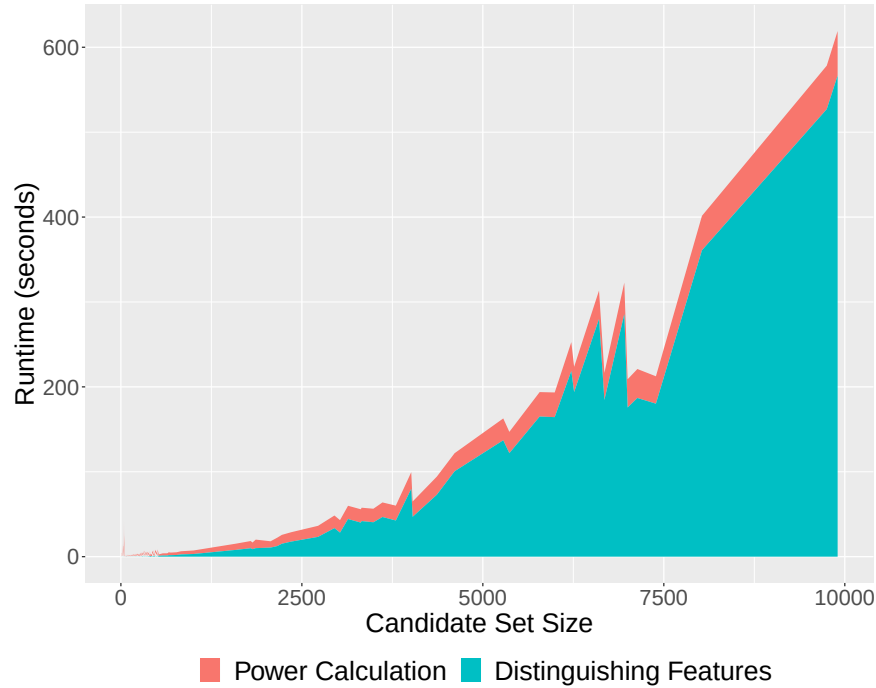


Figure A.10: **Empirical runtime analysis of PhyDOSE** Total runtime in seconds of PhyDOSE as a function of the input set $\mathcal{T}$ and broken down into the PhyDOSE phases of finding the distinguishing features and then performing the power calculation.

A.3.1 Retrospective analysis of an acute lymphoblastic leukemia patient

For each patient, the sequenced cells are clustered into 2 to 7 clones using an EM-based approach [85]. Based on the fact that false negatives occur more frequently than false positives, we designated an SNV as present if at least 30% of cells in the clone had the mutation. We then checked if the resulting binary, clone-by-SNV matrices adhered to the infinite sites assumption, which was the case for only patients 2 and 3. While the VAFs of all 16 SNVs in patient 2 are less than 0.5, patient 3 had 6 out of 49 SNVs with a VAF larger than 0.5, which is indicative of copy number aberrations. Since no copy number information was available to infer cancer cell fractions, we excluded patient 3 from our analysis, thus restricting our attention to patient 2.

For patient 2, 115 cells were clustered into 5 clones [85]. This patient has 16 SNVs, from which we excluded mutations *CMTM8*, *ATRNL1*, *LINC00052* and *TRRAP* for reasons that we described in the main text. The majority voting rule described above yielded a binary clone-by-SNV matrix with 4 mutations clusters that each correspond to SNVs that co-occur in every clone (Fig. A.11a), corresponding to a two-state perfect phylogeny T_{SCS} on the mutation clusters (Fig. 2.5B). To obtain the set $\mathcal{T}$ of candidate phylogenies, we considered the bulk data. Specifically, we merged mutations ZC3H3 and XPO7 as they had the same VAF in the bulk data and occurred in the same mutation cluster in the cleaned sCDNA-SEQ data (Fig. 2.5). Using SPRUCE [75], we enumerated $|\mathcal{T}| = 2576$ trees (Fig. A.11b). Only one tree $T^* \in \mathcal{T}$ was consistent with T_{SCS} , i.e. each mutation cluster of T_{SCS} formed a connected path in T^* and subsequently collapsing these paths in T^* yields T_{SCS} . Comparing the cleaned single-cell data to the raw values, we computed a false negative rate β of 0.2 for the 14 mutations (Fig. A.11a), which was in line with the value reported by [85].

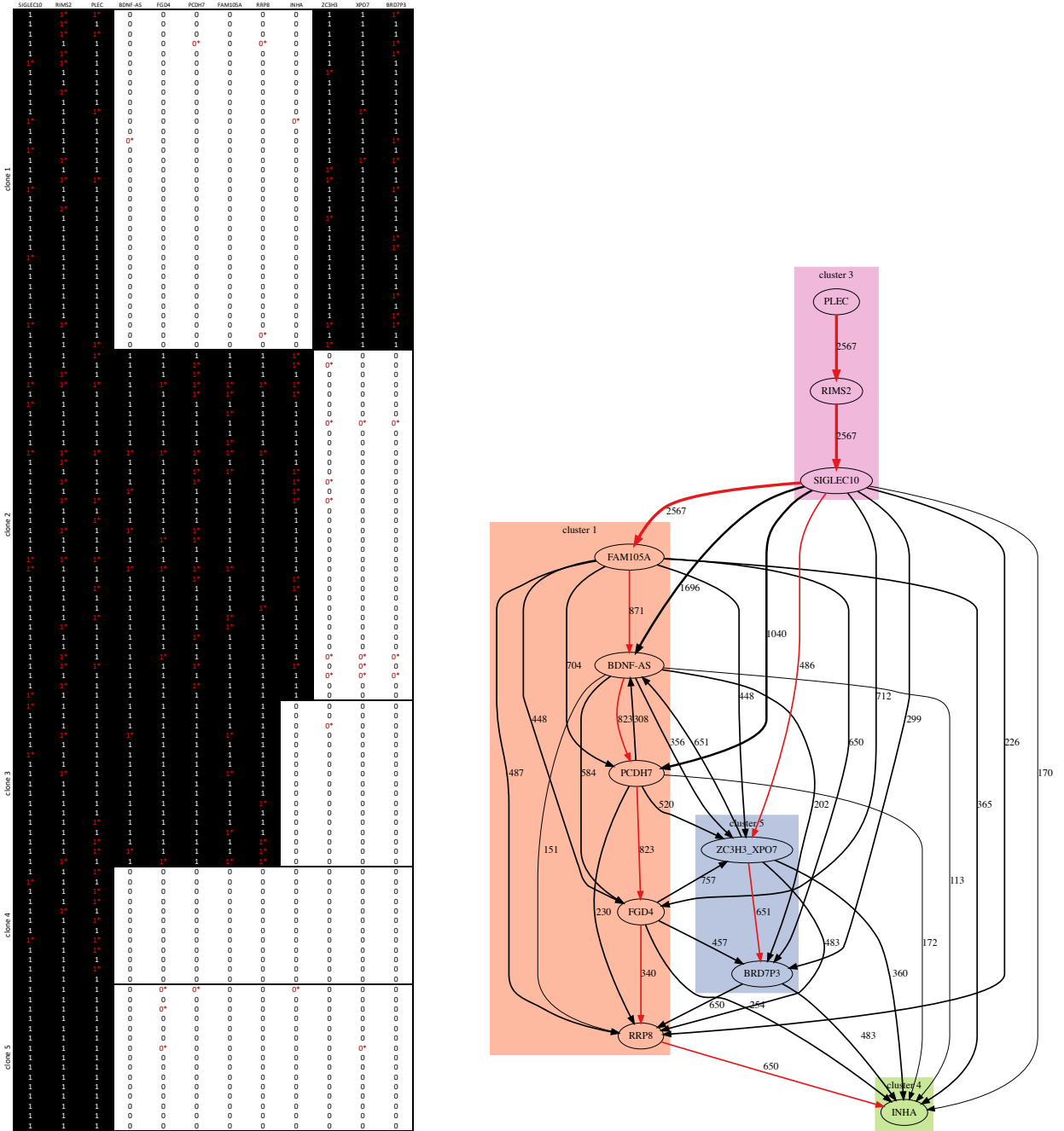


Figure A.11: **Retrospective analysis of ALL patient 2 where [85] sequenced 115 single cells, targeting 16 mutations.** (A) Overview of single-cell data, focusing on a subset of 14 mutations. Each row is a single cell and columns correspond to mutations. Black cells indicate presence of the SNV, whereas white cells indicate absence. Text in each cell indicates a true positive ('1'), false negative ('1*'), true negative ('0') or false positive ('0*') entry. (B) Overview of the set $\mathcal{T}$ of trees, where each edge is labeled by its frequency in $\mathcal{T}$. Red edges encode the tree T^* that is consistent with SCDNA-SEQ data in (a). In particular, the SCDNA-SEQ data encodes a tree composed of four mutation clusters, each indicated by a different shading.

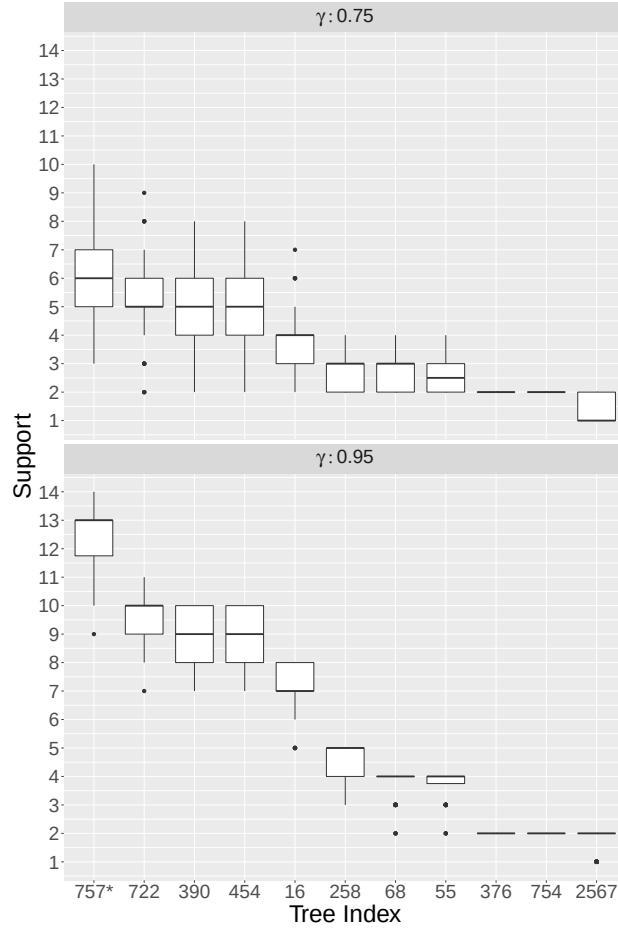


Figure A.12: Distribution of number of supporting cells for each candidate tree across 100 *in silico* sCDNA-SEQ experiments at varying success probabilities for **ALL patient 2** [85]. For $\gamma = 0.75$, we sampled $k(T^*) = 50$ cells from the sCDNA-SEQ data. For $\gamma = 0.95$, we sampled $k(T^*) = 103$ cells. Fig. A.13 shows the candidate trees.

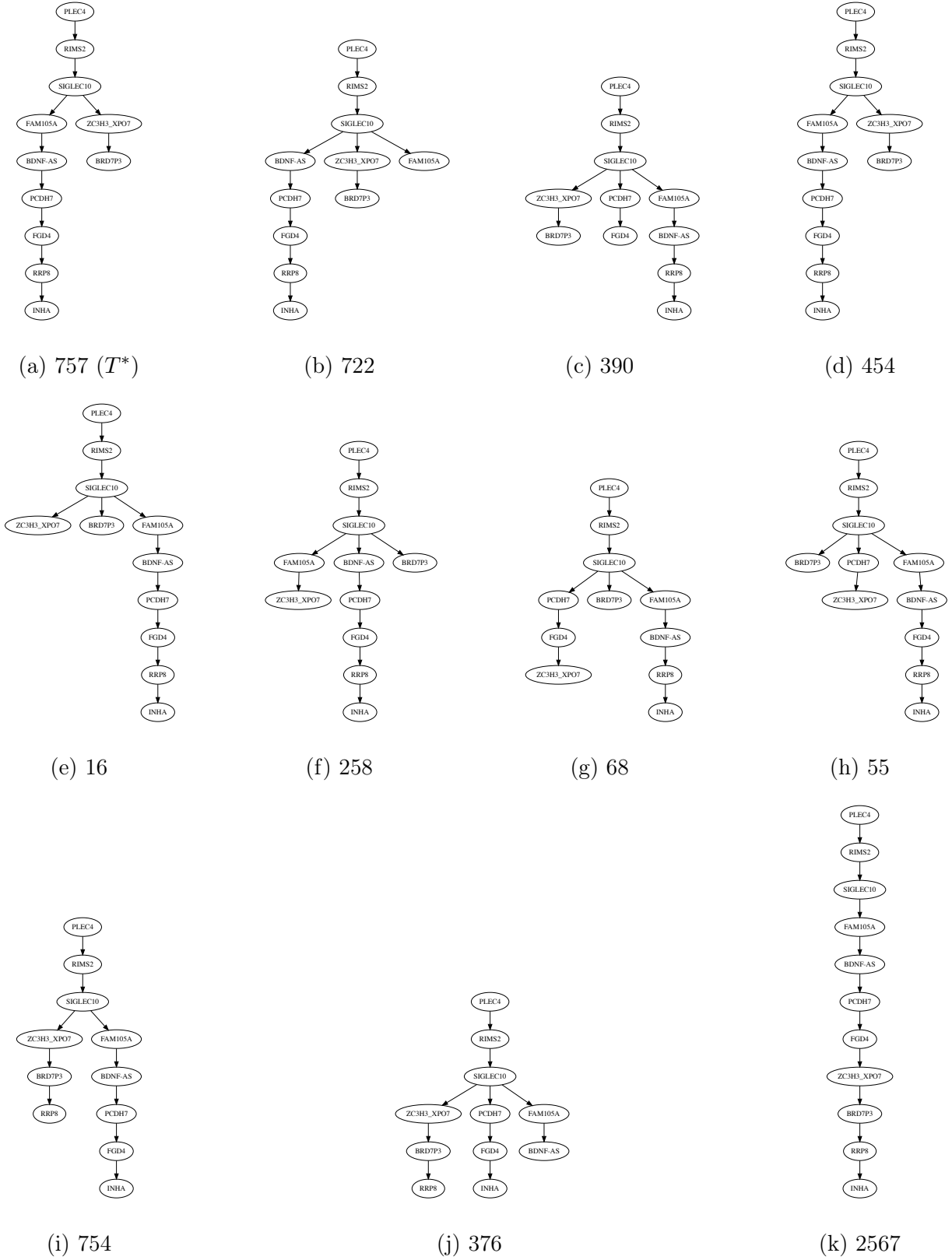


Figure A.13: **Candidate trees that had a non-zero support across 100 *in silico* scDNA-SEQ experiments for ALL patient 2 [85].** For $\gamma = 0.75$, we sampled $k(T^*) = 50$ cells from the scDNA-SEQ data. For $\gamma = 0.95$, we sampled $k(T^*) = 103$ cells. Labels match tree indices in Fig. A.12.

patient	mutations	total sequenced [88]	$ \mathcal{T} $	$k_{\gamma=0.75}^*$ (% red.)	SCOPIT $\gamma = 0.75$	$k_{\gamma=0.95}^*$ (% red.)	SCOPIT $\gamma = 0.95$
AML-4-001	5	5878	8	451 (92.3%)	505	973 (83.4%)	973
AML-5-001	4	10191	8	396 (96.1%)	1156	730 (92.8%)	2463
AML-14-001	3	7053	4	129 (98.2%)	574	278 (96.1%)	1018
AML-22-001	6	2925	57	209 (92.9%)	5322	402 (86.3%)	11500
AML-26-001	5	9601	24	306 (96.8%)	489	590 (93.9%)	916
AML-29-001	3	7795	2	78 (99.0%)	78	169 (97.8%)	169
AML-30-001	3	6668	2	83 (98.6%)	605	178 (97.3%)	1307
AML-32-001	4	8241	2	7 (99.9%)	75	14 (99.8%)	161
AML-36-001	3	8153	2	246 (97.0%)	256	531 (93.5%)	531
AML-38-001	8	7235	316	366 (95.0%)	5424	692 (90.4%)	10794
AML-39-001	4	6735	6	56 (99.2%)	101	120 (98.2%)	193
AML-40-001	3	6194	2	60 (99.0%)	60	128 (98.0%)	128
AML-41-001	7	5082	89	413 (91.9%)	34837	1155 (83.0%)	75281
AML-42-001	3	2252	2	125 (94.4%)	125	270 (88.0%)	270
AML-45-001	3	7716	2	29 (99.7%)	30	62 (99.2%)	62
AML-51-001	3	8219	2	216 (97.3%)	216	467 (94.3%)	467
AML-59-001	3	2662	2	41 (98.5%)	41	88 (96.7%)	88
AML-61-001	4	4320	2	64 (98.5%)	352	138 (96.8%)	643
AML-62-001	3	4207	2	9 (98.5%)	27	20 (99.5%)	55
AML-64-001	4	4999	3	225 (95.5%)	301	471 (95.5%)	550
AML-67-001	5	6024	2	21 (99.7%)	55	44 (99.2%)	91
AML-69-001	3	7462	2	202 (97.3%)	202	436 (94.2%)	436
AML-75-001	3	6073	2	20 (99.7%)	50	42 (99.3%)	102
AML-76-001	4	8033	3	196(97.5%)	211	423 (94.7%)	423

Table A.1: **Prospective analysis of an acute myeloid leukemia (AML) cohort.** Table shows the patient identifier, the number of mutations, the total cells sequenced [88], the size of the candidate set $\mathcal{T}$ of trees as determined by SPRUCE [75], PhyDOSE’s k^* calculated at confidence levels of $\gamma \in \{0.75, 0.95\}$ (% reduction from total sequenced), and comparison with SCOPIT at confidence levels of $\gamma \in \{0.75, 0.95\}$.

patient	clusters	samples	min muts	max muts	$ \mathcal{T} $	k^* single sample	k^* mul. sample	sel. sample
CRUK0004	7	4	10	78	2	32	32	R3
CRUK0005	6	4	24	536	2	26	26	R3
CRUK0011	8	3	12	335	3	34	34	R2
CRUK0012	5	2	11	84	2	15	15	R1
CRUK0013	9	5	5	114	8	1051	215	LN2=172 R3=43
CRUK0022	5	2	5	158	2	23	23	R1
CRUK0023	10	4	5	226	2	20	20	R1
CRUK0025	7	3	10	350	2	25	25	R2
CRUK0028	5	2	5	72	2	33	33	R1
CRUK0031	7	3	15	675	2	30	30	R1
CRUK0037	10	5	6	397	17	∞	234	R1=36 R2=49 R3=60 R4=38 R5=51
CRUK0038	4	2	6	107	2	43	43	R1
CRUK0046	5	4	5	186	2	29	29	R1
CRUK0049	6	2	7	882	4	37	37	R2
CRUK0063	8	5	5	167	2	33	33	R4
CRUK0067	5	2	24	263	2	29	29	R1
CRUK0068	10	4	11	532	3	35	35	R2
CRUK0070	10	5	6	254	2	24	24	R6
CRUK0076	9	4	8	846	4	47479	48	R1=23 R2=25
CRUK0077	7	4	9	586	2	22	22	R4
CRUK0084	6	4	5	332	2	17	17	R2
CRUK0094	6	4	6	50	2	22	22	R4
CRUK0095	4	3	6	216	2	42	42	R2
CRUK0099	5	4	5	438	2	46	46	R6
CRUK0100	8	3	5	777	3	24	24	R2

Table A.2: **Prospective analysis of TRACERx non-small-cell lung cancer cohort with confidence level $\gamma = 0.95$.** Table shows the patient identifier, the number of mutation clusters, the number of bulk samples, the minimum number of mutations per cluster, the maximum number of mutations per cluster, the size of the candidate set $\mathcal{T}$ of trees as determined by [89], PhyDOSE’s k^* calculated at confidence levels of $\gamma = 0.95$ for the minimum single sample and the multiple sample heuristic and the recommended sample label from which the single cells should be drawn.

APPENDIX B: SUPPLEMENTARY METHODS AND RESULTS FOR DOUBLET D

B.1 SUPPLEMENTARY METHODS

- Table B.1 shows the value of $P(y_{i,j} \mid x_{i,j}, z_i)$ for all possible combinations of post-ADO VAF $y_{i,j}$, pre-ADO VAF $x_{i,j}$ and doublet status z_i .
- Fig. B.1 gives a schematic representation of possible outcomes of ADO for two example droplets.

$x_{i,j} \backslash y_{i,j} \backslash z_i$		0	1/4	1/3	1/2	2/3	3/4	1	NaN
0	0	$1 - \beta^2$	0	0	0	0	0	0	β^2
1/2	0	$\beta(1 - \beta)$	0	0	$(1 - \beta)^2$	0	0	$\beta(1 - \beta)$	β^2
1	0	0	0	0	0	0	0	$1 - \beta^2$	β^2
0	1	$1 - \beta^4$	0	0	0	0	0	0	β^4
1/4	1	$\beta(1 - \beta)^3 + 3\beta^2(1 - \beta)^2 + 3\beta^3(1 - \beta)$	$(1 - \beta)^4$	$3\beta(1 - \beta)^3$	$3\beta^2(1 - \beta)^2$	0	0	$\beta^3(1 - \beta)$	β^4
1/2	1	$\beta^2(1 - \beta)^2 + 2\beta^3(1 - \beta)$	0	$2\beta(1 - \beta)^3$	$(1 - \beta)^4 + 4\beta^2(1 - \beta)^2$	$2\beta(1 - \beta)^3$	0	$\beta^2(1 - \beta)^2 + 2\beta^3(1 - \beta)$	β^4
3/4	1	$\beta^3(1 - \beta)$	0	0	$3\beta^2(1 - \beta)^2$	$3\beta(1 - \beta)^3$	$(1 - \beta)^4$	$\beta(1 - \beta)^3 + 3\beta^2(1 - \beta)^2 + 3\beta^3(1 - \beta)$	β^4
1	1	0	0	0	0	0	0	$1 - \beta^4$	β^4

Table B.1: This table shows the value of $P(y_{i,j} \mid x_{i,j}, z_i)$, i.e. the probability of having VAF $y_{i,j}$ at locus j in droplet i after allelic dropout (ADO) given pre-ADO VAF $x_{i,j}$ and doublet status z_i . The last column 'NaN' represents the case when all the alleles are dropped and, as a result, no reads span locus j in droplet i . The values in each row sum to 1. Schematic representations of allelic dropout in a singlet and a doublet droplet are shown in Fig. B.1.

¹¹This chapter contains identical material with an adapted introduction previously published in “doubletD: detecting doublets in single-cell DNA sequencing data” [101]. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with M. El-Kebir and P. Sashittal, MEK conceived the project; LLW, MEK and PS establish theory; LLW performed experiments; LLW, MEK and PS contributed to the writing of this chapter.

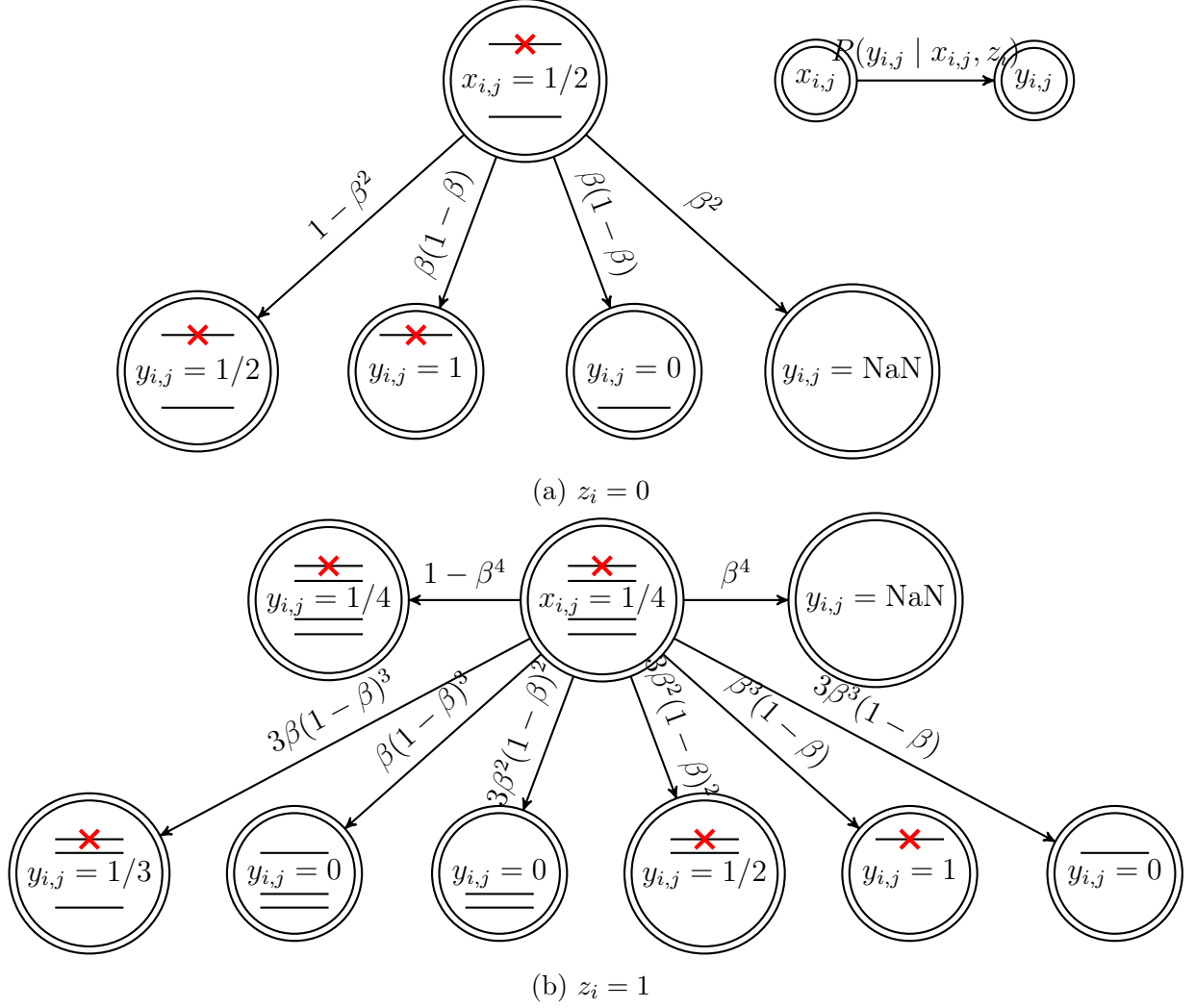


Figure B.1: Schematic representation of allelic dropout in a droplet with (a) single cell with a heterozygous mutation and (b) doublet composed of two cells, one with a heterozygous mutation and other with two wild-type alleles. The circle represents the droplet and the black lines inside the droplet indicate copies of the genome. We consider all cells to be diploid and therefore each cell contributes two copies of the genome. The red cross represents a variant allele. The expression on the arrows depict the probabilities $P(y_{i,j} | x_{i,j}, z_i)$. The values of $P(y_{i,j} | x_{i,j}, z_i)$ are shown in Table B.1.

B.1.1 Parameter estimation

DOUBLETD requires the user to input mutation probabilities μ_{wt} , μ_{het} and μ_{hom} at each locus j used in the genotype model (Section 2.1.2), and the precision parameter s used in the read count model (Section 2.1.4). In this section we describe a data-driven approach to estimate these parameters.

Due to the evolutionary pressures on the cells in the sample, the rate of mutations can change significantly across loci. We therefore use the data to get the mutation probabilities $\mu_{\text{wt},j}$, $\mu_{\text{het},j}$ and $\mu_{\text{hom},j}$ for each locus j which serve as the input parameters for the genotype model (Section 2.1.2). The observed VAFs $v_{i,j}/c_{i,j}$ for each droplet i at locus j are mapped to the closest value in Σ_{singlet} . For $x \in \Sigma_{\text{singlet}}$, let $d_j(x)$ be the number of cells in which the VAF at site j was mapped to x . We estimate the mutation rates as follows,

$$\mu_{\text{wt},j} = \frac{d_j(0)}{d_j(0) + d_j(1/2) + d_j(1)}, \quad (\text{B.1})$$

$$\mu_{\text{het},j} = \frac{d_j(1/2)}{d_j(0) + d_j(1/2) + d_j(1)}, \quad (\text{B.2})$$

$$\mu_{\text{hom},j} = \frac{d_j(1)}{d_j(0) + d_j(1/2) + d_j(1)}. \quad (\text{B.3})$$

Fig. B.8a shows that this method gives reliable estimates of the mutation probabilities in simulations.

To estimate the precision parameter s for the beta-binomial distribution, the observed VAFs $v_{i,j}/c_{i,j}$ for each droplet i at locus j are mapped to a value in Σ_{singlet} . For each $x \in \Sigma_{\text{singlet}}$, let ω_x be the set of observed VAFs mapped to x . We first fit shape parameters $\hat{\alpha}_x$, $\hat{\beta}_x$ for each $x \in \Sigma_{\text{singlet}}$ set ω_x , utilizing method of moments estimation [201] and obtain observed precision $\hat{s}_x = \hat{\alpha}_x + \hat{\beta}_x$. Since our method utilizes a global precision parameter for all droplets and loci, we set the precision parameter to the median of the set $\{\hat{s}_x \mid x \in \Sigma_{\text{singlet}}\}$. The estimation of this parameter can be supplemented from non-variant loci or SNP positions in addition to the observed VAFs. Fig. B.8b shows that we recover reliable estimates of the precision parameter s in simulations. See [202] for an alternative estimation procedure and MDA specific shape parameters that scale linearly with sequencing coverage.

B.2 SUPPLEMENTARY RESULTS

B.2.1 Simulations

Simulation setup We generate variant $\mathbf{V}$ and total read counts $\mathbf{C}$ for 500 *in silico* droplets as follows. First, we evolve 10 genotypes under an evolutionary model that incorporates CNAs and SNVs (detailed below) and use a symmetric Dirichlet distribution to obtain clonal abundances (concentration parameter $\alpha = 2$). The minimum allowable clonal abundance for any genotype was 0.02, which we enforce using rejection sampling. We vary the number of SNVs $m \in \{10, 50, 100\}$. Next, we decided for each droplet whether it is a doublet with

probability $\delta \in \{0.1, 0.2, 0.4\}$. Depending on the outcome, we randomly sample one or two cells from the 10 genotypes to comprise the droplet. We also vary the mean sequencing coverage $c \in \{10, 50, 100\}$ and ADO probability $\beta \in \{0.0, 0.05, 0.25\}$. To draw total reads $\mathbf{C}$, we use a negative binomial distribution parameterized by the mean sequencing coverage c and a dispersion of 5. The copy error rate was fixed at $\alpha_{\text{fp}} = \alpha_{\text{fn}} = 0.001$ for all experiments in accordance with [102]. Finally, we set the beta-binomial precision to $s = 15$ — a simulation regime that matches the Tapestry platform by Mission Bio (Section 3.2 as well as MDA-based single-cell DNA sequencing (Section 3.3)). Using the drawn total read counts $\mathbf{C}$ and the above parameters, we draw variant read counts $\mathbf{V}$. Each combination of simulation parameters was replicated with five different random number generator seeds, amounting to a total of 405 experiments.

Evolutionary model. We now describe the model that we used to simulate the evolutionary history of $k = 10$ genotypes comprised of both CNAs and SNVs. Using Prüfer sequences [95], we begin by drawing a labeled tree T comprised of 10 nodes uniformly at random. For each of the m SNVs, we decide whether that mutation will undergo copy-neutral loss of heterozygosity (CN-LOH) with a probability of 0.1 (as described above). Then, we uniformly assign each mutation to a node of T to generate mutation clusters when $m > k$. For each non CN-LOH mutation, we decide with probability γ if that mutation will undergo a CNA event. These define a set of CNA events. For the set of CNA events, we then decide if that event is a loss with probability ℓ or a gain otherwise. For any gain events, we determine the number of copies gained by drawing a number uniformly between 1 and the max number of copies (3 for base simulations and 5 for our extreme CNA scenario). Next, we randomly assign CNA events to nodes of the tree T . We then generate the set of k genotypes by evolving SNV events (het or hom) and CNV events down the tree. We start with a pair (ω_i, ρ_i) representing the number of variant alleles and reference alleles of each SNV locus $i \in [m]$. At each node, we update this genotype in accordance with the events encountered on the path from the root to the node. When applying CNA events, if the current number of mutated alleles is non-zero, we select either the mutated or reference allele to undergo the CNA event. Otherwise, it is applied to the reference allele. If an SNV and CNA are introduced at the same time, we randomly determine which event to apply first.

Previous methods. We benchmarked our method against scrublet [51] and SCG [40]. We provided scrublet with our simulated variant read counts and doublet rate as input. For SCG, we discretized the input by applying the binomial exact test with 0.001 and 0.999 as the null rates for input discrete states wild type=0 and cn-LOH=2 respectively with p-

values of 10^{-6} required to reject the null hypothesis in favor of the state heterozygous=1 in accordance with [40]. Total read depths of 0 were assigned to a missing state=3. For the symmetric Dirichlet prior, we set $\kappa = 2$ and set the doublet rate prior in accordance with the simulated doublet rate δ . All remaining parameters and inputs were left as the default values.

Sensitivity to input parameters. Our method takes several parameters as input (Fig. 2). The ADO probability β and the sequencing error rates α_{fp}, α_{fn} are sequencing platform specific and typically known a priori. While the doublet probability δ is also typically known beforehand, we find that maximum likelihood is good criterion for estimating this parameter in case it is unknown. Specifically, varying the doublet probability in $\{0.01, 0.1, 0.2, 0.4, 0.7, 0.9\}$ and selecting value with maximum likelihood, we achieve similar precision and recall values as using the ground truth doublet rate (Fig. B.5). Further, the likelihood function is maximized at or close to the simulated doublet rate indicating that it can be used as a reliable criteria for estimating the true doublet rate (Fig. B.6).

doubletD estimates the beta-binomial precision parameter s and mutation probabilities μ from input data. We find that procedure outlined in Section B.1.1 for estimation the beta-binomial precision parameter s and mutation probabilities μ from input data results in only minor error with a deviation of 0.08, -0.11 , 0.02, and 6.0 for μ_{wt} , μ_{het} and μ_{hom} (Fig. B.8a) and s (Fig. B.8a), respectively. We then fixed the inference doublet rate at the simulated rate and varied the beta precision by inputting half and twice the simulated beta precision into doubletD (Fig. B.7). We found that halving or doubling in inference beta precision parameter with respect to the simulated parameter had no significant impact when the simulated precision was high (1000). When the simulated precision was low (15), we noted that overestimation by utilizing double the simulated precision parameter did result in a small decrease in the median precision of 0.86 to 0.81 (Fig. B.7). Conversely, utilizing half the simulated rate resulted in improved precision without a reduction in recall. Thus, in the presence of uncertainty of this parameter, preference should be given to underestimation or lower values. In conclusion, we found that doubletD is resilient to variations in the user-inputted doublet rate parameter, especially in the range of typical experimental doublet rates (0.1-0.4). As the inference doublet rate increases beyond this range, precision is reduced since the threshold for calling an experiment a doublet is significantly lowered. The likelihood function calculated as a function of the predicted experimental labels is maximized at or close to the simulated doublet rate (Fig. B.5).

B.2.2 Two cell line mixture

We followed the procedure outlined in the vignette accompanying this dataset¹². The one notable exception is that we used relaxed filtering criteria to identify additional loci for orthogonal doublet validation. The filtering criteria were as follows.

```
gt.filter=TRUE, gt.gqc = 30, \  
gt.dpc = 10, gt.afc = 20, \  
gt.mv = 50, gt.mc = 50, \  
gt.mm = 0.5, gt.mask = TRUE
```

We extracted variant and total read counts from the AD and DP layers of the loom file for 1592 droplets. To identify a subset of 26 high-quality inference loci among the total number of 133 loci, we excluded loci that had a copy number greater than 2 (using the `compute_ploidy` function in the Tapestri R package).

Upon performing a preliminary dimensionality reduction (t-sne) and hierarchical clustering on the ‘zygosity’ (binning of VAFs into homozygous, heterozygous or wild type) of all 133 loci, we noted the existence of a third cluster composed of 23 droplets with a distinct genotype and mutually exclusive mutations from the other two cell lines. We excluded these droplets from our analyses. Further, we identified a set of 5 validation loci distinct from the 23 inference loci that were homozygous (hom) in one cell line cluster but wild type (wt) in the other. We use these loci to establish the ground truth assignment of droplets to the two cell lines, Raji and KG-1, and to compute the NCS score for each droplet.

B.2.3 Acute lymphoblastic leukemia tumors

Preprocessing. We utilized SCDNA-SEQ data in the form of FASTQ files from the Sequence Read Archive database (accession no. SRP044380). After adaptor trimming (Trimomatic), read alignment (bwa) to reference genome hg19 and PCR duplicate removal (Picard), we performed a pileup of the variant positions called by [120] to obtain the variant $\mathbf{V}$ and total read counts $\mathbf{C}$.

Doublet detection. To mitigate the impact of CNAs on doublet detection, we only included loci that were heterozygous and had a median VAF $\in [0.45, 0.55]$. We ran doubletD directly on the total and variant read count data obtained from the pileup utilizing the ADO rates reported by [120]. Since no information was published on the expected doublet rate,

¹²<https://support.missionbio.com/hc/en-us/articles/360045899834-Installation-instructions-for-tapestriR>

we performed a grid search to obtain the maximum likelihood estimate of the prior doublet probability δ . Fig. B.13 shows that the identified doublets have distinct VAF distributions compared to the remaining singlet droplets, both on the set of inference loci (that met the heterozygosity filtering criterion) as well as an orthogonal set of holdout loci (that did not meet the filtering criterion). Table B.2 shows the statistics and results generated by applying doubletD on data from all the patients in the dataset. Section 3.3 shows detailed analysis of Patient 1.

patient	n	m'	m	mean coverage	ADO β	doubletD
1	243	20	14	$14.3\times$	0.20	50 (0.21)
2	256	16	9	$15.6\times$	0.18	22 (0.09)
3	266	48	31	$9.1\times$	0.25	86 (0.32)
4	276	78	50	$7.1\times$	0.24	92 (0.33)
5	225	105	59	$7.0\times$	0.25	55 (0.24)
6	224	10	7	$16.2\times$	0.18	46 (0.21)

Table B.2: **Statistics and doublet detection results of an acute lymphoblastic leukemia cohort of six patients.** From left to right, the table shows for each patient the number n of droplets, the number m' of loci identified by [120], the number m of loci that meet our filtering criteria, the mean coverage of the samples, the ADO rate reported by [120] and the number (and fraction) of doublets identified by doubletD.

Phylogeny inference with PhISCS-B. PhISCS-B operates on a discretized input matrix that codes for the presence ('1') or absence ('0') of a mutation in a droplet as well as missing data ('?'). To discretize the input matrix, we used the binomial exact test to determine mutation status for each corresponding entry in the total and variant read count matrices $\mathbf{C}$ and $\mathbf{V}$ with a null error rate of 0.001 and a p-value of 10^{-6} . We provided PhISCS-B with the ADO rate $\beta = 0.2$ reported by [120] and a false positive rate of 0.001 that is typical for multiple displacement amplification (MDA) whole-genome amplification [94]. We imposed a maximum time limit of 3600 s.

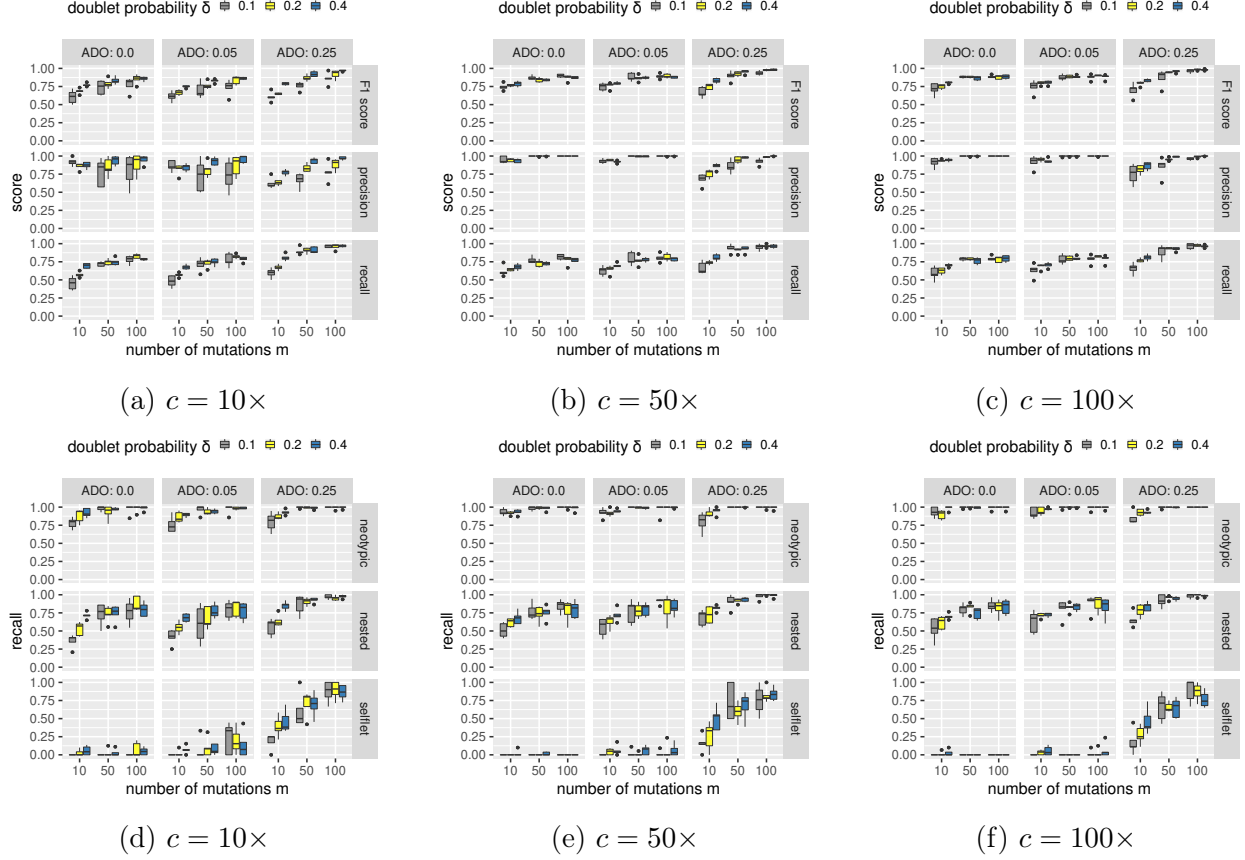


Figure B.2: F_1 score, recall and precision of doublet detection using DOUBLET-D for varying mean read depths c , ADO rates β and doublet probabilities δ . All simulations are run without copy number aberrations $\gamma = 0$ and precision parameter $s = 15$.

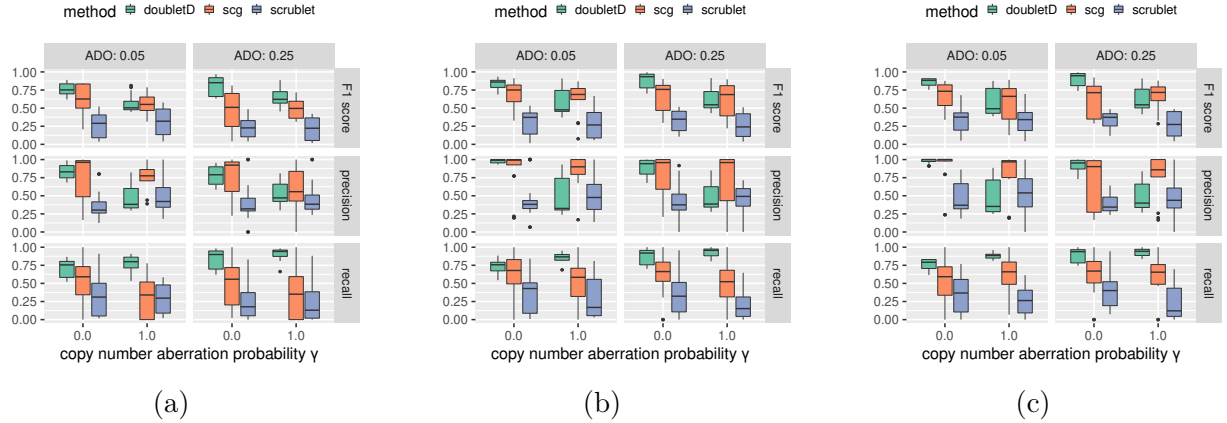


Figure B.3: F_1 score, recall and precision of doublet detection using DOUBLET-D on simulations without copy number losses and only gains. Results are shown for copy number aberrations probability $\gamma \in \{0, 1\}$ and ADO rates $\beta \in \{0.05, 0.25\}$. All simulations are run with doublet probability $\delta = 0.2$, mean read depth $c = 50\times$, number of mutations $m \in \{10, 50, 100\}$ and precision parameter $s = 15$.

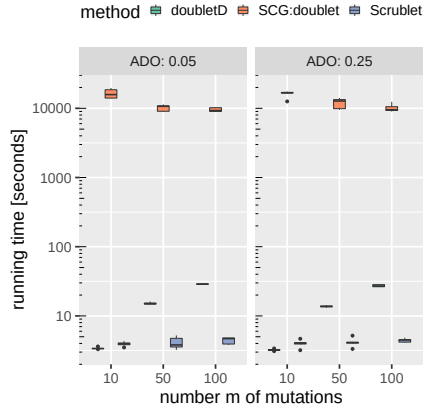


Figure B.4: Running time for doublet detection using DOUBLET D, SCG:doublet and SCRUBLET for simulations without CNAs ($\gamma = 0$) with varying number of mutations m . All simulations have doublet probability $\delta = 0.2$, mean read depth $c = 50\times$ and precision parameter $s = 15$.

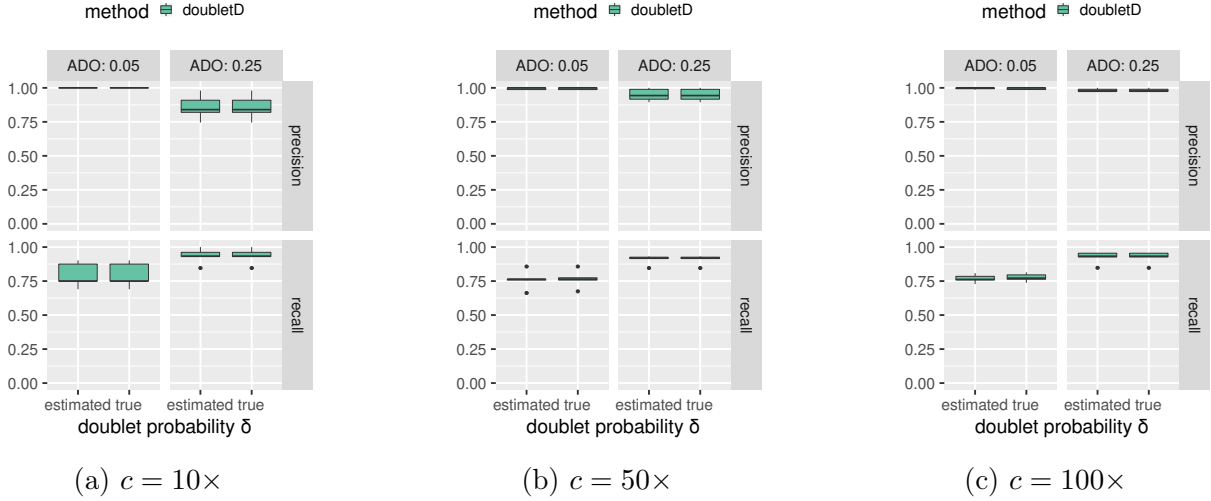


Figure B.5: Precision and recall for doublet detection using DOUBLET D with maximum likelihood estimate of the doublet probability δ and the true doublet probability used in the simulations for varying ADO rates β and mean read depth c . Results are shown for simulations with doublet probability $\delta = 0.2$, number of mutations $m \in \{10, 50, 100\}$ and precision parameter $s = 15$.

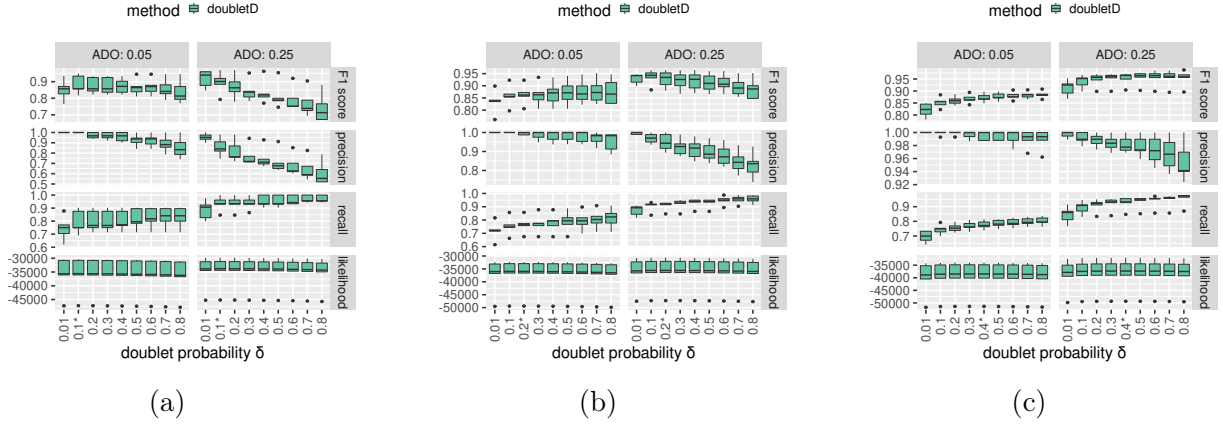


Figure B.6: F_1 score, precision, recall and posterior likelihood of doublet detection using DOUBLET D with varying input doublet probability δ . The simulations are run with doublet probability $\delta = \{0.1, 0.2, 0.4\}$, number of mutations $m = 50$, coverage $50\times$ and precision parameter $s = 15$. Copy number aberration probability γ was set to 0.

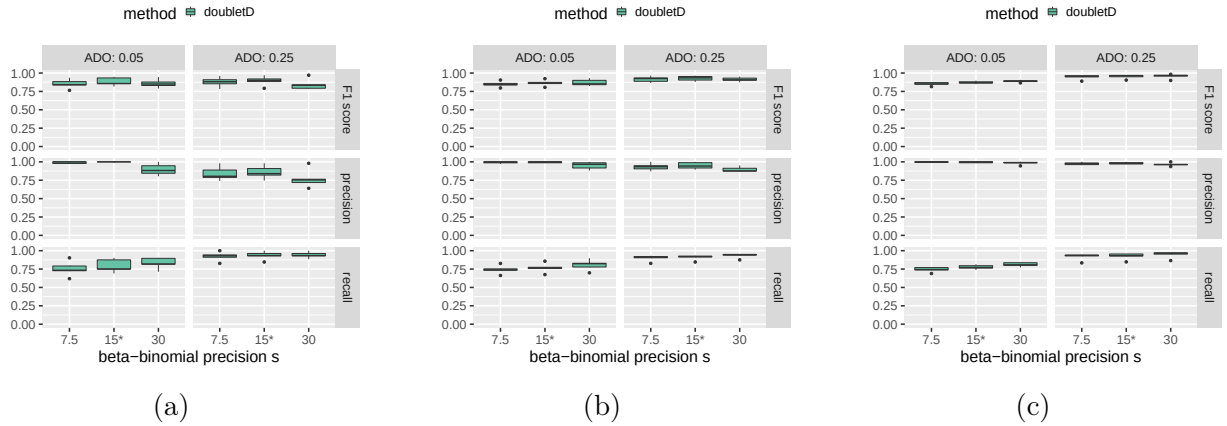


Figure B.7: F_1 score, precision, recall and posterior likelihood of doublet detection using DOUBLET D with varying input precision parameter s . The simulations are run with doublet probability $\delta = 0.2$, number of mutations $m \in \{10, 50, 100\}$ and precision parameter $s = 15$. Copy number aberration probability γ was set to 0.

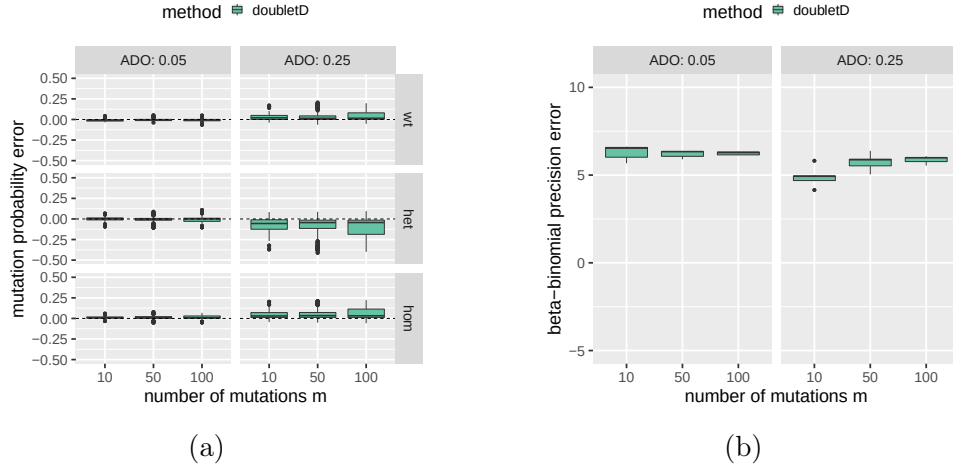


Figure B.8: Error in the estimation of (a) the mutation probabilities $\mu = \{\mu_{\text{wt}}, \mu_{\text{het}}, \mu_{\text{hom}}\}$ and (b) the precision parameter s from simulated data with varying number of mutations m and ADO rates β . All simulations are run with doublet probability $\delta = 0.2$, copy number aberration probability $\gamma = 0$ and precision parameter $s = 15$.

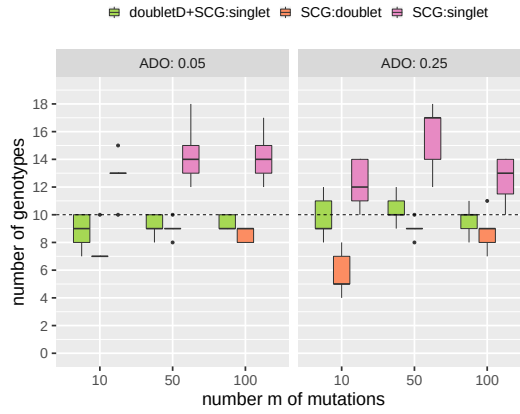


Figure B.9: Number of genotypes called by doubletD+SCG:singlet, SCG:doublet and SCG:singlet on simulations with varying number of mutations m and ADO rates β . All simulations have doublet probability $\delta = 0.2$, mean read depth $c = 50\times$, precision parameter $s = 15$ and copy number aberration $\gamma = 0$.

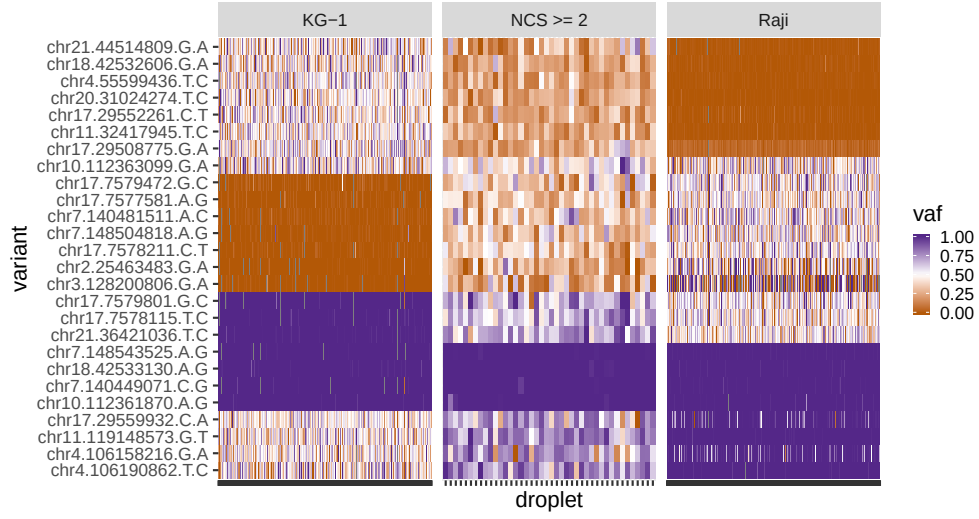


Figure B.10: Heatmap showing the observed variant allele frequency (VAF) of cell line droplets categorized by cell line or droplets with a neotypic doublet confidence score ($NCS \geq 2$).

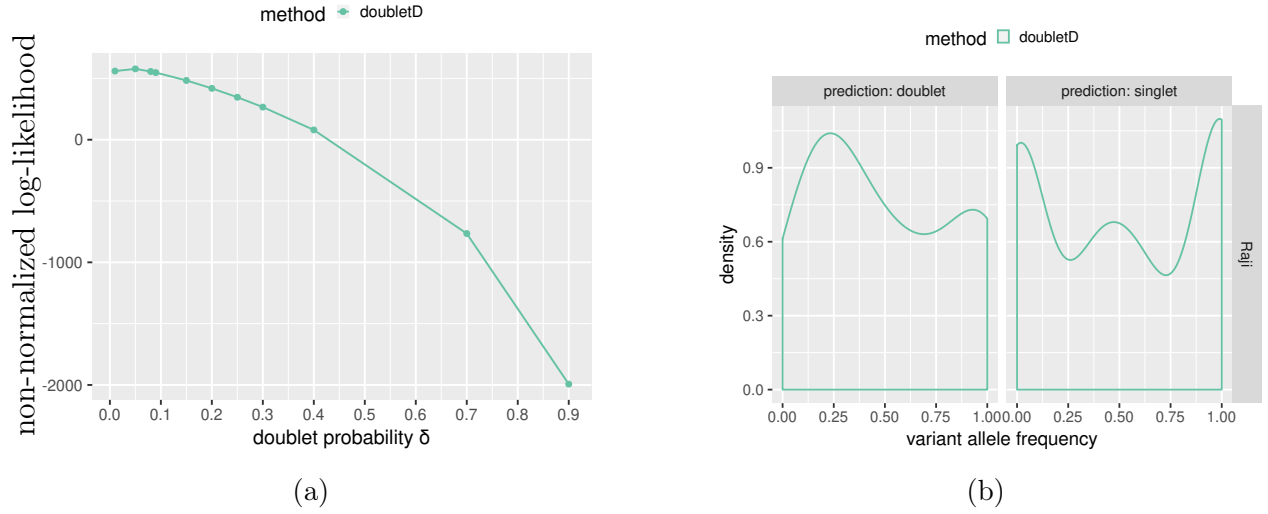


Figure B.11: DOUBLET-D results on cell line data. (a) The posterior likelihood (non-normalized) as a function of input doublet probability δ . Due to non-normalization, the log-likelihood is shifted by a constant explaining the positive values observed in the plot. (b) The observed VAF distributions for doubletD predicted doublets (1) and singlet droplets (8) for Raji droplets with neotypic doublet confidence score $NCS = 1$.

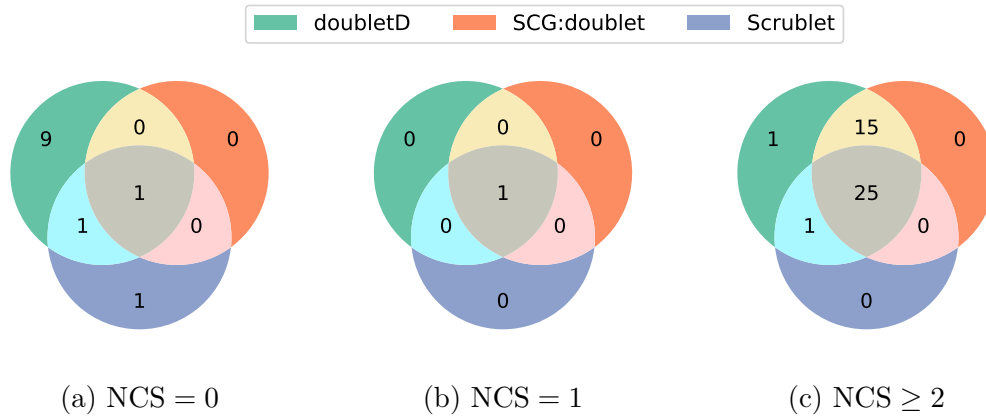
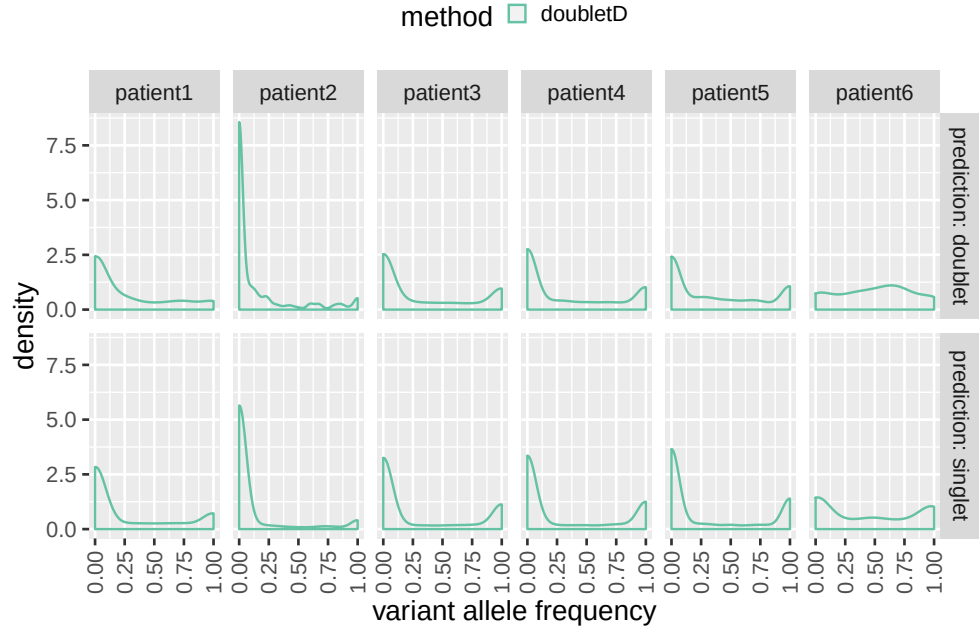
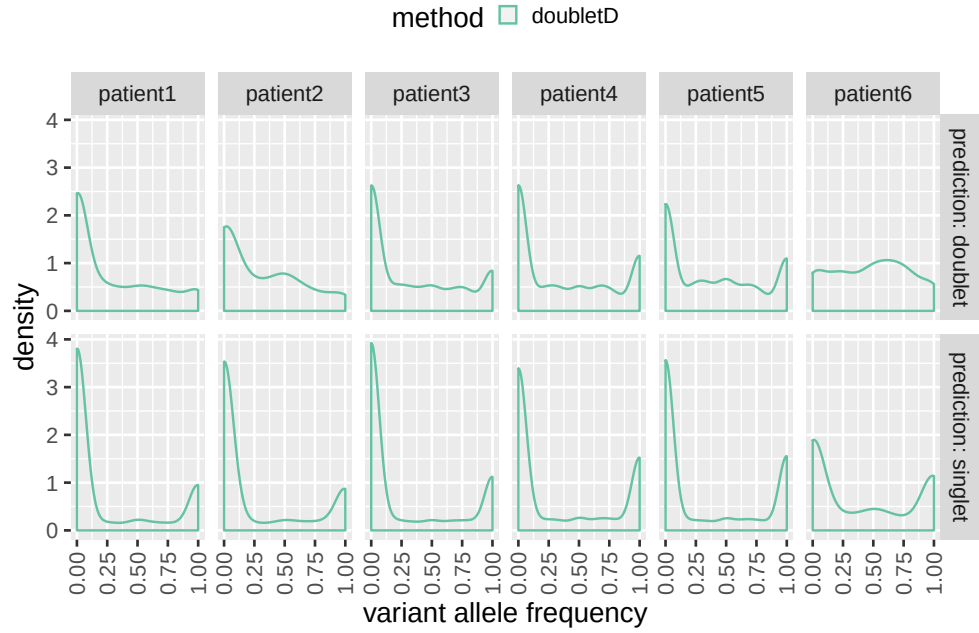


Figure B.12: Venn diagram of the droplets with NCS score of (a) 0 (b) 1 and (c) ≥ 2 that were predicted as doublets by the three methods, doubletD, SCG:doublet and SCRUBLET.



(a)



(b)

Figure B.13: Aggregated observed variant allele frequency distribution by patient and DOUBLET prediction for (a) across holdout loci (b) across inference loci.

APPENDIX C: SUPPLEMENTARY METHODS AND RESULTS FOR PHERTILIZER

C.1 SUPPLEMENTARY METHODS

C.1.1 Data processing

PHERTILIZER either takes as input binned read counts $\mathbf{R} \in \mathbb{N}^{n \times b}$ of n cells with read counts grouped into b bins or alternatively a binned read count embedding $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$ of read counts of n cells projected into $\ell \ll b$ dimensional space. If provided the former, PHERTILIZER will project the binned read counts $\mathbf{R}$ into $\ell = 2$ dimensions as follows. First, we compute normalized binned read counts $R' = [r'_{ip}] \in \mathbb{R}^{n \times b}$ as

$$r'_{ip} = \frac{r_{ip}}{\sum_{p'=1}^b r_{ip'}}. \quad (\text{C.1})$$

Next, we use UMAP [175] to project the normalized binned read counts $\mathbf{R}'$ to $\ell = 2$ dimensions with the following parameter settings:

`n_neighbors=40, spread=1, n_components=2, min_dist=0.`

This yields the binned read count embedding $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$ where $\ell = 2$.

C.1.2 Generative model

Our generative model (Fig. C.1) captures the evolution and the subsequent process of ultra-low coverage sequencing of n cells with m SNVs yielding our data $\mathbf{A}, \mathbf{D} \in \mathbb{N}^{n \times m}$ and $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$. For a number $k \in \mathbb{N}$ of clones, we generate a rooted tree T with k nodes with a uniform prior $P(T | k)$. We generate clonal genotypes $\mathbf{Y}$ for tree T under the infinite sites model with $P(\mathbf{Y} | T)$ uniformly distributed. We model the cell clustering $\phi : [k] \rightarrow 2^{[n]}$ with $P(\phi)$ distributed uniformly over all possible partitions of n cells into k clones/parts. The prior probability $P(\mathbf{D})$ of total read counts $\mathbf{D}$ is also uniform.

¹³This chapter contains identical material with an adapted introduction previously published in “Phertilizer: Growing a clonal tree from ultra-low coverage single-cell DNA sequencing of tumors” [69], which is an extended version of the paper from RECOMB-CCB 2022 by the same title. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with C. Zhang, I. Ochoa and M. El-Kebir. MEK and IO conceived the project; LLW and MEK established theory; LLW and CZ performed experiments; LLW, CZ, IO and MEK contributed to the writing of this chapter.

Latent variant allele frequency model. The latent variant allele frequency (VAF) x_{jq} denotes the fraction of chromosomal copies that harbor the SNV at locus q for clone j . Let $\bar{d}_{jq}$ be the number of chromosomal copies of the SNV locus q in clone j , and let $\bar{a}_{jq} \leq \bar{d}_{jq}$ be the number of chromosomal copies that harbor the SNV. Mathematically, we define the latent VAF as $x_{jq} = \bar{a}_{jq}/\bar{d}_{jq}$.

The latent VAF $x_{jq} \in \mathbb{Q}$ is conditioned on the latent clonal genotype $y_{jq} \in \{0, 1\}$. Given a pre-specified maximum copy number $c \in \mathbb{N} \setminus \{0\}$, we define the multiset S of latent VAFs associated with SNV state $y_{jq} = 1$ as

$$S = \{s/t \mid 1 \leq s \leq t \leq c\}. \quad (\text{C.2})$$

We denote with $|\{x \in S\}|$ the number of occurrences of latent VAF x in the multiset S . Then, we have

$$P(x_{jq} \mid y_{jq}, \theta) = \begin{cases} 1, & \text{if } y_{jq} = 0, x_{jq} = 0, \\ \frac{|\{x_{jq} \in S\}|}{|S|}, & \text{if } y_{jq} = 1, x_{jq} > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{C.3})$$

For example, if we have maximum copy number $c = 2$, the SNV is present (i.e., $y_{jq} = 1$), we have that $S = \{1/1, 1/2, 2/2\}$ and that $P(x_{jq} = 1/2 \mid y_{jq} = 1, \theta) = 1/3$.

Without assuming any prior information on the chromosomal copy numbers in each clone j , this model assumes that the latent VAF x_{jq} is directly proportional to the number of occurrences in multiset S . Using independence among clones $[k]$ and SNV loci $[m]$ when given clonal genotypes $\mathbf{Y}$, we have that

$$P(\mathbf{X} \mid \mathbf{Y}, \theta) = \prod_{j=1}^k \prod_{q=1}^m P(x_{jq} \mid y_{jq}, \theta). \quad (\text{C.4})$$

Variant read count model. Given latent VAFs $\mathbf{X}$, total read counts $\mathbf{D}$ and cell clustering ϕ , we model alternate read counts $\mathbf{A}$ as a binomial distribution accounting for sequencing errors. While previous models for medium-to-high coverage scDNA-seq data specifically account for errors introduced during PCR amplification [101, 102], the lack of preamplification in the latest generation of ultra-low scDNA-seq technology restricts errors to base calling errors that occur during sequencing [23, 24]. Without sequencing error, the success probability for a variant read would be latent VAF x_{jq} . Let α be the probability of misreading a single nucleotide during sequencing. For a successful trial, meaning we observe a variant allele, we

have two independent cases. In case I, we have no sequencing error ($1 - \alpha$ probability) and x_{jq} probability of reading a variant allele. In case I we have $1 - x_{jq}$ probability of sequencing a reference allele and $\alpha/3$ probability of misreading the reference allele for the variant allele.

Next we derive $P(\mathbf{A} \mid \mathbf{D}, \mathbf{X}, \phi, \theta)$.

$$P(a_{iq} \mid d_{iq}, \mathbf{x}_q, \phi, \theta) = \text{Binom} \left(a_{iq}; d_{iq}, x_{jq}(1 - \alpha) + \frac{(1 - x_{jq})\alpha}{3} \right), \quad (\text{C.5})$$

where j is the unique clone corresponding to cell i , i.e., $i \in \phi(j)$. Using an indicator variable, this is equivalent to

$$P(a_{iq} \mid d_{iq}, \mathbf{x}_q, \phi, \theta) = \prod_{j=1}^k P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta)^{\mathbf{1}\{i \in \phi(j)\}} \quad (\text{C.6})$$

$$= \prod_{j=1}^k \left[\text{Binom} \left(a_{iq}; d_{iq}, x_{jq}(1 - \alpha) + \frac{(1 - x_{jq})\alpha}{3} \right) \right]^{\mathbf{1}\{i \in \phi(j)\}}. \quad (\text{C.7})$$

Per the graphical model, we have independence among cells $[n]$ and SNV loci $[m]$ when conditioning on $\mathbf{D}, \mathbf{X}, \phi$ and θ , yielding

$$P(\mathbf{A} \mid \mathbf{D}, \mathbf{X}, \phi, \theta) = \prod_{j=1}^k \prod_{i=1}^n \prod_{q=1}^m P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta)^{\mathbf{1}\{i \in \phi(j)\}} \quad (\text{C.8})$$

$$= \prod_{j=1}^k \prod_{i=1}^n \prod_{q=1}^m \left[\text{Binom} \left(a_{iq}; d_{iq}, x_{jq}(1 - \alpha) + \frac{(1 - x_{jq})\alpha}{3} \right) \right]^{\mathbf{1}\{i \in \phi(j)\}}. \quad (\text{C.9})$$

Binned read count embedding model. The projection of binned reads counts $\mathbf{R}$ into a low dimension embedding $\tilde{\mathbf{R}}$ has two advantages. First, it allows us to identify sets of cells with similar copy number profiles, without the overhead of copy number calling. Second, distance functions behave counter-intuitively in high dimensions [203, 204] and therefore dimensionality reduction provides us with a useful proxy for the distance between the copy number profile of pairs of cells. We model the generation of the binned read counts embedding $\tilde{\mathbf{R}}$ using a Gaussian mixture model where $\tilde{r}_{ip} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \sim \mathcal{N}(\mu_p, \sigma_p^2)$. Latent variables $\boldsymbol{\mu} \in \mathbb{R}^{k \times \ell}$ and $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{k \times \ell}$ are independent for each clone $j \in [k]$ and dimension $p \in [\ell]$.

Following convention, let $z_i \in [k]$ be a random variable depicting the assignment of cell i to one of the k clones. Given a cell clustering ϕ , we have $P(z_i = j \mid \phi) = \mathbf{1}\{i \in \phi(j)\}$ meaning ϕ specifies a hard clustering of cells into k clones. Given cell clustering ϕ and latent variables $\boldsymbol{\mu}_j = [\mu_{jp}]$ and $\boldsymbol{\sigma}_j = [\sigma_{jp}]$ for each clone $j \in [k]$, the probability of binned read

count embedding for cell i and dimension p is thus

$$P(\tilde{r}_{ip}, | \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{j=1}^k P(z_i = j | \phi) \cdot \mathcal{N}(\mu_{jp}, \sigma_{jp}^2) \quad (\text{C.10})$$

$$= \prod_{j=1}^k \mathcal{N}(\mu_{jp}, \sigma_{jp}^2)^{\mathbf{1}_{\{i \in \phi(j)\}}}. \quad (\text{C.11})$$

Thus, using independence among cells and dimensions, we have the following probability

$$P(\tilde{\mathbf{R}} | \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \prod_{j=1}^k \prod_{i=1}^n \prod_{p=1}^{\ell} \mathcal{N}(\mu_{jp}, \sigma_{jp}^2)^{\mathbf{1}_{\{i \in \phi(j)\}}}. \quad (\text{C.12})$$

Posterior probability. As discussed in Main Text, given hyperparameters $\theta = (c, \alpha)$ we seek to solve

$$\operatorname{argmax}_{T, \mathbf{Y}, \phi} P(T, \mathbf{Y}, \phi | \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta). \quad (\text{C.13})$$

This is equivalent to marginalizing over latent variables $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$,

$$\operatorname{argmax}_{T, \mathbf{Y}, \phi} \int_{\boldsymbol{\mu}} \int_{\boldsymbol{\sigma}} P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} | \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) d\boldsymbol{\mu} d\boldsymbol{\sigma}. \quad (\text{C.14})$$

Computing the integrals over latent variables $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is challenging in practice, so we instead approximate the above expression by searching for the maximum *a posteriori* (MAP) estimates for latent variables $(\boldsymbol{\mu}, \boldsymbol{\sigma})$, giving the expression

$$\operatorname{argmax}_{T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}} P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} | \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta). \quad (\text{C.15})$$

We begin by applying Bayes' rule, obtaining

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, | \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) = \frac{P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} | \theta) P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} | T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta)}{P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} | \theta)}. \quad (\text{C.16})$$

We decompose $P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} | \theta)$ per the graphical model and

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, | \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) = \frac{P(T | \theta) P(\mathbf{Y} | T, \theta) P(\phi | \theta) P(\boldsymbol{\mu}, \boldsymbol{\sigma} | \theta) P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} | T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta)}{P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} | \theta)}. \quad (\text{C.17})$$

Using uniformity of $P(T | \theta)$, $P(\mathbf{Y} | T, \theta)$, $P(\phi | \theta)$, $P(\boldsymbol{\mu}, \boldsymbol{\sigma} | \theta)$ and dropping $P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} |$

θ) from the denominator, we obtain

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) \propto P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} \mid T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta). \quad (\text{C.18})$$

Using the law of conditional probability, we get

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) \propto P(\mathbf{A}, \tilde{\mathbf{R}} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) P(\mathbf{D}). \quad (\text{C.19})$$

We have that $P(\mathbf{D})$ is uniform, and after simplification, yields

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) \propto P(\mathbf{A}, \tilde{\mathbf{R}} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) \quad (\text{C.20})$$

$$= P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) P(\tilde{\mathbf{R}} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) \quad (\text{C.21})$$

$$= P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}). \quad (\text{C.22})$$

We now focus on $P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta)$, which equals

$$P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) = \prod_{j=1}^k \prod_{i=1}^n \prod_{q=1}^m P(a_{iq} \mid d_{iq}, y_{jq}, \phi, \theta)^{\mathbf{1}_{\{i \in \phi(j)\}}} \quad (\text{C.23})$$

$$= \prod_{j=1}^k \prod_{i=1}^n \prod_{q=1}^m \left[\sum_{x_{jq} \in S \cup \{0\}} P(a_{iq}, x_{jq} \mid d_{iq}, y_{jq}, \phi, \theta) \right]^{\mathbf{1}_{\{i \in \phi(j)\}}} \quad (\text{C.24})$$

$$= \prod_{j=1}^k \prod_{i=1}^n \prod_{q=1}^m \left[\sum_{x_{jq} \in S \cup \{0\}} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta) P(x_{jq} \mid y_{jq}, \theta) \right]^{\mathbf{1}_{\{i \in \phi(j)\}}} . \quad (\text{C.25})$$

Bringing everything together, we get

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) \quad (\text{C.26})$$

$$\propto P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) \quad (\text{C.27})$$

$$= \left(\prod_{j=1}^k \prod_{i=1}^n \prod_{q=1}^m \left[\sum_{x_{jq} \in S \cup \{0\}} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta) P(x_{jq} \mid y_{jq}, \theta) \right]^{\mathbf{1}_{\{i \in \phi(j)\}}} \right) \quad (\text{C.28})$$

$$\cdot \left(\prod_{j=1}^k \prod_{i=1}^n \prod_{p=1}^{\ell} \mathcal{N}(\mu_{jp}, \sigma_{jp}^2)^{\mathbf{1}_{\{i \in \phi(j)\}}} \right).$$

We now take the logarithm, yielding

$$P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) \quad (\text{C.29})$$

$$\propto \log P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) \quad (\text{C.30})$$

$$= \log P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) + \log P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) \quad (\text{C.31})$$

$$= \sum_{j=1}^k \sum_{i=1}^n \sum_{q=1}^m \mathbf{1}\{i \in \phi(j)\} \log \sum_{x_{jq} \in S \cup \{0\}} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta) P(x_{jq} \mid y_{jq}, \theta) \quad (\text{C.32})$$

$$+ \sum_{j=1}^k \sum_{i=1}^n \sum_{p=1}^{\ell} \mathbf{1}\{i \in \phi(j)\} \log \mathcal{N}(\mu_{jp}, \sigma_{jp}^2). \quad (\text{C.33})$$

Via the marginalization of latent VAFs $\mathbf{X}$ and taking the MAP estimates of latent variables $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, the above expression allows us to approximate the posterior probability $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$ of a given clonal tree T , clonal genotypes $\mathbf{Y}$ and cell clustering ϕ .

C.1.3 Detectability of a clone

We define *observations* $O(N, M)$ for a subset $N \subseteq [n]$ of cells and a subset $M \subseteq [m]$ of SNVs as

$$O(N, M) = \sum_{i \in N} \sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}. \quad (\text{C.34})$$

Thus, $O(i, M)$ is the number of SNV loci in M that have mapped reads in cell i . Similarly, $O(N, q)$ is the number of cells in N that have mapped reads for SNV locus q . Given a threshold parameter $t \in \mathbb{N}$, we use this function to define when a clone comprised of cells N and newly introduced SNVs M is detectable in the following way.

Definition C.1. Given parameter $t \in \mathbb{N}$, a clone comprised of cells N and newly introduced SNVs M is *detectable* provided (i) $\text{median}\{O(i, M) \mid i \in N\} \geq t$ and (ii) $\text{median}\{O(N, q) \mid q \in M\} \geq t$.

We refer to Fig. C.2 for an example.

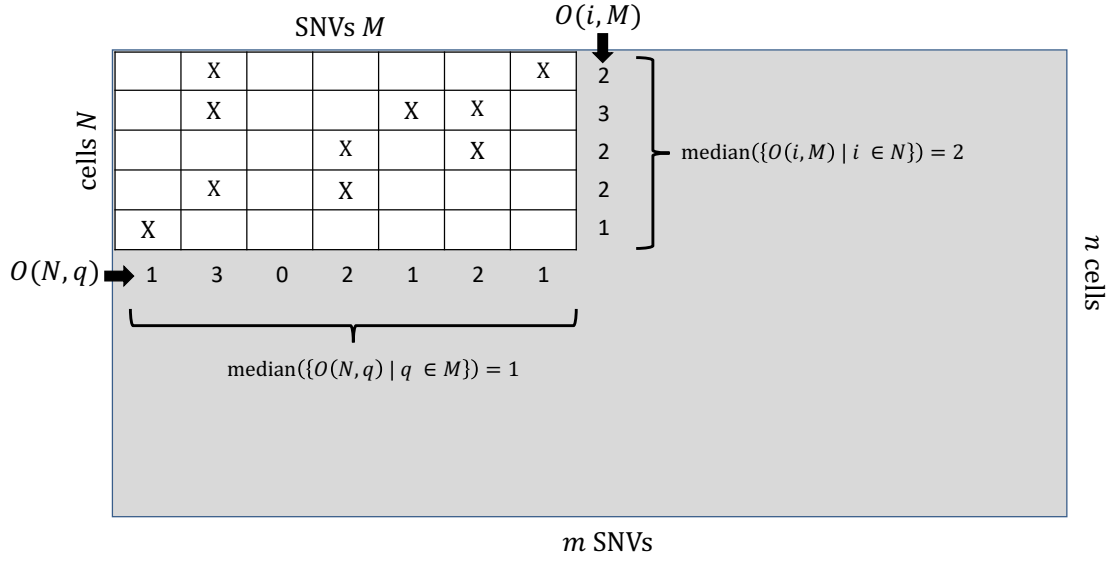


Figure C.2: **Visual depiction of the detectability of a clone with subset $N \subseteq [n]$ of cells and subset $M \subseteq [m]$ of SNVs.** Here, ‘X’ indicates the presence of a mapped read for SNV locus q in cell i . Observations $O(N, q)$ is the marginal sum of observations over cells N for each SNV $q \in M$. Similarly, $O(i, M)$ is the marginal sum of observations over SNVs M for each cell $i \in N$. For a clone to be detectable, the medians of these marginal sums must be at least a specified threshold $t \in \mathbb{N}$ (Definition C.1).

C.1.4 PHERTILIZER

PHERTILIZER maintains a set $\mathcal{T}$ of candidate trees throughout three phases: (i) initialization, (ii) growing, and (iii) ranking each tree in $\mathcal{T}$ by its posterior probability. First, in the initialization phase, the set $\mathcal{T}$ is initialized with a single tree containing only a root node v_1 . All n cells are assigned to the node’s cell cluster $\phi(v_1)$ and all genotypes are initialized to $y_{v_1, q} = 1$ for each SNV q .

Second, in the growing phase (Appendix C.1.4), PHERTILIZER recursively constructs the candidate set $\mathcal{T}$ of clonal trees and the respective clonal genotypes and cell clusterings by performing three different elementary tree operations (**Linear**, **Branching** and **Identity**) on each leaf node v_j of each candidate tree $T \in \mathcal{T}$ (Main Text Fig. 2, Fig. C.4 and Fig. C.5).

The input to each elementary tree operation is a clonal tree T with genotypes $\mathbf{Y}$, cell clustering ϕ and a leaf node v_j . The output is a new clonal tree T' with new genotypes $\mathbf{Y}'$ and cell clustering ϕ' . To define these elementary tree operations more formally, we introduce some notations. Given node v_j not equal to the root v_1 , the function $\text{par}(j)$ returns its parent node $v_{\text{par}(j)}$ in clonal tree T . Recall that $\mathbf{y}_j \in \{0, 1\}^m$ is the clonal genotype of clone j corresponding to node v_j in T . With slight overloading of notation, $\mathbf{y}_j$ also indicates the

set of SNVs present in clone j . We denote the set of SNVs introduced on the incoming edge $(v_{\text{par}(j)}, v_j)$ of node v_j by $\Delta(\mathbf{y}_j) = \mathbf{y}_j \setminus \mathbf{y}_{\text{par}(j)}$.

A **Linear** operation transforms node v_j of clonal tree T with cells $\phi(j)$ and mutations $\Delta(\mathbf{y}_j)$ into a clonal tree T' with a new node v_a and directed edge (v_j, v_a) , partitioning original cells $\phi(j)$ and mutations $\Delta(\mathbf{y}_j)$ among nodes v_j and v_a such that the resulting clonal genotypes $\mathbf{y}'_j, \mathbf{y}'_a$ adhere to the infinite sites model, i.e., $\mathbf{y}'_j \subseteq \mathbf{y}'_a$ and $\mathbf{y}'_j \cup \mathbf{y}'_a = \Delta(\mathbf{y}_j)$ (Main Text Fig. 2). Similarly, a **Branching** operation results in a clonal tree T' with two new nodes v_a, v_b that are children of v_j such that the original cells $\phi(j)$ are partitioned between child nodes v_a, v_b , the mutations $\Delta(\mathbf{y}_j)$ are partitioned between all three nodes v_j, v_a, v_b , and the genotypes $\mathbf{y}'_a, \mathbf{y}'_b, \mathbf{y}'_c$ also adhere to the infinite sites model, i.e., $\mathbf{y}'_j \subseteq \mathbf{y}'_a, \mathbf{y}'_j \subseteq \mathbf{y}'_b$, $\Delta(\mathbf{y}'_a) \cap \Delta(\mathbf{y}'_b) = \emptyset$ and $\mathbf{y}'_j \cup \mathbf{y}'_a \cup \mathbf{y}'_b = \Delta(\mathbf{y}_j)$. We note that this operation does not assign cells to v_j , thus modeling an extinct clone with SNVs that are common to its children but absent from its parent (Main Text Fig. 2). Lastly, an **Identity** operation on v_j does not alter the tree and prevents the application of other elementary operations on that node (Main Text Fig. 2).

Performing an elementary operation on a clonal tree T with leaf node v_j is equivalent to solving a constrained CTI problem. That is, we seek to infer a subtree rooted at node v_j with maximum posterior probability that is constrained to either be a two node linear subtree or a three node branching subtree. Both operations use coordinate descent to alternately find a partition of cells and SNVs that maximizes the posterior probability of the inferred clonal subtree. For ease of exposition, we describe the input to these operations as a set $N = \phi(j)$ of cells assigned to clone j in tree T and a set $M = \Delta(v_j)$ of SNVs gained on the incoming edge to node v_j . Next, we describe in more detail (i) the growing phase (Appendix C.1.4), (ii) the **Linear** elementary operation (Appendix C.1.4), (iii) the **Branching** elementary operation (Appendix C.1.4), (iv) the **Identity** (Appendix C.1.4) operation, (v) the running time of one elementary tree operation (Appendix C.1.4), (vi) a number of regularization steps to avoid overfitting the data (Appendix C.1.4) and (vii) a post-processing phase to improve clonal genotyping and cell clustering in the inferred clonal tree (Appendix C.1.4).

Growing phase. Elementary tree operations are the building of blocks of clonal trees. Starting from a single tree that contains a single clone containing all SNVs and cells, we consider all possible clonal trees that can be generated via a sequence of elementary tree operations. This is achieved via a growing phase, where all valid candidate clonal trees are generated, or grown, via recursive enumeration.

Specifically, we maintain a candidate set $\mathcal{T}$ of clonal trees. Each candidate tree T is marked as either visited or unvisited. Whenever a candidate tree is added to the set it is

initially marked as unvisited. After a candidate tree is explored, it is marked as visited. To visit a candidate tree $T \in \mathcal{T}$ (Fig. C.3a), we consider the subset $L'(T) \subseteq L(T)$ of leaf nodes such that each node $v \in L'(T)$ is not marked as terminal. A leaf node $v \in L(T)$ is marked as terminal if an **Identity** operation was previously applied to the leaf node v . We then apply valid **Linear**, **Branching** and **Identity** operations to each non-terminal leaf node $v \in L'(T)$, storing the extension of the tree (Fig. C.3b) to facilitate re-use in other operations. Lastly, we generate new candidate trees to add to the set $\mathcal{T}$ of candidates by applying at most $3^{|L'(T)|} - 1$ different sequences of operations, excluding invalid extensions of a leaf node and the simultaneous application of the identity operation to all leaf nodes (Fig. C.3c). Each of these newly generated clonal trees are added to the candidate set $\mathcal{T}$ and marked as unvisited. Candidate tree T is now marked as visited. The growing phase is complete when every candidate tree $T \in \mathcal{T}$ is marked visited.

Although the size of the candidate set grows exponentially, in practice we find that the size of the candidate set is manageable due to the sparsity of data and regularization (Appendix C.1.4). A node tends to become a terminal leaf node after only a few repeated applications of elementary tree operations as one or both of the resulting clones will be undetectable or result in a poor quality extension. This is attributed to either an insufficient number of cells and SNVs that results in the clone failing below the detectability threshold or because the sequence of prior of elementary operations is such a poor fit to the data that it fails to pass the quality of extension check. These regularization steps (Appendix C.1.4) are key to maintaining a manageable candidate set of clonal trees.

Once the growing phase is complete, we compute the posterior probability $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$ of each clonal tree T and the corresponding clonal genotypes $\mathbf{Y}$ and cell clustering ϕ in the candidate set $\mathcal{T}$ in order to rank and prioritize the candidate clonal trees. We return the clonal tree with maximum posterior probability.

Linear. For a given clonal tree T , we specify leaf node v_j as the node on which the **Linear** operation is to be performed. The **Linear** operation (Fig. C.4) extends tree T to create tree T' by replacing node v_j of tree T by a subtree rooted at node v_j with a single child v_a . Briefly, the **Linear** operation uses a coordinate descent approach to alternately optimize a two-part cell partition and a two-part SNV partition. To optimize the cell partition for a fixed SNV partition, we use the normalized cut algorithm [149]. To optimize the SNV partition for a fixed cell partition, we use our generative model to assign each SNV to the part that maximizes the posterior probability of the extended tree. Next, we describe these steps more formally.

Let $N = \phi(j) \subseteq [n]$ be the cells currently assigned to clone j associated with node v_j

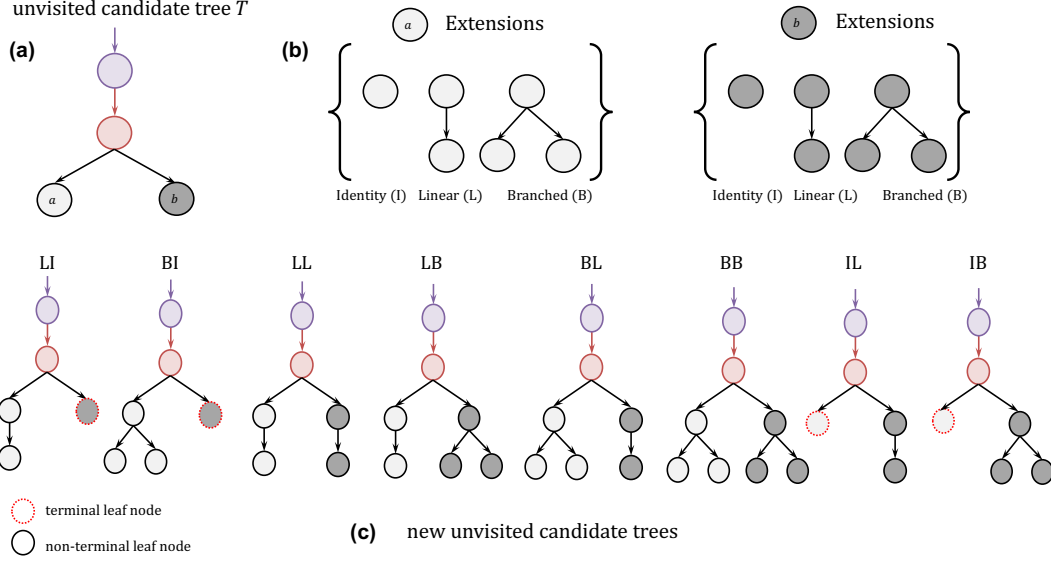


Figure C.3: **A graphical depiction of PHERTILIZER's recursive enumeration tree building process.** (a) An unvisited candidate clonal tree T in the candidate set $\mathcal{T}$ with two non-terminal leaf nodes $L'(T) = \{a, b\}$. (b) Valid tree extensions of non-terminal leaf nodes $L'(T)$ after application of each of the elementary tree operations. (c) Enumeration of new candidate clonal trees generated by applying each of the 8 valid sequences of elementary operations, i.e., $\{LI, BI, LL, LB, BL, BB, IL, IB\}$, to candidate tree T . These trees are then added to the candidate set $\mathcal{T}$ and marked unvisited. Candidate tree T is marked visited. This process is repeated until all candidate trees $T \in \mathcal{T}$ are marked visited.

and let $M = \Delta(\mathbf{y}_j) \subseteq [m]$ be the set of SNVs introduced on the incoming edge to node v_j of tree T . The output of a **Linear** operation is a two-part partition of the cells N into $\{N_{j'}, N_a\}$ and a two-part partition of the SNVs M into $\{M_{j'}, M_a\}$ (Fig. C.4). Parts $N_{j'}$ are the cells associated with node v_j in extended tree T' and N_a are the cells associated with the newly added child clone of node v_j in extended tree T' . The part $M_{j'}$ contains SNVs that are introduced on the incoming edge to node v_j in extended tree T' and the part M_a correspond to the SNVs newly introduced on the the incoming edge of the child node of v_j after the completion of the operation.

First, we initialize a **Linear** operation by partitioning uniformly at random the set M of SNVs into two parts $\{M_{j'}, M_a\}$. Given the SNV partition $\{M_{j'}, M_a\}$, we use the normalized cut algorithm [149] to find a two part cell partition $\{N_{j'}, N_a\}$ of cells N . The input to the normalized cut algorithm is a weighed adjacency matrix $\mathbf{W}$, where $w_{ii'}$ relates the similarity of two cells i and i' in the input set. We set the weight $w_{ii'}$ to incorporate both an SNV feature $\mathbf{f}$ and a CNA feature represented by the binned read count embedding $\tilde{\mathbf{R}}$.

The SNV feature f_i is defined as $f_i = \frac{\sum_{q \in M_a} \mathbf{1}\{a_{iq} > 0\}}{\sum_{q \in M_a} \mathbf{1}\{d_{iq} > 0\}}$. Intuitively, f_i measures for each cell

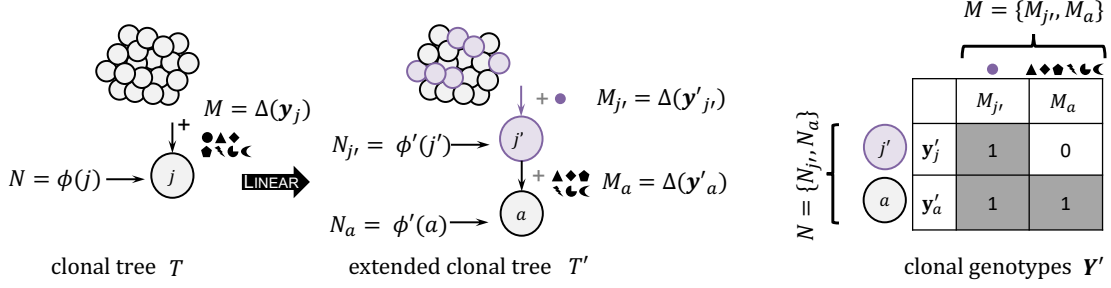


Figure C.4: **A graphical depiction of a Linear elementary tree operation.** The cells $N = \phi(j)$ and the SNVs $M = \Delta(\mathbf{y}_j)$ associated with clone j of tree T are partitioned into $N = \{N_{j'}, N_a\}$ and $M = \{M_{j'}, M_a\}$, respectively. These partitions are then used to update clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' associated with extended tree T' .

i the proportion of introduced SNV loci M_a with mapped reads for which we observe at least one variant read. In case the denominator is 0 for a cell i , we set $f_i := f_{i''}$ where i'' is the closest cell to i in $\tilde{\mathbf{R}}$ with a nonzero denominator (using Euclidean distance). Then the weight $w_{ii'}$ between cell i and i' is defined as

$$w_{ii'} = e^{\frac{-\|f_i - f_{i'}\|_2^2}{s_f}} \cdot \begin{cases} e^{\frac{-\|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2}{s_c}}, & \text{if } \|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2 < \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{C.35})$$

where τ is a user-specified parameter. We find that in practice, setting τ to the maximum $\|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2$ over all pairs of cells $i, i' \in N$ works well. However, setting τ more aggressively, i.e., lower, would put greater emphasis on the CNA features and result in a sparser input to the normalized cut algorithm. Following Shi and Malik [149], parameters s_f and s_c are set to 20% of the total range of the $\mathbf{L}^2$ norm for SNV and CNA features, respectively. We use the `scikit-learn` [205] spectral clustering implementation as it solves the normalized cut problem when the number of clusters is 2. We choose the `cluster-qr` [206] strategy to partition cells in the spectral embedding space.

Then, given the updated cell partition $\{N_{j'}, N_a\}$, we optimize the SNV partition $\{M_{j'}, M_a\}$. Let $\mathbf{a}'_q$ and $\mathbf{d}'_q$ be the column vector of variant and total read counts, respectively, corresponding to cells $N_{j'}$. For each SNV $q \in M$, we compute the log-likelihood $\log P(\mathbf{a}'_q \mid \mathbf{d}'_q, N_{j'}, q \in M', \theta)$ where $M' \in \{M_{j'}, M_a\}$. These probabilities are computed as follows.

$$\log P(\mathbf{a}'_q \mid \mathbf{d}'_q, N_{j'}, q \in M_{j'}, \theta) = \sum_{i \in N_{j'}} \log P(a'_{iq} \mid d'_{iq}, y_{j'q} = 1, \theta), \quad (\text{C.36})$$

$$\log P(\mathbf{a}'_q \mid \mathbf{d}'_q, N_{j'}, q \in M_a, \theta) = \sum_{i \in N_{j'}} \log P(a'_{iq} \mid d'_{iq}, y_{j'q} = 0, \theta). \quad (\text{C.37})$$

In other words, we compare the probabilities of assigning SNV q to either $M_{j'}$ or M_a . We assign q to the part $M' \in \{M_{j'}, M_a\}$ with maximum log-likelihood. We then iterate until either the cell clusters are not altered from the previous iteration or a user-specified maximum number of iterations is reached (default 50 iterations).

Due to the random initialization of the SNV partition, we repeat this process for a number of restarts. After a restart terminates, we compute a normalized log likelihood term for only the clonal genotypes that will not change even if the leaf nodes of the output tree T' are later extended. Among all restarts, the **Linear** operation stores the extended tree T' , clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' with maximum normalized log likelihood. For our simulation study, we found that 16 restarts led to stable results. For experimental data, we conservatively used 25 restarts.

Next, we recurse on an input with cells $N := N_{j'}$ and SNVs $M := M_{j'}$. The recursion terminates when either of the newly generated clones fails a regularization check, i.e., a detectability check or quality assessment of an extension. See Appendix C.1.4 for details on these criteria. The recursion step is necessary because the normalized cut algorithm is a recursive partitioning algorithm. Unlike **Branching** (Appendix C.1.4), where all the partitioned cells are associated with leaf nodes in the extended tree nodes, the **Linear** updates the cell cluster of an internal node in the extended tree T' . Since these cells will never be taken as input in any further elementary tree operation, recursions helps to refine the clones in the extended tree T' as much as possible. After the recursion terminates, the **Linear** operation returns the extended tree T' , clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' with maximum normalized log-likelihood.

Branching. For a given clonal tree T , we specify leaf node v_j as the node on which the **Branching** operation is to be performed. The **Branching** operation (Fig. C.5) extends tree T to create tree T' by replacing node v_j of tree T by a binary subtree rooted at node v_j with two children v_a and v_b . Briefly, the **Branching** operation uses a coordinate descent approach to alternately optimize a two-part cell partition and a three-part SNV partition. To optimize the cell partition for a fixed SNV partition, we use the normalized cut algorithm [149].

To optimize the SNV partition for a fixed cell partition, we use our generative model to

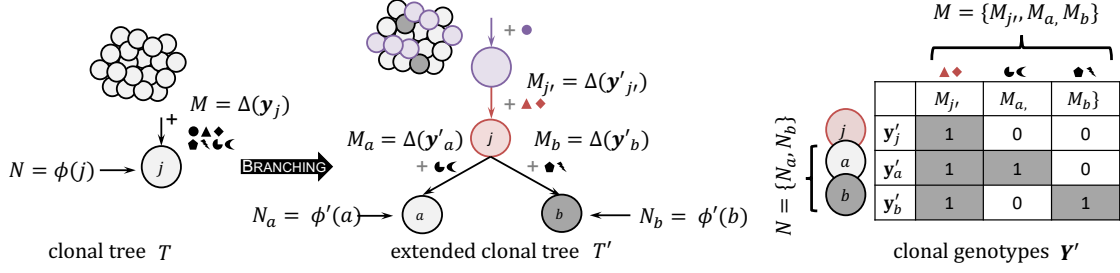


Figure C.5: **A graphical depiction of a Branching elementary tree operation.** The cells $N = \phi(j)$ and the SNVs $M = \Delta(\mathbf{y}_j)$ associated with clone j of tree T are partitioned into $N = \{N_a, N_b\}$ and $M = \{M_{j'}, M_a, M_b\}$, respectively. These partitions are then used to update clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' associated with extended tree T' .

assign each SNV to the part that maximizes the posterior probability of the extended tree. Next, we describe these steps more formally.

Let $N = \phi(j) \subseteq [n]$ be the cells currently assigned to clone j associated with node v_j and let $M = \Delta(\mathbf{y}_j) \subseteq [m]$ be the set of SNVs introduced on the incoming edge to node v_j for tree T . The output of a **Branching** operation is a two-part partition of the cells N into $\{N_a, N_b\}$ and a partition of SNVs M into three parts $\{M_{j'}, M_a, M_b\}$ (Fig. C.5). Parts N_a and N_b are the cells assigned to the two clones associated with the newly added two children of node v_j in extended tree T' . The part $M_{j'}$ contains SNVs that are introduced on the incoming edge to node v_j in extended tree T' . Parts M_a and M_b correspond to the SNVs newly introduced on the two incoming edges of the two children of node v_j after the completion of the operation.

First, we initialize a **Branching** operation by partitioning uniformly at random the set M of SNVs into three parts $\{M_{j'}, M_a, M_b\}$. Similarly to **Linear** (Appendix C.1.4), we use the SNV partition to find a two part cell partition $\{N_a, N_b\}$ of cells N .

The SNV feature is set as $\mathbf{f}_i = [f_{ia}, f_{ib}]^\top$, where

$$f_{ia} = \frac{\sum_{q \in M_a} \mathbf{1}\{a_{iq} > 0\}}{\sum_{q \in M_a} \mathbf{1}\{d_{iq} > 0\}}. \quad (\text{C.38})$$

In words, f_{ia} measures for each cell i the proportion of SNVs in set M_ℓ with mapped reads for which we observe at least one variant read. In case the denominator is 0 for a cell i , we set $f_{ia} := f_{i''a}$ where i'' is the closest cell to i in $\tilde{\mathbf{R}}$ with a nonzero denominator (using Euclidean distance). SNV feature f_{ib} is similarly defined for SNVs M_b .

$$w_{ii'} = e^{\frac{-\|\mathbf{f}_i - \mathbf{f}_{i'}\|_2^2}{s_f}} \cdot \begin{cases} e^{\frac{-\|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2}{s_c}}, & \text{if } \|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2 < \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{C.39})$$

where τ is a user-specified parameter. We find that in practice, setting τ to the maximum $\|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2$ over all pairs of cells $i, i' \in N$ works well. However, setting τ more aggressively, i.e., lower, would put greater emphasis on the CNA features and result in a sparser input to the normalized cut algorithm. Following Shi and Malik [149], parameters s_f and s_c are set to 20% of the total range of the $\mathbf{L}^2$ norm for SNV and CNA features, respectively. We use the `scikit-learn` [205] spectral clustering implementation as it solves the normalized cut problem when the number of clusters is 2. We choose the `cluster-qr` [206] strategy to partition cells in the spectral embedding space.

Then, given the updated cell partition $\{N_a, N_b\}$, we optimize the SNV partition $\{M_{j'}, M_a, M_b\}$. Let $\mathbf{a}'_q$ and $\mathbf{d}'_q$ be the column vector of variant and total read counts, respectively, corresponding to cells N . For each SNV $q \in M$, we compute the log-likelihood $\log P(\mathbf{a}'_q | \mathbf{d}'_q, N = \{N_a, N_b\}, q \in M', \theta)$ where $M' \in \{M_{j'}, M_a, M_b\}$. These probabilities are computed as follows.

$$\log P(\mathbf{a}'_q | \mathbf{d}'_q, N = \{N_a, N_b\}, q \in M_{j'}, \theta) = \sum_{i \in N} \log P(a'_{iq} | d'_{iq}, y_{j'q} = 1, \theta), \quad (\text{C.40})$$

$$\log P(\mathbf{a}'_q | \mathbf{d}'_q, N = \{N_a, N_b\}, q \in M_a, \theta) = \sum_{i \in N_a} \log P(a'_{iq} | d'_{iq}, y_{aq} = 1, \theta) \quad (\text{C.41})$$

$$+ \sum_{i \in N_b} \log P(a'_{iq} | d'_{iq}, y_{bq} = 0, \theta), \quad (\text{C.42})$$

$$\log P(\mathbf{a}'_q | \mathbf{d}'_q, N = \{N_a, N_b\}, q \in M_b, \theta) = \sum_{i \in N_b} \log P(a'_{iq} | d'_{iq}, y_{bq} = 1, \theta) \quad (\text{C.43})$$

$$+ \sum_{i \in N_a} \log P(a'_{iq} | d'_{iq}, y_{aq} = 0, \theta). \quad (\text{C.44})$$

In other words, we compare the probabilities of assigning SNV q to either $M_{j'}$, M_a or M_b . We assign q to the part $M' \in \{M_{j'}, M_a, M_b\}$ with maximum log-likelihood. We then iterate until either the cell clusters are not altered from the previous iteration or a user-specified maximum number of iterations is reached.

Due to the random initialization of the SNV partition, we repeat this process for a number of restarts. After a restart terminates, we compute a normalized log likelihood term for only

the clonal genotypes that will not change even if the leaf nodes of the output tree T' are later extended. Among all restarts, the **Branching** operation returns the extended tree T' , clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' with maximum normalized log likelihood. For our simulation study, we found that 16 restarts led to stable results. For experimental data, we conservatively used 25 restarts.

Identity. An **Identity** elementary operation is a trivial operation that simply returns an unmodified clonal tree and marks the inputted leaf node as terminal. For **PHERTILIZER**, application of the **Identity** operation is most useful during the growing, or recursive enumeration, stage (Appendix C.1.4) of the algorithm.

Running time of an elementary tree operation. Both **Linear** and **Branching** elementary tree operations require a cell partitioning step and SNV partitioning step. Thus, the running time of a single elementary tree operation is the combination of these two steps. The running time of cell clustering is $O(n^3)$ as it involves an eigenvalue decomposition [149]. In practice, we use the spectral clustering algorithm included in scikit-learn [205], which for two clusters reduces to the normalized cut algorithm. To partition SNVs, we pre-compute the log-likelihood $\log P(a_{iq} \mid d_{iq}, y = \ell, \theta)$ for each $\ell \in \{0, 1\}$, for each cell $i \in [n]$ and for each SNV $j \in [m]$ at the start of the algorithm. Given these log-likelihood values and a cell partition, we can partition SNVs in $O(nm)$. Thus, the total running time of a elementary tree operation is $O(n^3 + nm)$.

Regularization. To avoid overfitting the input data, we execute a number of regularization steps after performing an elementary tree operation on a node v_j of clonal tree T .

Detectability check. We assess whether the clones associated with new nodes v_j, v_a, v_b for **Branching** (Appendix C.1.4) and nodes v_j, v_a for **Linear** (Appendix C.1.4) are detectable (Appendix C.1.3). We set the default as $t = 4$.

Quality assessment of an extension. Even though we may have enough data observations to theoretically perform an elementary operation on a leaf node v_j in tree T , a high quality extension might not exist. We define the *cell mutational burden* (CMB) for a cell $i \in N \subseteq [n]$ and set $M \subseteq [m]$ of SNVs as

$$\text{CMB}(i, M) = \frac{\sum_{q \in M} \mathbf{1}\{a_{iq} > 0\}}{\sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}}. \quad (\text{C.45})$$

More details and intuition on the CMB is provided in Appendix C.2.2.

We define a quality check $\text{qc}(N, M)$ as $\text{median}\{\text{CMB}(i, M) \mid i \in [N]\}$. When attempting a **Linear** operation, which partitions $N = \phi(j)$ into parts $\{N_{j'}, N_a\}$ and SNVs $M = \Delta(\mathbf{y}_j)$ into parts $\{M_{j'}, M_a\}$ (Appendix C.1.4) we expect $\text{qc}(N, M_{j'})$ to be high, i.e., at least 0.15, since we expect all $M_{j'}$ SNVs to be present in the set N of cells. Conversely, we expect $\text{qc}(N_{j'}, M_a)$ to be low, i.e., at most 0.05, since we do not expect to see SNVs M_a in the set $N_{j'}$ of cells. For similar reasoning as above, we define the following quality checks for the **Branching** operation (Appendix C.1.4).

$$\text{qc}(N, M_{j'}) \geq 0.15, \quad (\text{C.46})$$

$$\text{qc}(N_a, M_b) \leq 0.05, \quad (\text{C.47})$$

$$\text{qc}(N_b, M_a) \leq 0.05. \quad (\text{C.48})$$

We use 0.05 and 0.15 for illustrative purposes and because we set these as the default values. However, the user may specify different values for these parameters to be less conservative.

Postprocessing. Recall that the input to an elementary operation is a node v_j with attached cells $\phi(j)$ and introduced SNVs $\Delta(\mathbf{y}_j)$. It may be the case that a cell i in $\phi(j)$ has zero variant reads for all SNVs in $\Delta(\mathbf{y}_j)$, i.e., $a_{iq} = 0$ for all SNVs $q \in \Delta(\mathbf{y}_j)$. Similarly, an SNV q in $\Delta(\mathbf{y}_j)$ may not have any cell in $\phi(j)$ with supporting variant reads, i.e., $a_{iq} = 0$ for all cells $i \in \phi(j)$. These SNVs and cells are uninformative during the elementary operations and their placement within the tree is best determined after the clonal tree has been identified. As such, prior to performing an elementary operation on node v_j , we remove such SNVs and cells, and add them to a set M of SNVs and a set N of cells, respectively. After PHERTILIZER returns the clonal tree with maximum posterior probability, we then perform an optional post-processing phase that contains two stages: (i) the placement of SNVs and cells that were removed from the tree during inference and (ii) the reassignment of SNVs that fit poorly in the tree.

For post-processing, we are given a clonal tree T , clonal genotypes $\mathbf{Y}$ and cell clustering ϕ , as well as the cells N and SNVs M as described above. In the first stage, we first fix the cell clustering ϕ . Recall that $\Delta(\mathbf{y}_j)$ is the set of SNVs introduced on the incoming edge to node v_j in tree T . Then, for each SNV $q \in M$ and each node $v_j \in V(T)$, we compute the

likelihood that SNV q is gained at node v_j in tree T as follows.

$$\log P(\mathbf{a}_q \mid \mathbf{d}_q, \phi, q \in \Delta(\mathbf{y}_j), T, \theta) = \sum_{v_{j'} \in \text{desc}(v_j) \cup \{v_j\}} \sum_{i \in \phi(j')} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 1, \theta) \quad (\text{C.49})$$

$$+ \sum_{v_{j'} \in \text{anc}(v_j)} \sum_{i \in \phi(j')} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 0, \theta), \quad (\text{C.50})$$

where $\text{desc}(v_j)$ is the set of descendant nodes and $\text{anc}(v_j)$ is the set of ancestor nodes of node v_j in tree T . We assign q to the incoming edge of node $v_j \in T$ with maximum log-likelihood, resulting in updated clonal genotypes $\mathbf{Y}'$. We then set clonal genotypes $\mathbf{Y} := \mathbf{Y}'$.

Next, we repeat the above process but fix the clonal genotypes $\mathbf{Y}$ and update the cell clustering ϕ . For each cell $i \in N$ and for each node $v_j \in V(T)$ such that $\phi(j) \neq \emptyset$, we compute the log-likelihood of cell i being assigned to clone j as follows.

$$\log P(\mathbf{a}_q \mid \mathbf{d}_q, \mathbf{Y}, i \in \phi(j), T, \theta) = \sum_{v_{j'} \in \text{anc}(v_j) \cup \{v_j\}} \sum_{q \in \Delta(\mathbf{y}_{j'})} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 1, \theta) \quad (\text{C.51})$$

$$+ \sum_{v_{j'} \in \text{desc}(v_j)} \sum_{q \in \Delta(\mathbf{y}_{j'})} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 0, \theta). \quad (\text{C.52})$$

We assign cell i to the cell cluster $j \in [k]$, with maximum log-likelihood resulting in updated cell clustering ϕ' . We then set cell clusterings $\phi := \phi'$.

In the second stage, given a tree T and a cell clustering ϕ , we attempt to improve clonal genotypes $\mathbf{Y}$ by moving SNVs that fit poorly with the given cell clustering ϕ and tree T . We iterate over the nodes of tree T using a postorder traversal. At each node v_j , we identify poorly fitting SNVs by computing a metric we refer to as the binary variant allele frequency (BVAf), which is as follows for a given SNV q and node v_j

$$\text{BVAf}(q, v_j) = \frac{\sum_{v_{j'} \in \text{desc}(v_j) \cup \{v_j\}} \sum_{i \in \phi(j')} \mathbf{1}\{a_{iq} > 0\}}{\sum_{v_{j'} \in \text{desc}(v_j) \cup \{v_j\}} \sum_{i \in \phi(j')} \mathbf{1}\{d_{iq} > 0\}}. \quad (\text{C.53})$$

In words, $\text{BVAf}(q, v_j)$ is the fraction of cells in clade v_j with mapped variant reads at SNV q . We compute $\text{BVAf}(q, v_j)$ for every SNV $q \in \Delta(\mathbf{y}_j)$. We identify a cutoff t' at the 10th percentile of this distribution. Then, any SNV q where $\text{BVAf}(q, v_j) \leq t'$ is removed from the tree and placed in a candidate set M of SNVs to be reassigned. Reassignment then proceeds as described above.

We repeat this second stage of post-processing for either a fixed number of iterations or until no SNVs are reassigned, resulting in updated clonal genotypes $\mathbf{Y}'$. After termination, we update clonal genotypes $\mathbf{Y} := \mathbf{Y}'$ and return clonal genotypes $\mathbf{Y}$ and cell clustering ϕ .

C.2 SUPPLEMENTARY RESULTS

C.2.1 Simulation study

BASELINE method. The BASELINE method seeks to infer clonal genotypes for cells utilizing copy-number information embedded in the binned read counts $\mathbf{R}$ via the following three steps. First, the method projects binned read counts $\mathbf{R} \in \mathbb{R}^{n \times b}$ into a low dimensional space $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$ (i.e., $\ell \ll b$) — see Appendix C.1.1. Second, it clusters cells into clones in the low-dimensional binned read count embedding $\tilde{\mathbf{R}}$. Third, each cluster of cells is treated as a pseudobulk sample and the cluster is genotyped. In our implementation, we used OPTICS [207] for the clustering step, since it is in the family of DBSCAN clustering methods that is a common choice used for the BASELINE in practice and the number of clusters does not need to be specified *a priori*. To genotype a cell cluster j , we set clonal genotype $y_{jq} = 1$ for each SNV $q \in [m]$ whenever the variant allele frequency, i.e., the total number of variant reads/total number of reads, was greater than 0.05 and $y_{jq} = 0$ otherwise. Fig. C.6 shows good concordance of clonal genotypes using this approach with that of the inferred clonal genotypes on a DLP+ dataset [24].

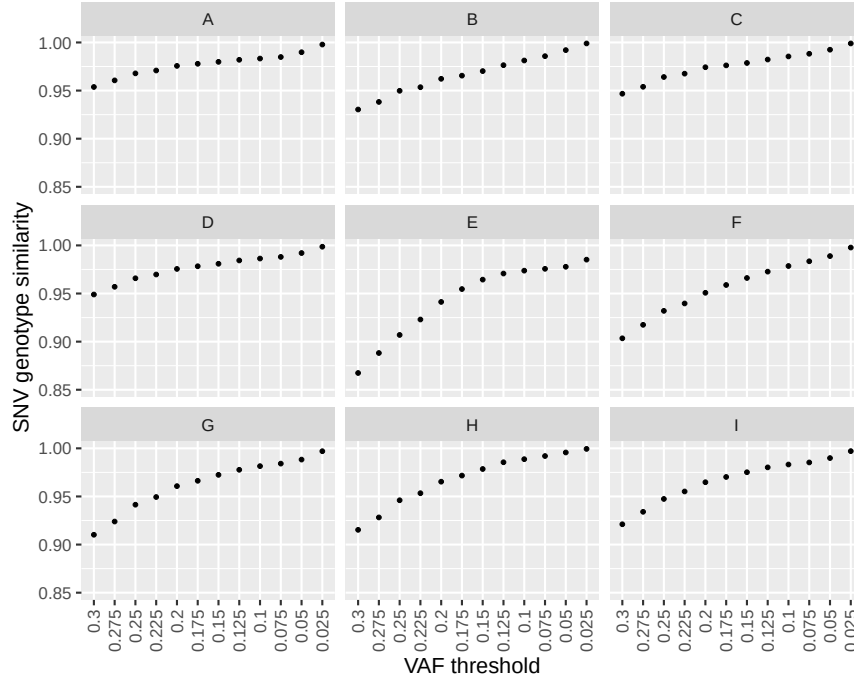


Figure C.6: Genotype similarity between inferred genotypes per cell cluster using the BASELINE method and varying the variant allele frequency (VAF) threshold and inferred clonal genotypes per cell cluster by Laks et al.[24]

Simulation setup. We generated scDNA-seq data from *in silico* heterogeneous tumors with both CNAs and SNVs. We generated a tree T^* with $k \in \{5, 9\}$ nodes. We varied the number $n \in \{1000, 2000\}$ of sequenced cells and the number $m \in \{5000, 10000, 15000\}$ of SNVs. We generated read counts following a beta-binomial distribution with coverage of $0.01\times$, $0.05\times$, and $0.1\times$. We introduced 3 chromosome-level and 8 chromosome-arm level CNAs, affecting a total of $\ell = 577$ bins each of length 5 MB spanning all autosomes. We sampled a cell clustering ϕ^* as well as SNV and CNA placement on edges from a symmetric Dirichlet distribution with a concentration parameter of 2.

Each combination of simulation parameters was replicated with ten different random number generator seeds, amounting to a total of 180 experiments. In addition, we generated a set of simulation experiments in the same manner as above with exception of fixing the copy number as heterozygous diploid at every locus in the genome. We also varied the SNV evolutionary model, using either a Dollo (SNV loss, [150]) or an infinite sites model (without SNV loss, [148]). We used a mean sequencing coverage of $0.01\times$ drawn from a Poisson distribution, to generate data matrices $\mathbf{D}, \mathbf{R}$ and sampled the variant count matrix $\mathbf{A}$ from a binomial distribution parameterized by the ground truth VAF.

Runtime parameters.

PHERTILIZER. We used default parameters specified below for all simulation instances.

```
phertilizer -f $input_dataframe_tsv$
    -bin_count_data $input_reads_per_bin_file$
    -radius 1 -a 0.001
    -min_obs 7
    -c 5
    -post_process
    -use_copy_kernel
    -d $seed$
    -j 50
    -s 16
    -low_cmb 0.05
    -high_cmb 0.15
    -nobs_per_cluster 3
```

SBMCLONE. We used the same seed for simulation as the seed for SBMClone.

```
python SBMClone/sbmclone.py -o ${out_dir} --seed $SEED \
    ${input_dataframe_tsv}
```

SCITE. We set the maximum MCMC chain length to be 900,000, repetitions to 3, false positive rate to be 0.02, false negative rate to be 0.01, and seed to 42.

```
scite -i ${input} -n ${mut_clusters} -m ${cell_clusters} \
    -r 3 -l 900000 \
    -fd 0.02 -ad 0.01 \
    -names ${names} \
    -seed 42 \
    -a -o ${outprefix}
```

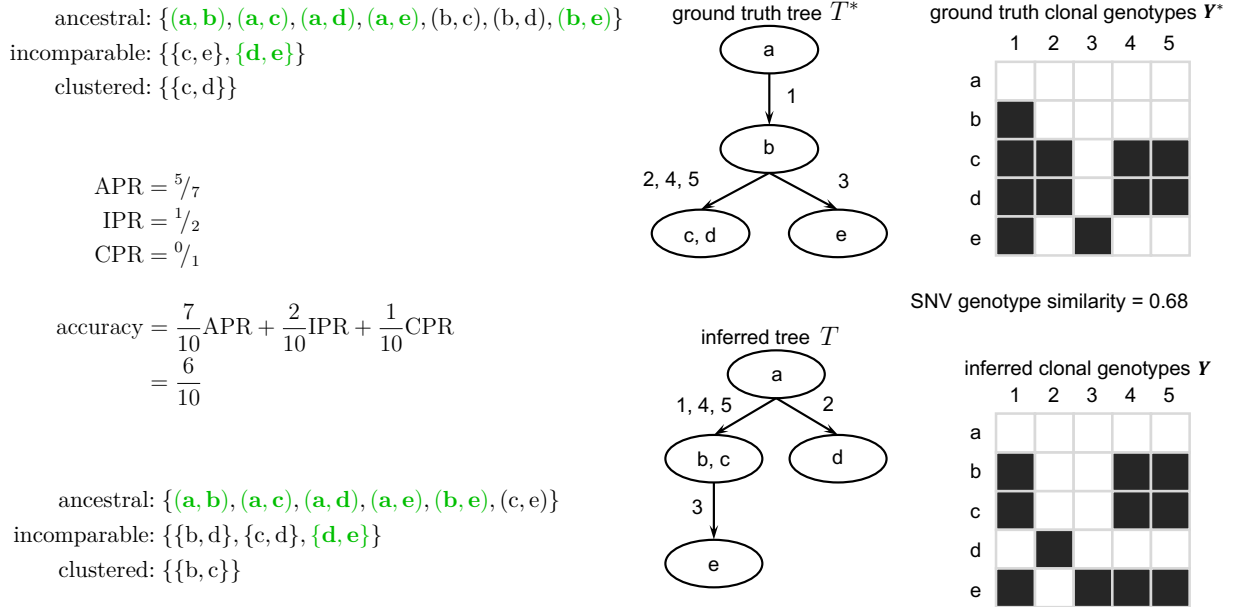


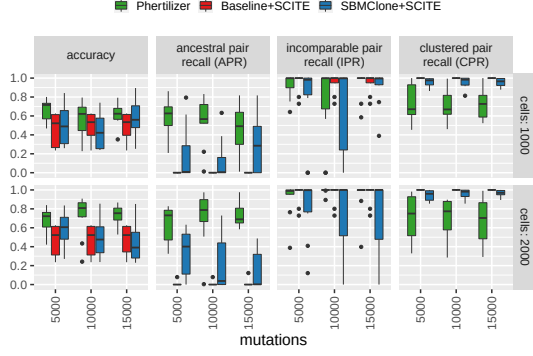
Figure C.7: **Example for ancestral pair recall (APR), clustered pair recall (CPR), incomparable pair recall (IPR), and accuracy for cells.** To the right of the trees, we show the ground truth and inferred genotypes $\mathbf{Y}^*$ and $\mathbf{Y}$ projected down to individual cells. Main Text Fig. 3 provides an example of APR, CPR, IPR and accuracy metrics for SNVs.

Performance metrics. We assessed the quality of an inferred solution $(T, \phi, \mathbf{Y})$ against a ground-truth tree T^* , cell clustering ϕ^* and clonal genotypes $\mathbf{Y}^*$ using ancestral pair recall (APR), incomparable pair recall (IPR), and clustered pair recall (CPR) metrics for cells and SNVs [127], as well as genotype similarity. In addition, we compute a single *accuracy* value

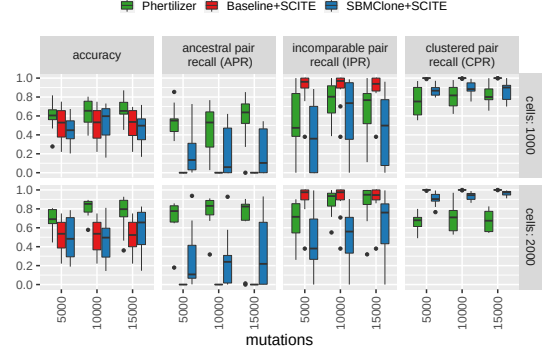
($\in [0, 1]$) composed of the weighted average of APR, IPR and CPR — where the weights are proportional to the number of pairs in each class. We note that an SNV and cell accuracy of 1 implies that the inferred solution perfectly matches ground truth.

APR, IPR, CPR. For any two SNVs $q \neq q'$ there are three possible placements in the tree: (i) *ancestral*: q is gained on a node that is distinct and ancestral to the node where q' is gained; (ii) *clustered*: q and q' are both gained on the same node; and (iii) *incomparable*: q and q' are gained on distinct nodes that occur on distinct branches of the tree. Similarly, two distinct cells $i \neq i'$ have the same three possible placements. The APR assesses the ratio of ancestral pairs from T^* recalled in T , whereas CPR and IPR do so for clustered and incomparable pairs, respectively. Thus, if $\text{APR} = \text{CPR} = \text{IPR} = 1$ for both cells and SNVs then the inferred solution $(T, \phi, \mathbf{Y})$ is identical to the ground truth $(T^*, \phi^*, \mathbf{Y}^*)$. Main Text Fig. 3 and Fig. C.7 provide a graphical depiction of these metrics.

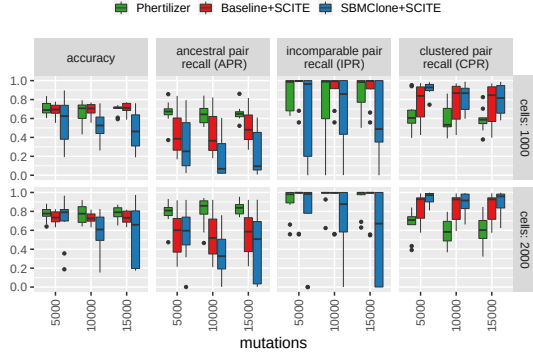
Genotype similarity. We define the *genotype similarity* as 1 minus the normalized Hamming distance between ground truth genotypes and inferred genotypes of cells. More formally, this is defined as $1/(nm) \sum_{i=1}^n \sum_{q=1}^m \mathbb{1}(y_{\bar{\phi}^*(i),j} = y_{\bar{\phi}(i),q}^*)$ where $\bar{\phi}(i)$ ($\bar{\phi}^*(i)$) is the unique clone j such that $i \in \phi(j)$ ($i \in \phi^*(j)$), given an inferred genotype $\hat{\mathbf{y}}_i \in \{0, 1\}^m$ and a ground truth genotype $\mathbf{y}_i^*$. Thus, a genotype similarity of 1 implies that every sequenced cell was correctly genotyped. Fig. C.7 above provides a visualization for the computation of this metric.



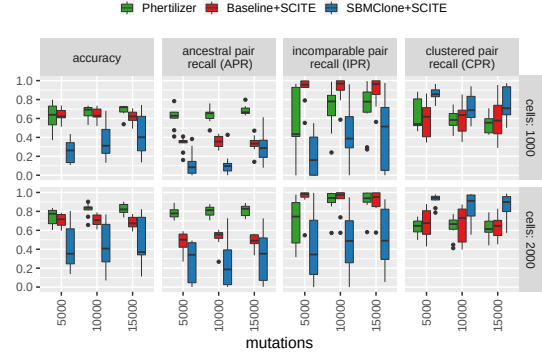
(a) cell placement metrics for $k = 5$ clones



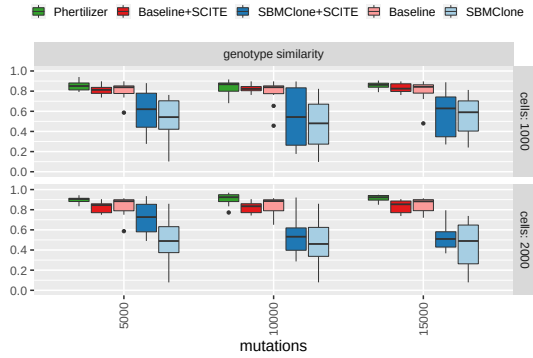
(b) cell placement metrics for $k = 9$ clones



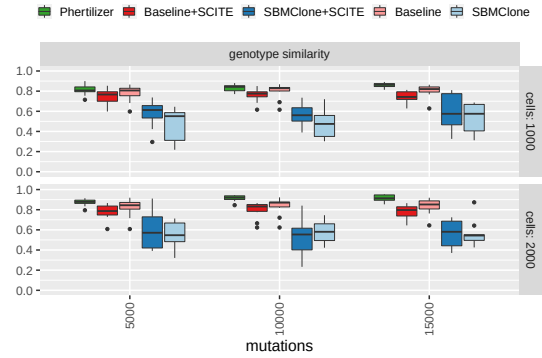
(c) SNV placement metrics for $k = 5$ clones



(d) SNV placement metrics for $k = 9$ clones

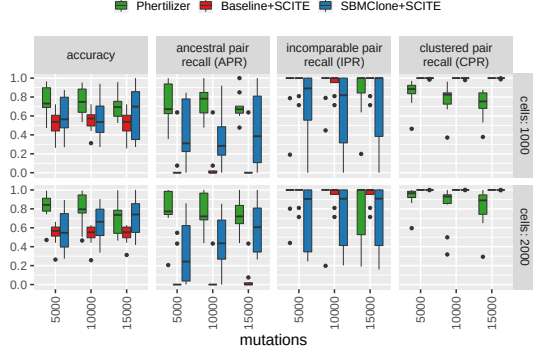


(e) genotype similarity for $k = 5$ clones

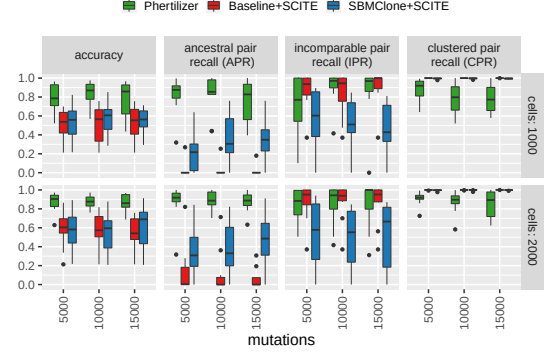


(f) genotype similarity for $k = 9$ clones

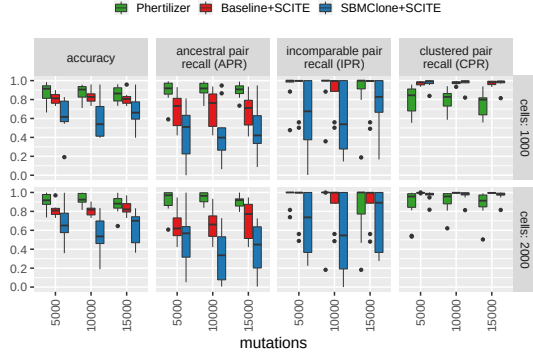
Figure C.8: **Simulation results for coverage $g = 0.01 \times$**



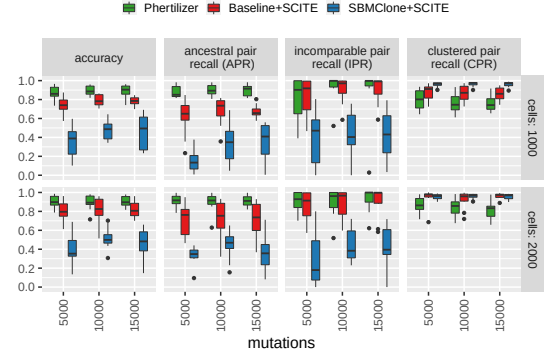
(a) cell placement metrics for $k = 5$ clones



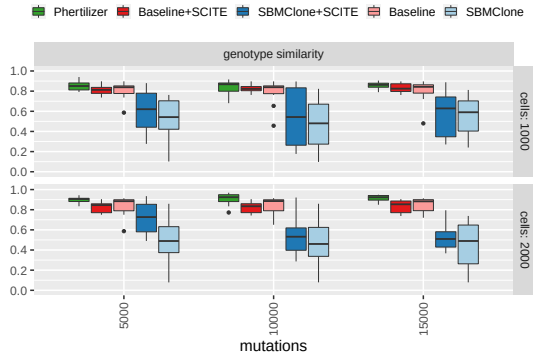
(b) cell placement metrics for $k = 9$ clones



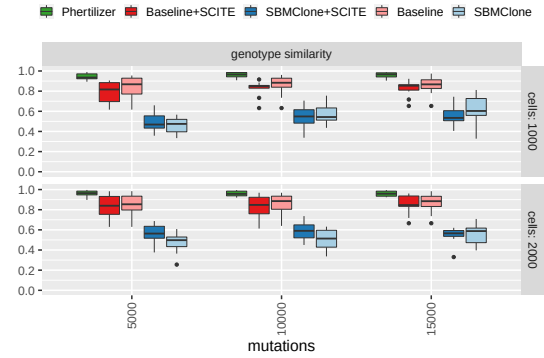
(c) SNV placement metrics for $k = 5$ clones



(d) SNV placement metrics for $k = 9$ clones

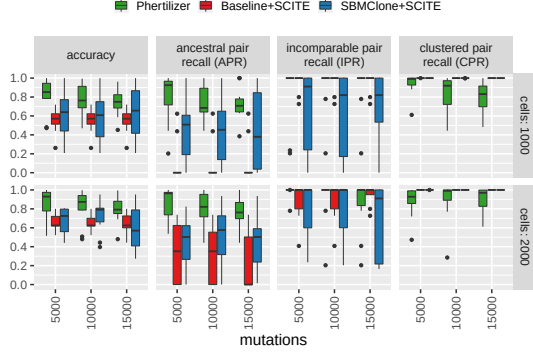


(e) genotype similarity for $k = 5$ clones

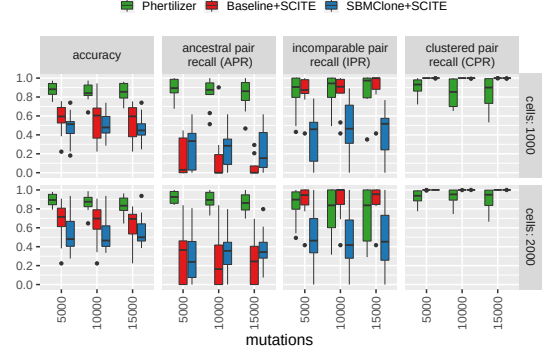


(f) genotype similarity for $k = 9$ clones

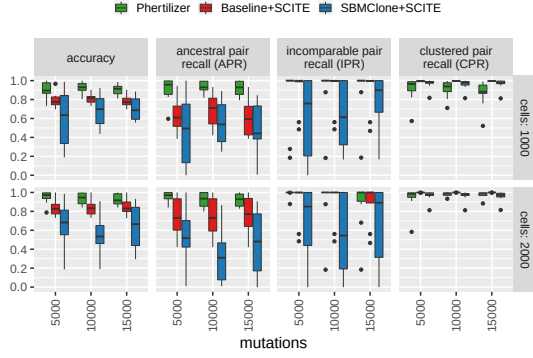
Figure C.9: Simulation results for coverage $g = 0.05\times$.



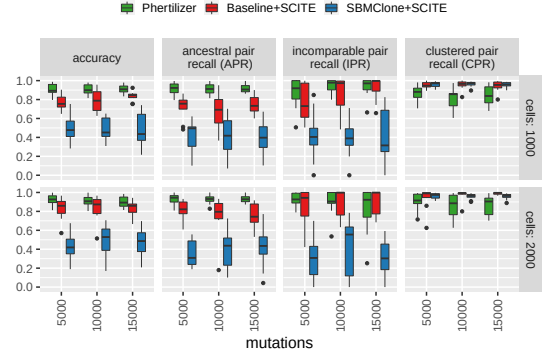
(a) cell placement metrics for $k = 5$ clones



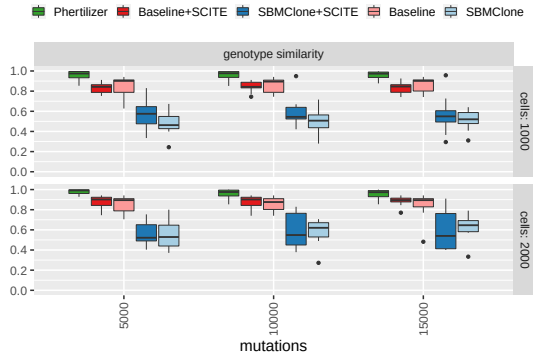
(b) cell placement metrics for $k = 9$ clones



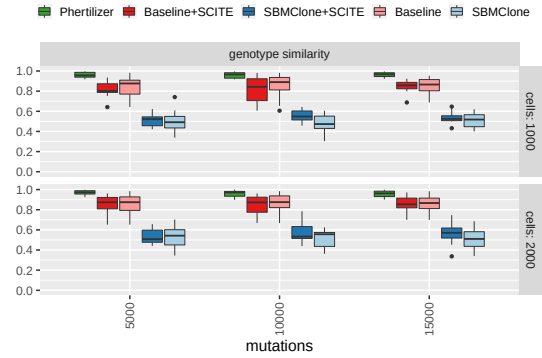
(c) SNV placement metrics for $k = 5$ clones



(d) SNV placement metrics for $k = 9$ clones



(e) genotype similarity for $k = 5$ clones



(f) genotype similarity for $k = 9$ clones

Figure C.10: Simulation results for coverage $g = 0.1 \times$.

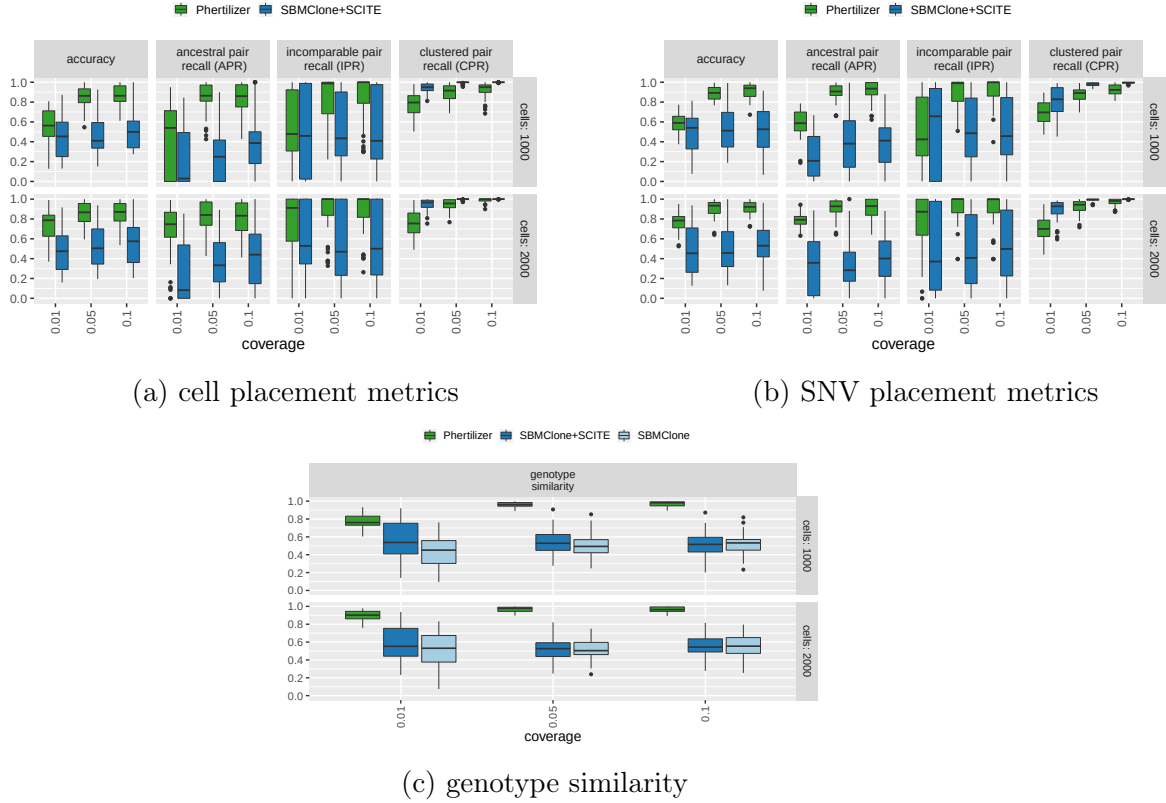


Figure C.11: **Heterozygous diploid simulation results aggregated over $k \in \{5, 9\}$ clones and $m \in \{5000, 10000, 15000\}$ SNVs.** BASELINE+SCITE was excluded from comparison because the lack of CNAs in the data resulted in only a single clone being identified in the read count embedding space for each instance.

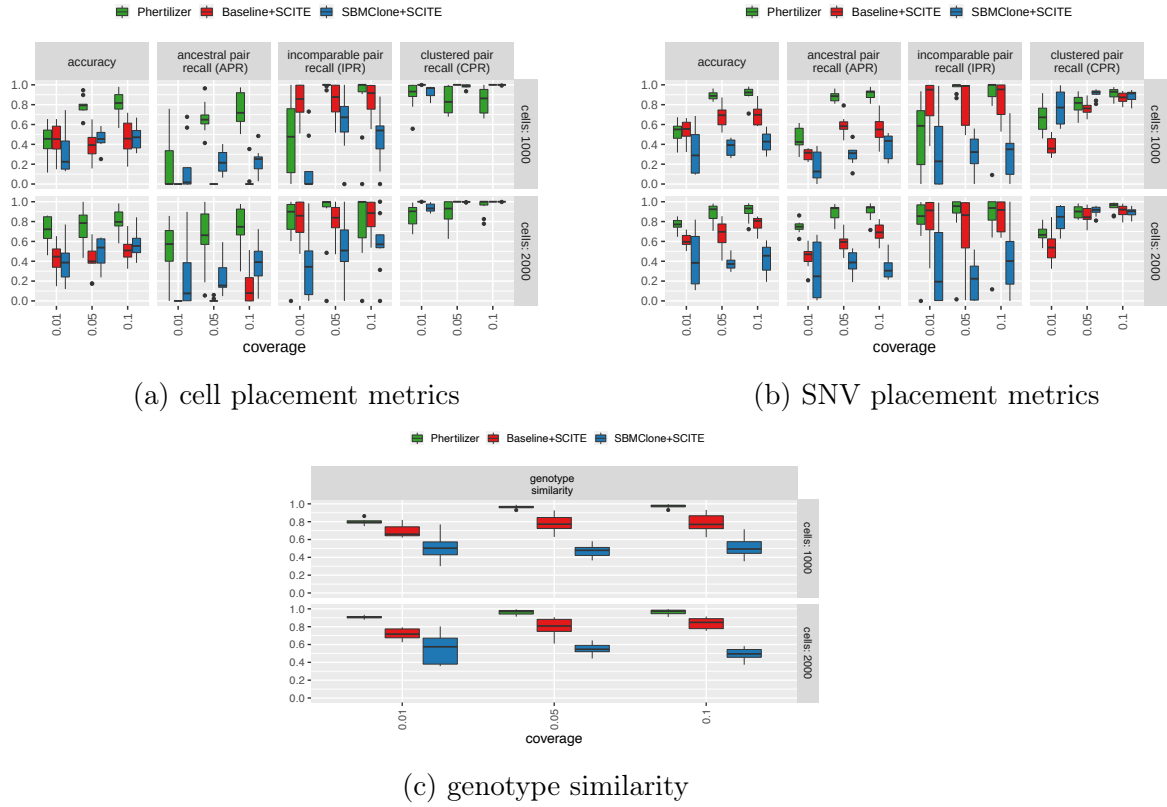


Figure C.12: Dollo evolutionary model simulation results for $k = 9$ clones and $m = 15000$ SNVs

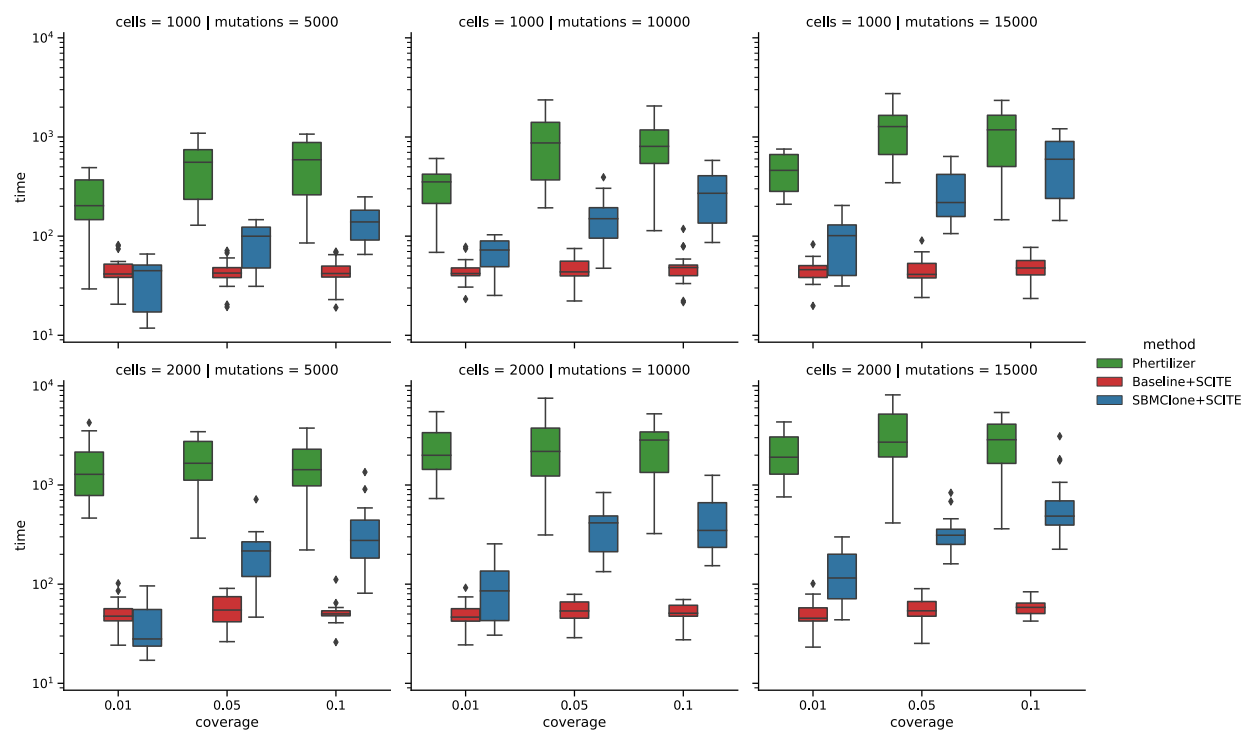


Figure C.13: **Running time (in seconds) on simulation data**

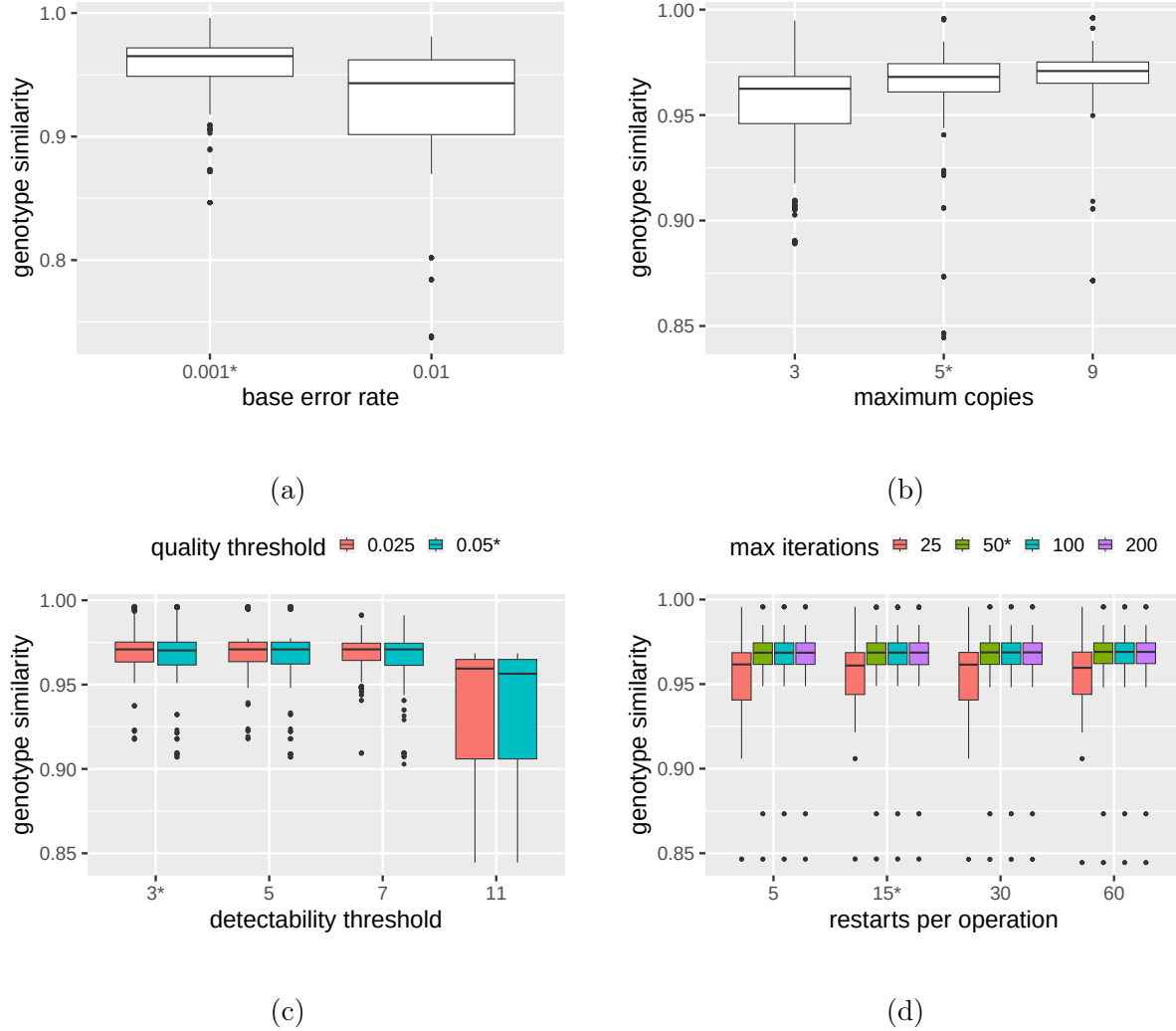


Figure C.14: **Hyperparameter impact on genotype similarity for $k = 9$ clones, $n = 2000$ cells, $m = 10000$ SNVs and coverage $g = 0.05\times$.** Hyperparameter choices shown with an asterisk (*) indicates the default setting using in simulation study. (a) Impact of hyperparameter base sequence error rate α . (b) Impact of hyperparameter maximum number of copies c with base sequencing error rate held constant at $\alpha = 0.001$. (c) Impact of hyperparameter detection threshold t and quality of extension (qc) threshold (Appendix C.1.4) with base sequencing error rate held constant at $\alpha = 0.001$. (d) Impact of hyperparameter choice on the number of restarts per operation and the maximum number of iterations per operations with base sequencing error rate held constant at $\alpha = 0.001$.

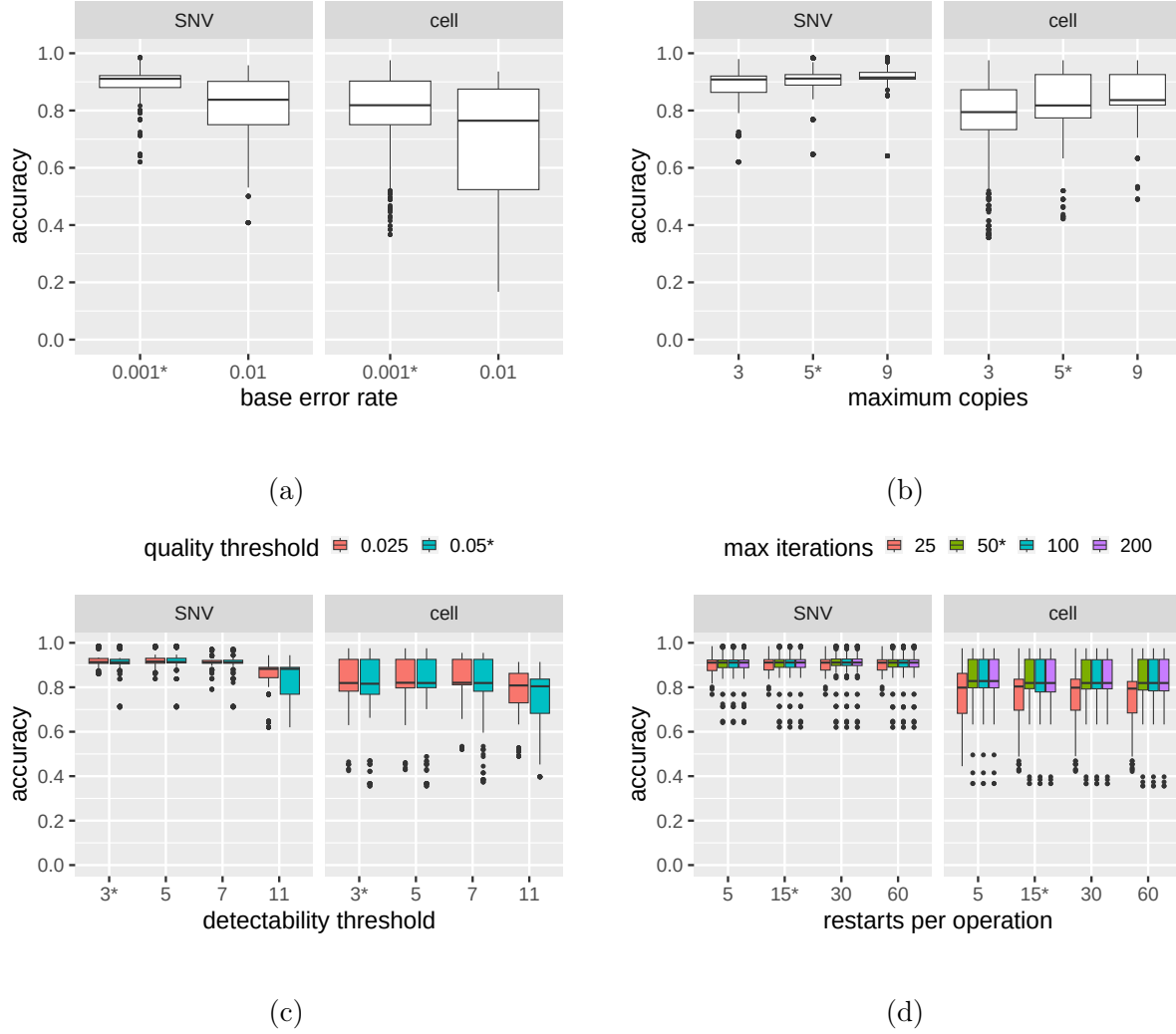


Figure C.15: **Hyperparameter impact on tree reconstruction accuracy for $k = 9$ clones, $n = 2000$ cells, $m = 10000$ SNVs and coverage $g = 0.05\times$.** Hyperparameter choices shown with an asterisk (*) indicates the default setting using in simulation study. (a) Impact of hyperparameter base sequence error rate α . (b) Impact of hyperparameter maximum number of copies c with base sequencing error rate held constant at $\alpha = 0.001$. (c) Impact of hyperparameter detection threshold t and quality of extension (qc) threshold (Appendix C.1.4) with base sequencing error rate held constant at $\alpha = 0.001$. (d) Impact of hyperparameter choice on the number of restarts per operation and the maximum number of iterations per operations with base sequencing error rate held constant at $\alpha = 0.001$.

C.2.2 Experimental data

Below we provide pertinent details and supplemental results for our analysis of high-grade serous ovarian cancer cells sequenced via DLP+ and triple negative breast cancer tumors sequenced via ACT.

PHERTILIZER runtime parameters for experimental data. We set hyperparameters $c = 5$ and $\alpha = 0.001$ for all experimental data analyses. We used 25 restarts and a maximum of 50 iterations for each elementary tree operation. Additionally, we performed a grid search over detectability threshold $t \in \{5, 6, 7\}$ (Appendix C.1.3) and the lower bound for the quality check $qc \in \{0.05, 0.075\}$ (Appendix C.1.4). The upper bound for the quality check qc was set to 0.15. After running PHERTILIZER on each of these 6 combinations of parameters, we selected the clonal tree with maximum posterior probability.

Placement of driver genes on inferred trees. We annotated inferred clonal trees with cancer-related genes listed in the Cancer Gene Census (CGC) [153] from COSMIC v97 and cBioPortal [151, 152]. In particular, we identified missense variants in cancer-related genes as well as genes with stop-gain variants. We first selected stop-gain variants. Then, we annotated variants with VEP [208], and looked for missense variants that are predicted to be damaging or deleterious by both SIFT [209] and PolyPhen [210]. Among these, we selected variants either present in the CGC, or in more than 0.4% of the total 76,639 samples from 10 pan-cancer studies included in cBioPortal. As a large number of SNVs are found in patient TN8 of the ACT dataset, we only labeled genes present in the CGC.

Cell mutational burden (CMB). To assess the quality of each inferred clade for real data, we developed a performance metric called *cell mutational burden* (CMB) defined as

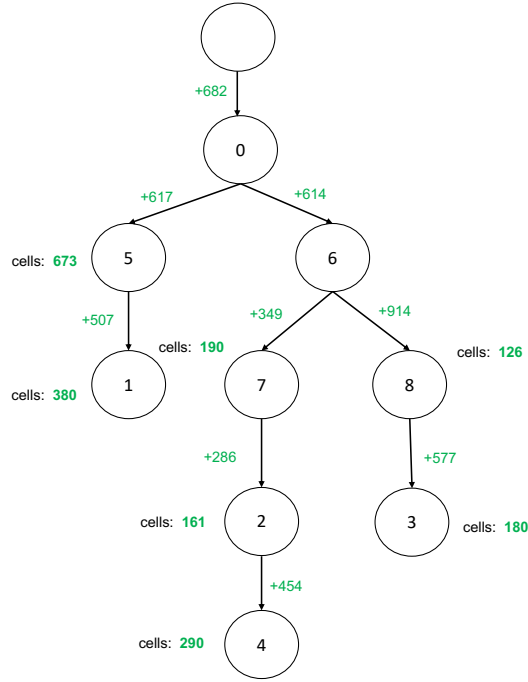
$$\text{CMB}(i, M) = \sum_{q \in M} \mathbf{1}\{a_{iq} > 0\} / \sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}. \quad (\text{C.54})$$

In words, $\text{CMB}(i, M)$ is the fraction of mapped SNV loci M with mapped variant reads in cell i .

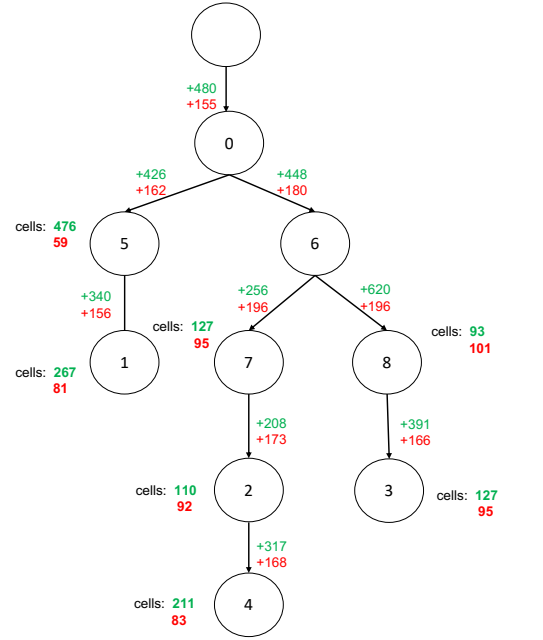
For a specified *clade* j or subtree rooted at node v_j , SNVs M_j are the SNVs gained at node v_j . CMB is designed to assess the goodness of fit of a proposed clonal tree without a known ground truth tree and succinctly captures the relationship between the inferred tree, clonal genotyping, and cell clustering. For a cell i placed within clade j , we expect $\text{CMB}(i, M_j)$ to be high, although the value will depend on copy number. By contrast, for cells placed

outside of clade j , we expect $\text{CMB}(i, M_j)$ to be low.

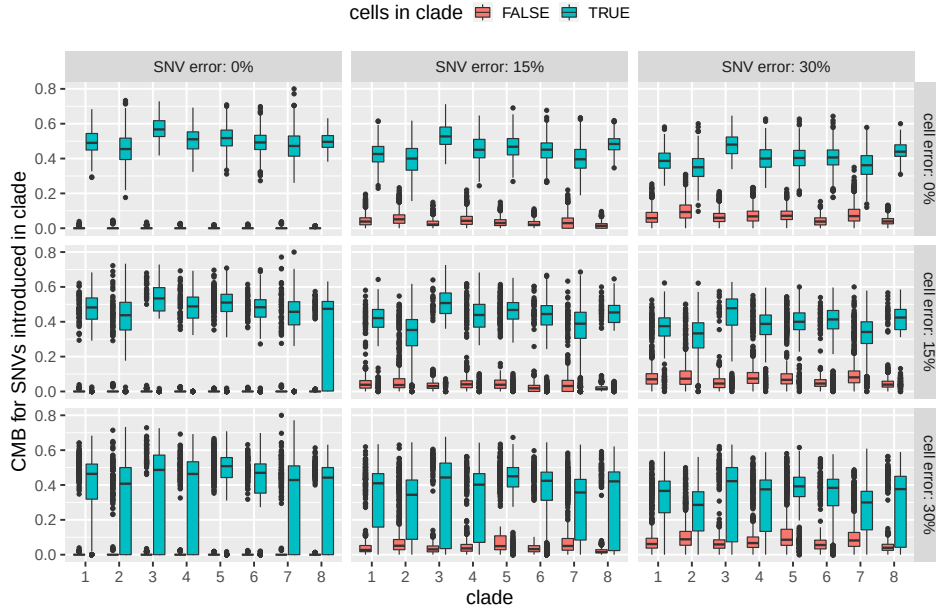
Because this is a newly proposed metric that has complex interactions between inference errors, we performed a sensitivity analysis on a simulated tree (Fig. C.16a) with varying rates in $\{0\%, 15\%, 30\%\}$ of cell and SNV placement errors (Fig. C.16). For reference, we also show cell and SNV placement on the simulated ground truth tree (Fig. C.16b) at the highest error rate for both (30%). This analysis showed that in the error free regime, CMB value for cells in clade ranges from 0.2 to 0.8 but the value will depend on copy number. We also found that as SNV error rates increase, the median CMB for cells outside of the clade increases with median values as high as 0.1 and an increased number of outliers with values as high as 0.7. Finally, as cell error rates increase the variability of CMB values for cells in clade increases. Although this analysis was only performed on a single simulation instance, it is helpful for interpreting this metric on experimental data where ground truth is unknown.



(a) Simulated tree with cell 0% and SNV 0% error rate.

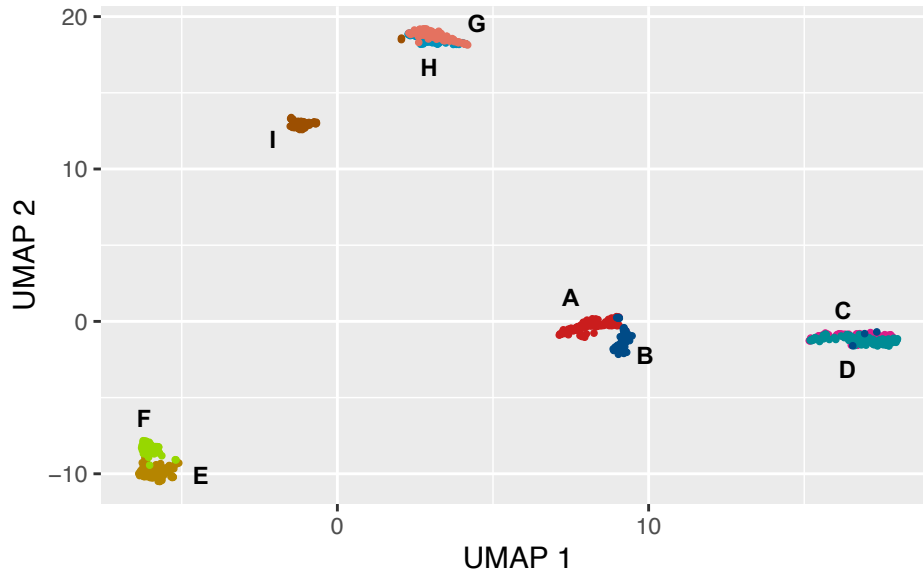


(b) Simulated tree with cell 30% and SNV 30% error rate.

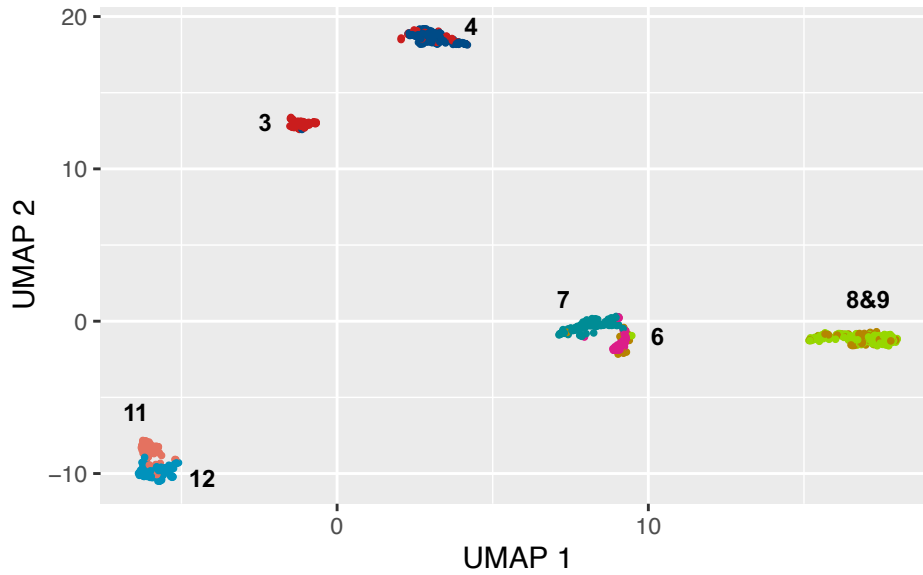


(c) Cell mutational burden (CMB) with varying cell and placement error rates.

Figure C.16: **Analysis of cell mutational burden (CMB) on a simulated instance with varying error rates in $\{0\%, 15\%, 30\%\}$ for both cell and SNV placement.** (a) The simulated ground truth tree with 0% cell placement error rate and 0% SNV placement error rate. (b) The simulated ground truth tree with 30% cell placement error rate and 30% SNV placement error rate. (c) Cell mutational burden (CMB) comparison between cells within (blue) and outside of (red) each clade in the inferred clonal tree at varying cell and SNV placement error rates.



(a) Laks et al. [24] cell clusters



(b) Phertilizer cell clusters

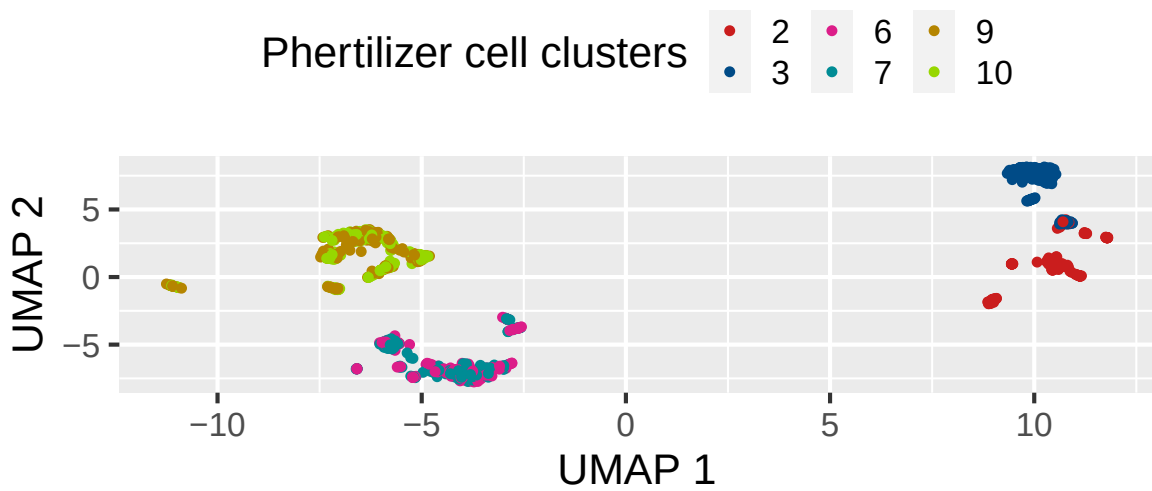
Figure C.17: UMAP for the high-grade serous ovarian cancer patient depicted with inferred cell clusterings.

Tumor	Cells	SNVS	Coverage	Phertilizer clones	Minussi et al. super clones	Minussi et al. subclones	SBMclone clones
TN1	1100	13,934	0.031×	6	4	17	1
TN2	1024	8,474	0.039×	6	4	15	1
TN3	1101	13,064	0.020×	1	4	9	1
TN4	1307	14,333	0.022×	5	4	22	1
TN5	1238	10,271	0.020×	1	4	7	1
TN6	1205	11,827	0.017×	1	4	15	1
TN7	915	7,873	0.019×	1	3	18	1
TN8	1224	73,359	0.021×	2	4	15	1

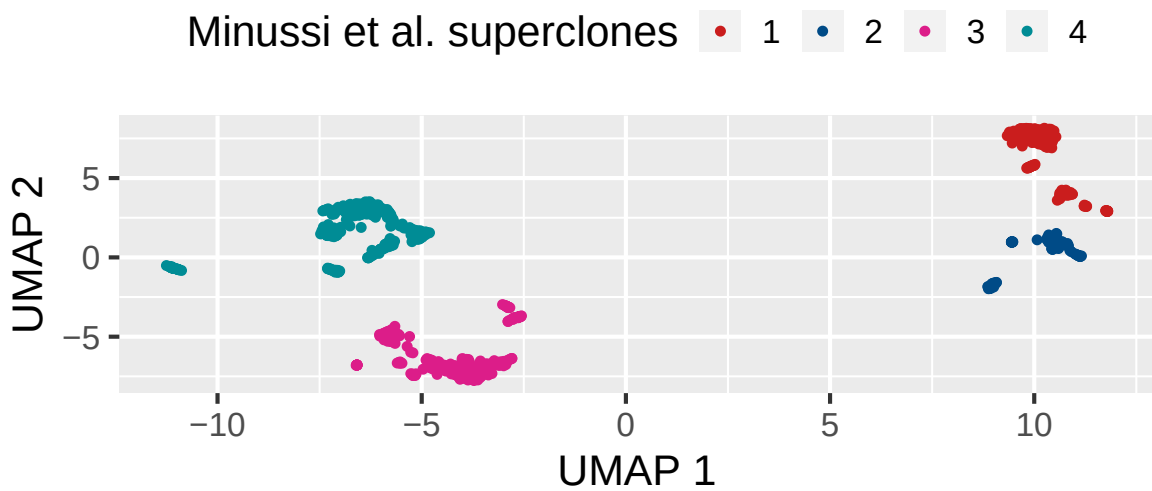
Table C.1: **Summary of results on the Minussi et al. [23] breast cancer data.** Table depicting the tumor, number of cells and SNVs, the coverage, and the number of clones inferred by different methods for the eight analyzed breast cancer tumors.

Processing the triple negative breast cancer tumors sequencing data from ACT.

We merged reads from all cells into a single FASTQ file for each sample, aligned the merged reads to the Human reference genome hg19 (GRCh37) using bowtie2 (v.2.4.4), and sorted by samtools (v.1.15) forming pseudo-bulk samples. We then get a set of SNVs by running Mutect2 [154] in tumor-only mode on each pseudo-bulk sample, followed by FilterMutectCalls in GATK. We further select SNVs whose read depth is greater or equal to 11, VAF greater than 0.033, and have greater or equal to 4 variant reads in the pseudo-bulk sample. In order to match the bins in the original paper, we directly used the normalized read counts from ACT and find the low-dimensional embedding using UMAP [175].

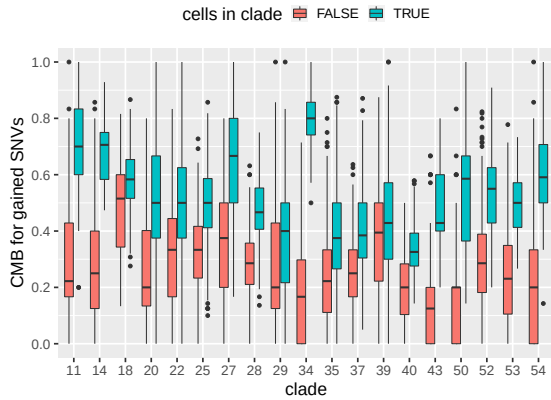


(a) PHERTILIZER UMAP for the triple negative breast tumor TN1.

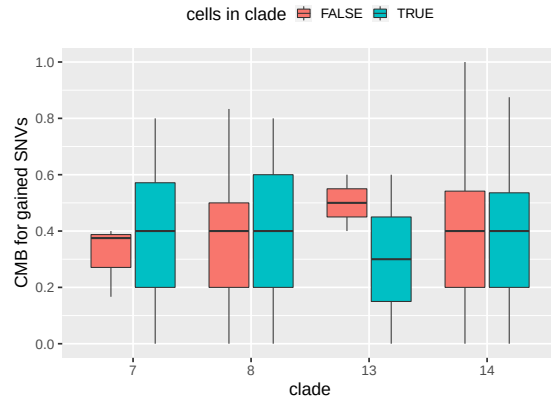


(b) Minussi et al. [23] UMAP for the triple negative breast tumor TN1.

Figure C.18: Cell clustering comparison for triple negative breast tumor TN1 in embedding space between PHERTILIZER and Minussi et al. [23]



(a) Triple negative breast tumor TN3



(b) Triple negative breast tumor TN5

Figure C.19: Cell mutational burden per clade in trees inferred by BASELINE+SCITE for tumors TN3 and TN5

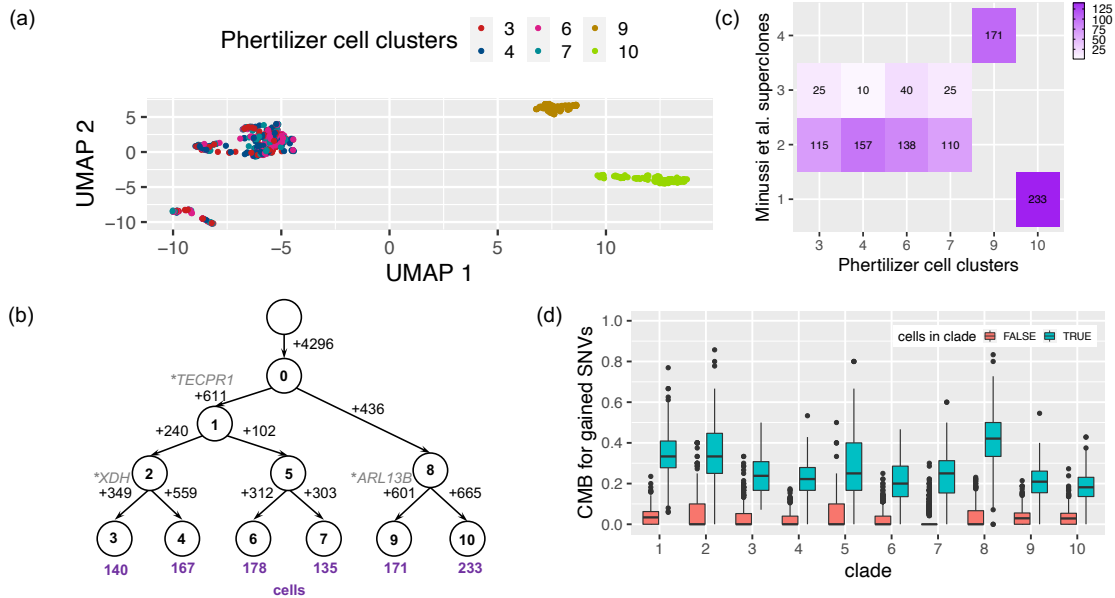


Figure C.20: **PHERTILIZER** inferred clonal tree for breast cancer tumor TN2. (a) UMAP for the triple negative breast tumor TN2. (b) The tree inferred by **PHERTILIZER** with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs (*: stop-gain variant). (c) A mapping between **PHERTILIZER**'s cell clusters and the Minussi et al. [23] superclones. (d) The CMB distribution for the inferred clades for TN2.

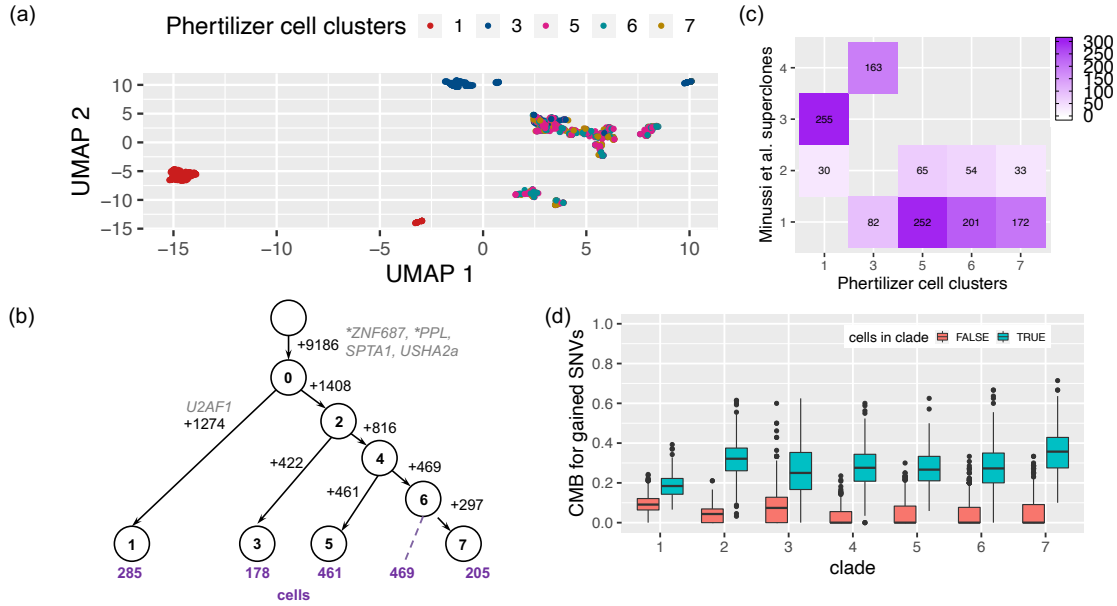


Figure C.21: **PHERTILIZER** inferred clonal tree for breast cancer tumor TN4. (a) UMAP for triple negative breast tumor TN4. (b) The tree inferred by **PHERTILIZER** with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs (*: stop-gain variant). (c) A mapping between **PHERTILIZER**'s cell clusters and the Minussi et al. [23] superclones. (d) The CMB distribution for the inferred clades for TN4.

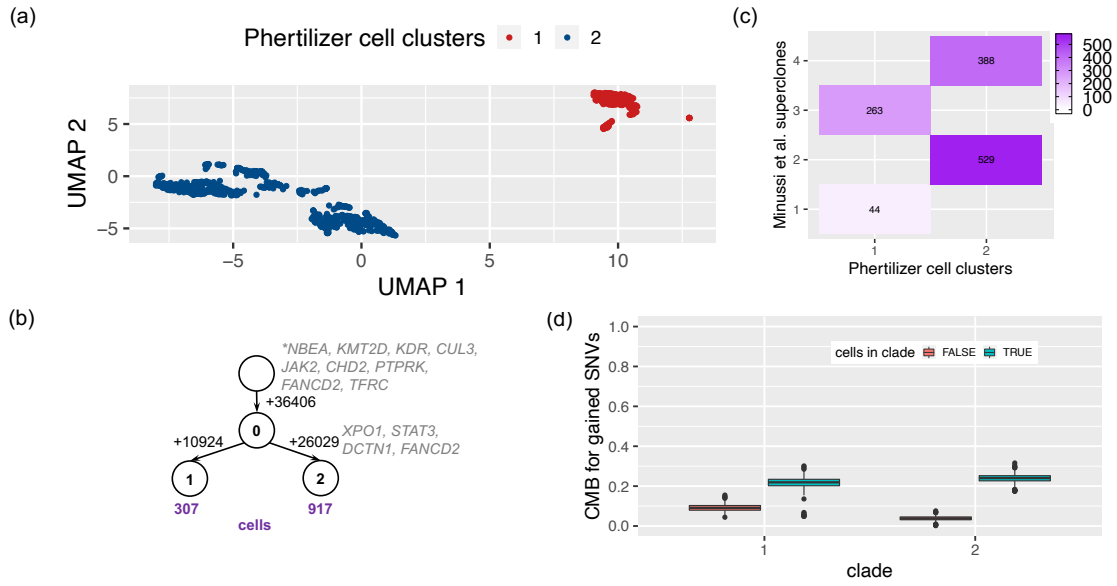


Figure C.22: **PHERTILIZER** inferred clonal tree for breast cancer tumor TN8. (a) UMAP for triple negative breast tumor TN8. (b) The tree inferred by **PHERTILIZER** with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs (*: stop-gain variant). (c) A mapping between **PHERTILIZER**'s cell clusters and the Minussi et al. [23] superclones. (d) The CMB distribution for the inferred clades for TN8.

APPENDIX D: SUPPLEMENTARY METHODS AND RESULTS FOR PHARMING

D.1 SUPPLEMENTARY METHODS

D.1.1 Modified DECiFER algorithm for DCF inference

One drawback of DECiFER [174] is that the genomic segment where each SNV is located is not considered and consequently, the algorithm enumerates a set of valid SNV trees for each SNV j starting from a set $\{(x, y)\}$ of observed allele-specific copy number states at SNV locus j . However, two SNVs j and j' located in the same segment may be assigned to SNV trees $\hat{T}$ and $\hat{T}'$ which respectively refine two distinct CNA trees S and S' — see Definition 6.5. Therefore, we propose a modification to DECiFER, referred to as DCF clustering, to ensure consistency of an underlying CNA tree for a segment with multiple SNVs. This modification works by adding a step to select a CNA tree S for each segment ℓ . During the SNV cluster and SNV tree assignment step of the DECiFER algorithm, we first enumerate the set $\mathcal{S}[\ell]$ of all CNA trees for the given copy number states in each segment ℓ . Then for each CNA tree S , we enumerate the set of valid SNV trees $\hat{\mathcal{T}}$, such that each SNV tree $\hat{T}$ is a refinement of CNA tree S . SNVs in segment ℓ may only be assigned to an SNV tree in the set $\hat{\mathcal{T}}$. We use the DECiFER posterior probability to select a CNA tree S and a corresponding assignment of SNVs to valid SNV trees for all segments prior to moving on to the cluster center optimization stage of the DECiFER algorithm.

In practice, the number of inferred segments by copy number callers may be large, i.e., $r > 100$, and our proposed modification increases the running time by a factor of r . In these cases, our implementation offers a speed by randomly sampling a user-specified number of segments during each restart. This is a valid approach because the underlying assumption is that the SNVs will cluster into a small number k of clusters across the genome. Therefore, only a small number of segments may be needed to detect the presence of an SNV cluster. It also allows us to quickly prune poor random initializations of the DCFs δ . After running our DCF clustering algorithm for a specified number of restarts, we return the top n inferred DCFs δ with the highest posterior probability and uses these to initialize restarts of our DCF clustering for all r segments.

D.1.2 Inferring a clonal tree for a single segment

A tree $T[\ell]$ is a clonal tree for segment ℓ such that the genotypes $g(v, j)$ are restricted to the set $\sigma^{-1}(\ell) \subseteq [m]$ of SNVs in located in segment ℓ . Therefore, the problem of inferring tree $T[\ell]$ is a special case of the CITCN problem where only a single segment ℓ is considered. We describe our algorithm to solve this special case of the CITCN problem for a single segment ℓ given fixed DCFs δ , SNV cluster tree T_m .

Since we focus on a single segment ℓ , we expect the size of the set $\{(\bar{x}_{i,\ell}, \bar{y}_{i,\ell}), i \in [n]\}$ depicting the unique allele-specific copy number states occurring in segment ℓ to be small. Consequently, it is feasible to enumerate a set $\mathcal{S}$ of CNA trees (Def. 6.3) such that there is a bijection between copy number states μ for CNA tree S and the observed set $\{(\bar{x}_{i,\ell}, \bar{y}_{i,\ell}), i \in [n]\}$ of allele-specific copy number states given a threshold on the proportion of cells that support an observed copy number state in segment ℓ . We use a default threshold of 0.05 for this parameter.

Consider a fixed CNA tree S and an augmented mutation cluster T_m , i.e., $|V(T_m)| = k + 1$ after attaching a normal clone to the root of T_m . A clonal tree $T[\ell]$ that is a refinement of CNA tree S and mutation cluster T_m , per the definition below, explicitly relates the evolutionary relationships between SNV clusters to the evolutionary relationships of copy number states for a single segment. In particular, for an SNV cluster q and copy number state (x, y) , the segment tree determines if cluster q was introduced before, after or on a distinct branch from the introduction of copy number state (x, y) .

Another way to think of clonal tree $T[\ell]$ is that it defines a mapping $\rho : [k] \rightarrow 2^{|\hat{\mathcal{T}}(S)|}$ of SNV clusters to subsets of SNV trees, where $\hat{\mathcal{T}}(S)$ is the set of SNV trees that are refinements of CNA tree S . Thus, if SNV j was introduced in SNV cluster q , the set $\rho(q)$ of SNV trees depicts possible evolutionary histories for SNV j . For a given CNA tree S and a SNV cluster tree T_m , we utilize a modification of PACTION integer linear program (ILP) proposed by Sashittal et al. [166] to find the most parsimonious refinement of a CNA tree and SNV tree given clonal proportions of each feature. In our modified ILP, we remove constraints related to clonal abundance and the objective function to enumerate all possible refinements $\mathcal{T}[S \times T_m]$ of CNA tree S and SNV cluster tree T_m . Each refinement (i) yields a clonal tree T with copy number states specified for each node, (ii) a mapping ω assigning each SNV cluster q to a node $u \in V(T)$ and (iii) a unique mapping ρ . Thus, given such a refinement, the problem of inferring a clonal tree for a single segment, reduces to finding an optimal assignment of each SNV j to an SNV cluster q and for an optimal assignment of each SNV j to an SNV tree $\hat{T} \in \rho(q)$. However, the cost of given SNV clustering ψ and SNV to SNV tree assignment is also dependent on finding an optimal cell assignment ϕ . This is achieved

via a coordinate descent approach that includes an initialization of genotypes g , followed by alternating optimization of cell assignment ϕ and then genotypes g .

Initialization of genotypes. For each SNV j and each SNV tree $\hat{T} \in \bigcup_{q \in [k]} \rho(q)$, we find a cell assignment $\hat{\phi}$ such that $\hat{\phi}(i) = \operatorname{argmin}_{u \in V(\hat{T})} J(\hat{T}, u)$ for each cell i , where in a slight abuse of notation $J(\hat{T}, u)$ is the cost of assigning cell i to node u in SNV tree $\hat{T}$. Then $c(j, \hat{T}) = J(\hat{T}, \hat{\phi})$ is the cost of SNV j having the evolutionary history depicted by SNV tree $\hat{T}$. However, by construction the map ρ is not injective and multiple SNV clusters may map to the same subset of SNV trees if they have the same copy number state. To differentiate between these SNV clusters, we utilize the DCFs δ to uniquely determine the initial SNV cluster assignment for each SNV j . Following the Satas et al. [174] probabilistic model for DCFs, we use a binomial model to compute the following likelihood

$$c'(j, \hat{T}, q) = P \left(\sum_{i \in [n]} a_{i,j} \mid \sum_{i \in [n]} d_{i,j}, V(\delta(q)), \hat{T} \right), \quad (\text{D.1})$$

where $V(d)$ is the function that converts DCF d to a VAF utilizing SNV tree $\hat{T}$.

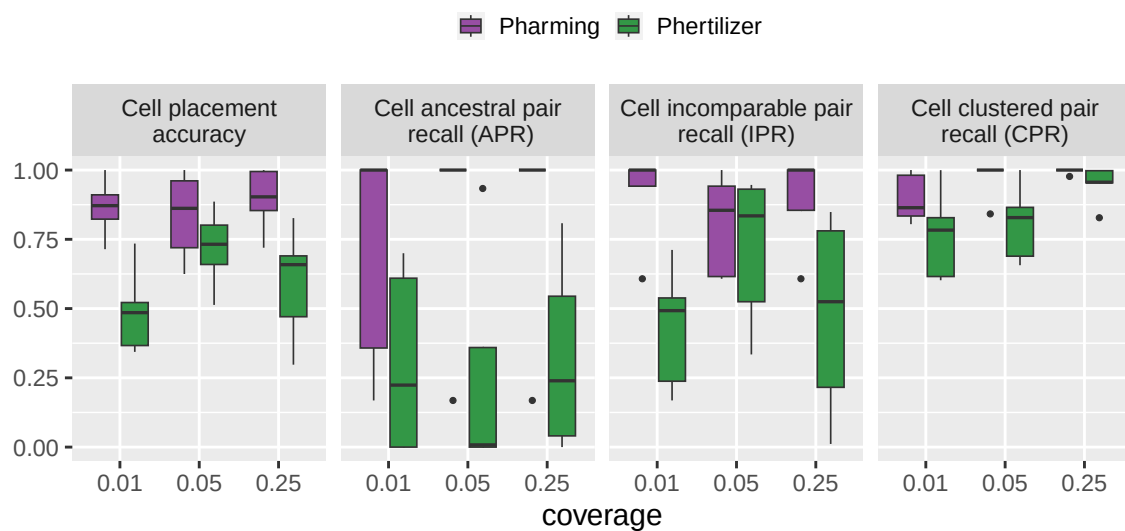
We then assign SNV j to an SNV cluster q , i.e., $\psi(j) = q$, and SNV tree $\hat{T} \in \rho(q)$, i.e., $\text{SNVclustertree}(j) = \hat{T}$, such that $c(j, \hat{T})$ is minimum and then $c'(j, \hat{T}, q)$ is maximum.

Then, given SNV clustering ψ and SNV tree assignments SNVclustertree , we initialize the genotypes g for each node u and SNV j in clonal tree $T[\ell]$. Following the initialization of the genotypes, we next begin our coordinate descent algorithm to alternately optimize cell assignment ϕ and genotypes g for clonal tree $T[\ell]$.

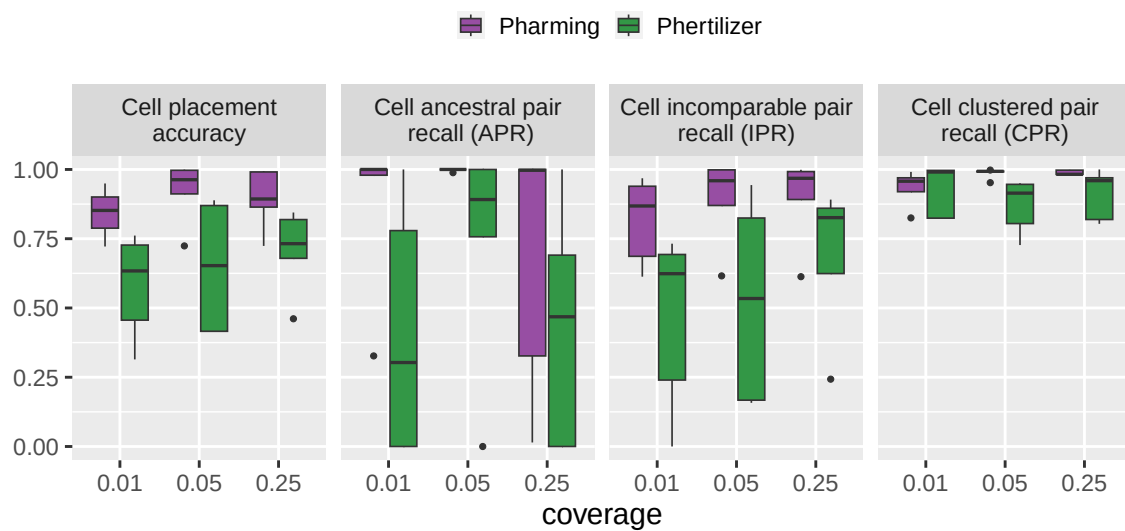
Optimizing cell assignments. Given a clonal tree $T[\ell]$ for segment ℓ and genotypes g , we then seek the cell assignment ϕ that minimizes $J(T[\ell], \phi)$. This is accomplished by independently finding a optimal assignment of each cell i to a node u in clonal tree $T[\ell]$, i.e., cell i , $\phi(i) = \operatorname{argmin}_{u \in V(T[\ell])} J(T[\ell], u)$ for each where we abuse notation slightly such that $J(T[\ell], u)$ is the cost of assigning cell i to node u in clonal tree $T[\ell]$.

Optimizing genotypes. Given a clonal tree $T[\ell]$ for segment ℓ and a fixed cell assignment ϕ , we then infer genotypes g that optimizes $J(T[\ell], \phi)$. This is accomplished by independently computing the cost of assigning an SNV j to each SNV cluster q and SNV tree $\hat{T} \in \rho(q)$ and corresponding SNV tree in $\rho(q)$. For each SNV j , we select the SNV cluster q^* and SNV tree $\hat{T}$ that yield the genotypes $g(:, j)$ across all nodes in clonal tree $T[\ell]$ that minimizes $J(T[\ell], \phi)$ for SNV j .

D.2 SUPPLEMENTARY RESULTS

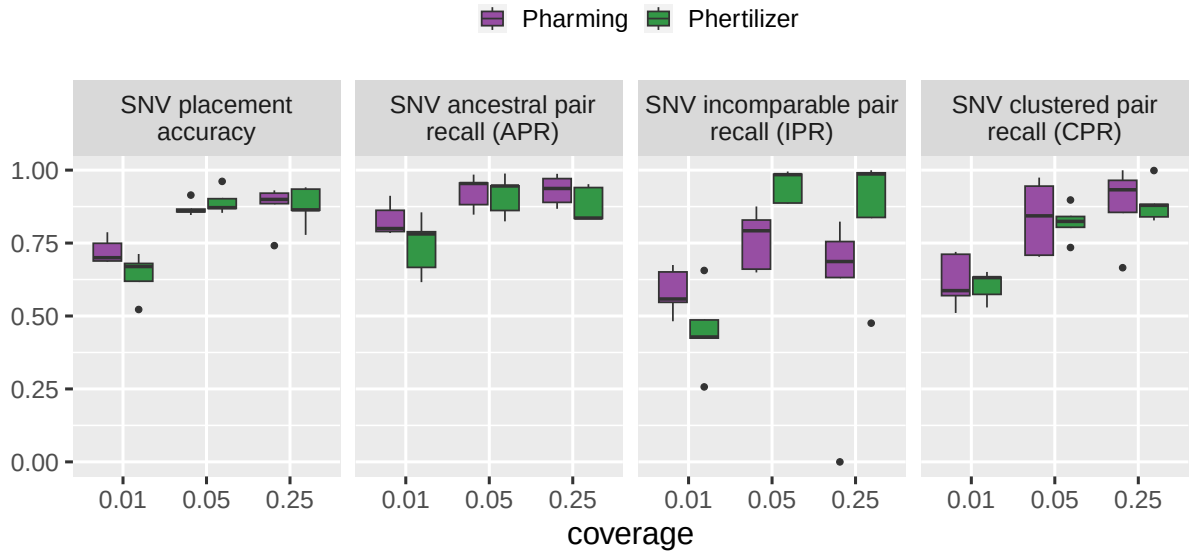


(a) CNA error rate = 0

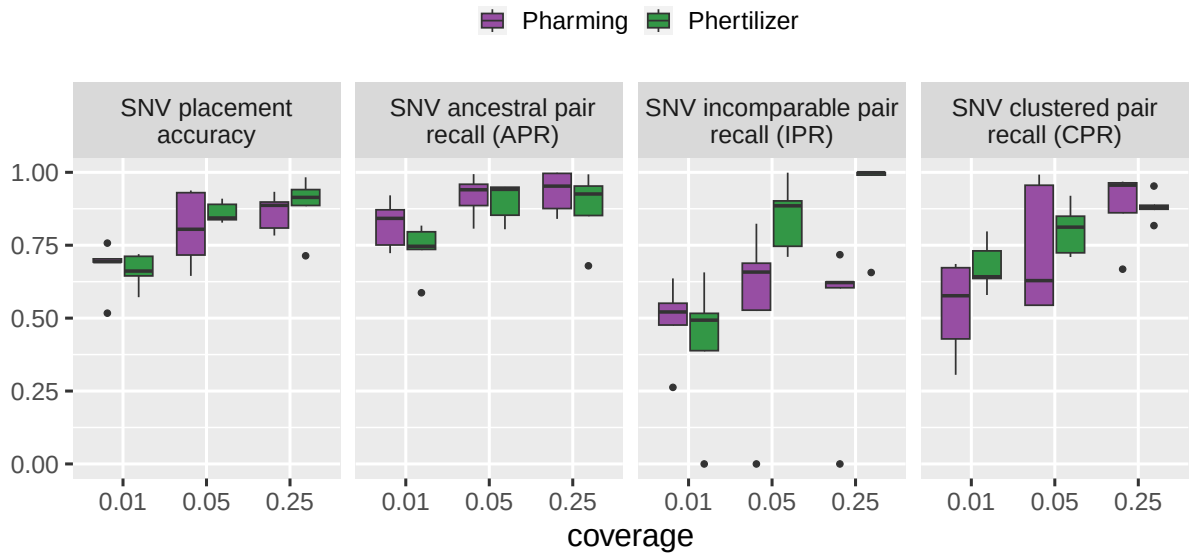


(b) CNA error rate = 0.035

Figure D.1: Cell placement accuracy metrics.



(a) CNA error rate = 0



(b) CNA error rate = 0.035

Figure D.2: SNV placement accuracy metrics.

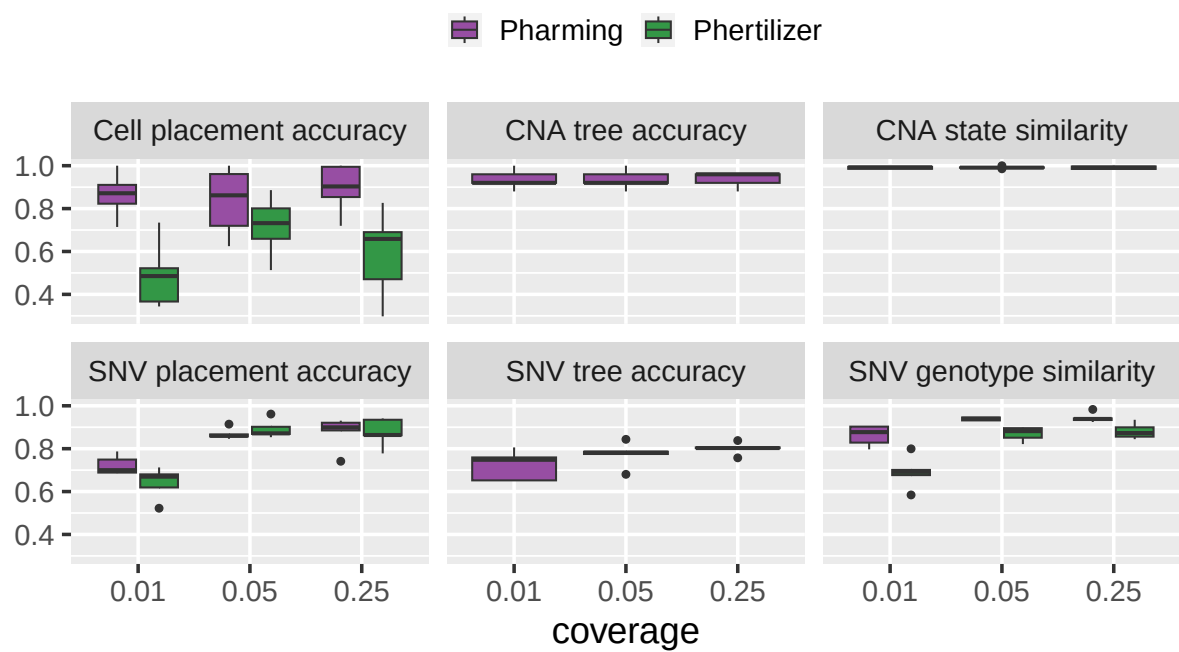


Figure D.3: **PHARMING** simulations results for copy number error rate = 0.

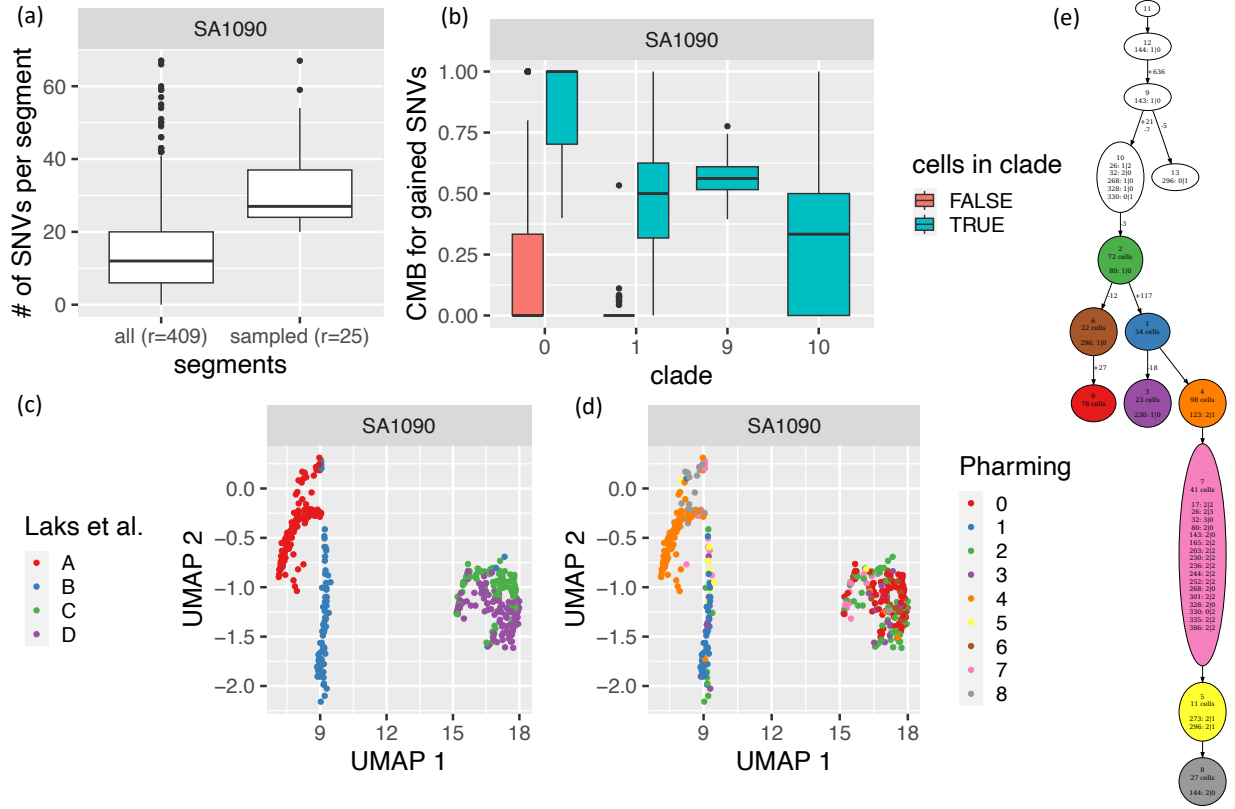


Figure D.4: **PHARMING** analysis of the SA1090 ovarian cancer cell lines sequenced with DLP+ [24]. (a) Distributions of the number of SNVs in all segments ($r = 409$) and the number of SNVs in the sampled segments ($r = 25$). (b) Distribution of the cell mutational burden (CMB) for the clades in inferred PHARMING clonal tree. (c) UMAP projection of the cells in SA1090 colored by Laks et al. [24] cluster. (d) UMAP projection of the cells in SA1090 colored by PHARMING cell assignment ϕ . (e) PHARMING inferred clonal tree for the 25 sampled segments of SA1090. Nodes with more 5 cells assigned are colored to be consistent with (d) and labeled by the number of assigned cells as well as the introduction of a copy number state change, i.e., segment: $x|y$. The incoming edges are labeled by the number of newly introduced SNVs and the number of lost of SNVs when applicable.

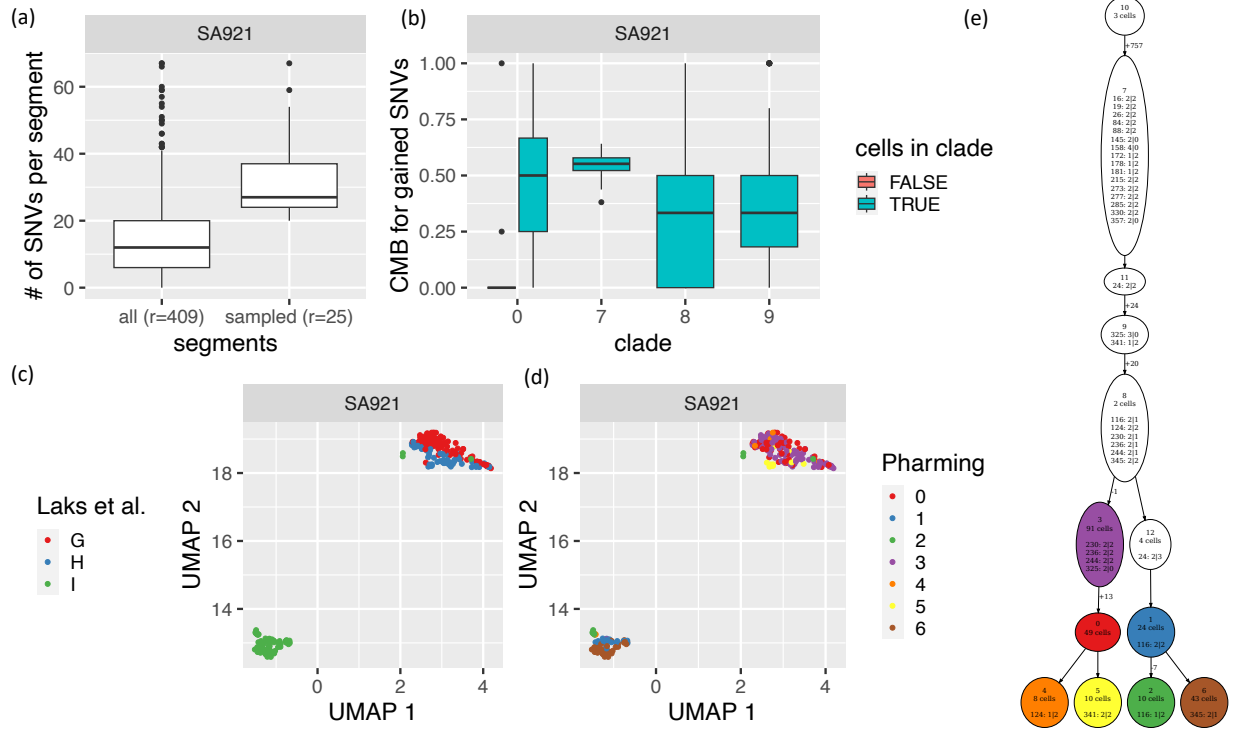


Figure D.5: **PHARMING analysis of the SA921 ovarian cancer cell lines sequenced with DLP+ [24].** (a) Distributions of the number of SNVs in all segments ($r = 409$) and the number of SNVs in the sampled segments ($r = 25$). (b) Distribution of the cell mutational burden (CMB) for the clades in inferred PHARMING clonal tree. (c) UMAP projection of the cells in SA921 colored by Laks et al. [24] cluster. (d) UMAP projection of the cells in SA1090 colored by PHARMING cell assignment ϕ . (e) PHARMING inferred clonal tree for the 25 sampled segments of SA921. Nodes with more 5 cells assigned are colored to be consistent with (d) and labeled by the number of assigned cells as well as the introduction of a copy number state change, i.e., segment: $x|y$. The incoming edges are labeled by the number of newly introduced SNVs and the number of lost of SNVs when applicable.

APPENDIX E: SUPPLEMENTARY METHODS AND RESULTS FOR TRIBAL

E.1 EVOLUTIONARY MODEL FOR CLASS SWITCH RECOMBINATION

A dependence tree $\bar{T}$ with n nodes, sometimes referred to as a state tree, is a tree that defines the conditional independence structure of the random variables associated with the nodes of the tree [183, 184]. Simply put, it is a type of Bayesian network, where the underlying directed acyclic graph is a tree. For each node i in dependence tree $\bar{T}$, we associate a random variable $Y_i \in S$, where S is a discrete state space. Additionally, Y_1 is the random variable associated with the root node. The joint probability $\mathbf{Y} = (Y_1, \dots, Y_n)$ of these random variables given the underlying structure of dependence tree $\bar{T}$ is defined as follows

$$P(\mathbf{Y} \mid \bar{T}) = P(Y_1) \prod_{i=2}^n P(Y_i \mid Y_{\phi(i)}), \quad (\text{E.1})$$

where $[n] = \{1, \dots, n\}$ and $\phi(i)$ is a function that returns the parent of node i specified by dependence tree $\bar{T}$. Like Markov chains, this model is parameterized by a distribution over the starting state, $\pi_s = P(Y_1 = s)$, where $\sum_{i \in [r]} \pi_s = 1$, and transition probabilities $p_{s,t} = P(Y_i = t \mid Y_{\phi(i)} = s)$ from state s to t . Transition probabilities $\mathbf{P} = [p_{s,t}] \in [0, 1]^{|S| \times |S|}$ have the property that $\sum_{t \in S} p_{s,t} = 1$ for every state s .

We model class switch recombination with a dependence tree $\bar{T}$ for each lineage tree T with nodes $V(\bar{T}) = V(T)$ and edges $E(\bar{T}) = E(T)$ and isotype labels $\beta(v) = Y_v$ for every node v . In words, the corresponding dependence tree $\bar{T}$ for B cell lineage tree T has the same topology but the dependence tree $\bar{T}$ has a random variable for isotype associated with each node.

For the leaf nodes $v_i \in L(T)$, i.e., the sequenced B cells, we observe isotype b_i directly from scRNA-seq data, i.e., $P(Y_{v_i} = b_i) = 1$. Additionally, root node v_0 in T , represents the naive B cell post V(D)J recombination and therefore $P(Y_{v_0} = 1) = 1$, meaning that root of the dependence tree has isotype state IgM and $\pi_1 = 1$. This means that any dependence

¹⁴This chapter contains identical material with an adapted introduction previously published as the bioRxiv preprint “TRIBAL: Tree inference of B cell clonal lineages” [176], which is an extended version of the proceedings for RECOMB 2024 [177]. The figures, tables, and text in this chapter are reproduced under the Creative Commons Attribution 4.0 International License. This work was done in conjunction with M. El-Kebir, A.A. Khan, D. Reiman, M.S. Roddur and Y. Qi. MEK and AAK conceived the project; LLW, MEK established theory and the method; MSR and YQ provided a proof of NP hardness. DR prepared the experimental data. LLW performed experiments; LLW, MEK, AAK contributed to the writing of this chapter.

tree without $Y_{v_0} = 1$ has zero probability and we omit the initial state probability term. For any other node u in T , we have $\beta(u) = Y_u$. Thus, to compute (E.1) for isotype labels $\beta(v)$, it suffices to know the isotype transition probabilities $\mathbf{P}$, which we formally define below.

(Main Text) Definition 2. An $r \times r$ matrix $\mathbf{P} = [p_{s,t}]$ is an *isotype transition probability matrix* provided for all isotypes $s, t \in [r]$ it holds that (i) $p_{s,t} = 0$ if $t > s$, (ii) $p_{s,t} \geq 0$, and (iii) $\sum_{t=1}^r p_{s,t} = 1$ for all isotypes $s \in [r]$.

The additional conditions on the transition probabilities beyond row stochasticity on the transition probabilities are to properly model the irreversibility of class switch recombination. Using the above model for class switch recombination, we compute the likelihood $\text{CSR}(T, \beta, \mathbf{P})$ for observed isotypes $\mathbf{b}$ given a lineage tree with isotypes β and isotype transition probabilities as follows.

$$\text{CSR}(T, \beta, \mathbf{P}) = \Pr(\mathbf{b} \mid T, \beta, \mathbf{P}) \quad (\text{E.2})$$

$$\begin{aligned} &= \prod_{(u,v) \in E(T)} p_{\beta(u), \beta(v)} \\ &= \prod_{v \in V(T) \setminus \{v_0\}} \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{\mathbf{1}(\beta(v)=t, \beta(\phi(v))=s)} \end{aligned} \quad (\text{E.3})$$

$$= \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{N_{s,t}} \quad (\text{E.4})$$

where $N_{s,t}$ is the count of occurrences in lineage tree T such that $\beta(v) = t$ and $\beta(\phi(v)) = s$. This is easily extended for a forest of k lineage trees $T_1, \dots, T_k$ with corresponding isotypes $\beta_1, \dots, \beta_k$. Given isotype transition probabilities $\mathbf{P}$, the joint probabilities $\text{CSR}(T_1, \beta_1, \mathbf{P}), \dots, \text{CSR}(T_k, \beta_k, \mathbf{P})$ are conditionally independent, resulting in the joint likelihood

$$\prod_{j=1}^k \text{CSR}(T_j, \beta_j, \mathbf{P}) = \prod_{j=1}^k \Pr(\mathbf{b}_j \mid T_j, \beta_j, \mathbf{P}) \quad (\text{E.5})$$

$$\begin{aligned} &= \prod_{j=1}^k \prod_{(u,v) \in E(T_j)} p_{\beta_j(u), \beta_j(v)} \\ &= \prod_{j=1}^k \prod_{v \in V(T_j) \setminus \{v_0\}} \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{\mathbf{1}(\beta_j(v)=t, \beta_j(\phi(v))=s)} \end{aligned} \quad (\text{E.6})$$

$$= \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{\sum_{j=1}^k N_{j,s,t}} \quad (\text{E.7})$$

where $N_{j,s,t}$ is the count of occurrences in lineage tree T_j such that $\beta(v) = t$ and $\beta(\phi(v)) = s$.

E.2 COMBINATORIAL CHARACTERIZATION AND COMPLEXITY RESULTS

E.2.1 B cell lineage forest inference

Recall the B CELL LINEAGE FOREST INFERENCE PROBLEM (BLFI) from Sec. 7.2.1, restated below for convenience.

(Main Text) Problem 1 (B CELL LINEAGE FOREST INFERENCE (BLFI)). Given MSAs $\mathbf{A}_1, \dots, \mathbf{A}_k$ and isotypes $\mathbf{b}_1, \dots, \mathbf{b}_k$ for k clonotypes, find isotype transition probabilities $\mathbf{P}^*$ for r isotypes and lineage trees $T_1^*, \dots, T_k^*$ for $(\mathbf{A}_1, \mathbf{b}_1), \dots, (\mathbf{A}_k, \mathbf{b}_k)$ whose nodes are labeled by sequences $\alpha_1^*, \dots, \alpha_k^*$ and isotypes $\beta_1^*, \dots, \beta_k^*$, respectively, such that $\sum_{j=1}^k \text{SHM}(T_j^*, \alpha_j^*)$ is minimum and then $\prod_{j=1}^k \text{CSR}(T_j^*, \beta_j^*, \mathbf{P}^*)$ is maximum.

Theorem E.1. The BLFI problem is NP-hard even if $k = 1$ and $r = 1$.

We prove that the BLFI problem is NP-hard via a simple reduction from the LARGE PARSIMONY problem [211]. Although this problem is well known, we restate it here for completeness.

Problem E.1 (LARGE PARSIMONY (LP)). Given a matrix $\mathbf{A} \in \{0, 1\}^{n \times m}$, find a rooted tree T whose nodes are labeled by sequences $\alpha : V(T) \rightarrow \{0, 1\}^m$ such that the n leaves are labeled by the rows of $\mathbf{A}$ and $\sum_{(u,v) \in E(T)} D(\alpha(u), \alpha(v))$ is minimum.

The reduction to BLFI proceeds by using the same MSA $\mathbf{A}$ directly for a single clonotype, i.e., $k = 1$. Additionally, we restrict the number r of isotypes to 1, and set isotypes $\mathbf{b} = [1]^n$.

Lemma E.1. Tree T and node labeling α form an optimal solution to LP instance $\mathbf{A}$ if and only if tree T , sequences α and isotypes β , the isotype transition probabilities $\mathbf{P}$ form an optimal solution to BLFI instance $(\mathbf{A}, \mathbf{b})$.

Proof. ($\Rightarrow$) Let tree T and sequence labeling α be an optimal solution to the LP problem. We will show that T and α can be augmented to form an optimal solution to the corresponding BLFI problem. We set $\mathbf{P} = [1]$. We also set $\beta(v) = 1$ for all nodes $v \in T$. We claim that $(T, \alpha, \beta, \mathbf{P})$ form an optimal solution to BLFI. Assume for a contradiction there exists a solution $(T', \alpha', \beta', \mathbf{P}')$ such that $\text{SHM}(T', \alpha') < \text{SHM}(T, \alpha)$, or $\text{SHM}(T', \alpha') = \text{SHM}(T, \alpha)$ and $\text{CSR}(T', \beta', \mathbf{P}') > \text{CSR}(T, \beta, \mathbf{P})$. Clearly, any feasible solution to BLFI must use $\beta(v) = 1$ for all nodes v and $\mathbf{P} = [1]$ as $r = 1$. This means that any feasible solution to BLFI will

have a CSR objective value of 1. Therefore, $\text{CSR}(T', \beta', \mathbf{P}') = \text{CSR}(T, \beta, \mathbf{P}) = 1$. Hence, $\text{SHM}(T', \alpha') < \text{SHM}(T, \alpha)$. As can be seen in (7.1), the SHM objective equals the objective of the LP problem. Therefore, T' and α' have a lower parsimony score than T and α , a contradiction.

($\Leftarrow$) Let $(T, \alpha, \beta, \mathbf{P})$ be an optimal solution to BLFI. Again, as the SHM objective equals the objective of the LP problem, it directly follows that (T, α) form an optimal solution to the LP problem instance. QED.

E.2.2 Most parsimonious tree refinement

Combinatorial characterization. Recall from the main text the definition of isotype transition probabilities $\mathbf{P}$, the CSR log-likelihood for isotypes $\mathbf{b}$ of a tree T with nodes labeled by isotypes β , and the MOST PARSIMONIOUS TREE REFINEMENT problem, provided below for convenience.

(Main Text) Definition 2. An $r \times r$ matrix $\mathbf{P} = [p_{s,t}]$ is an *isotype transition probability matrix* provided for all isotypes $s, t \in [r]$ it holds that (i) $p_{s,t} \geq 0$, (ii) $p_{s,t} = 0$ if $s > t$, and (iii) $\sum_{t=1}^r p_{s,t} = 1$ for all isotypes $s \in [r]$.

$$\log \text{CSR}(T, \beta, \mathbf{P}) = \log \prod_{(u,v) \in E(T)} p_{\beta(u), \beta(v)} = \sum_{(u,v) \in E(T)} \log p_{\beta(u), \beta(v)}. \quad (\text{E.8})$$

(Main Text) Problem 2 (MOST PARSIMONIOUS TREE REFINEMENT (MPTR)). Given a tree T on n leaves, isotypes $\mathbf{b} = [b_0, \dots, b_n]$ and isotype transition probabilities $\mathbf{P}$, find a tree T' with root v'_0 and isotype labels $\beta' : V(T') \rightarrow [r]$ such that (i) T' is a refinement of T , (ii) $\beta'(v'_0) = b_0 = 1$, (iii) $\beta'(v'_i) = b_i$ for each leaf $v'_i \in \{v'_1, \dots, v'_n\}$ and (iv) $\log \text{CSR}(T', \beta', \mathbf{P})$ is maximum.

Let σ be a mapping from $V(T')$ to $V(T)$ that reverses all EXPAND operations of each node u' in refinement T' in order to obtain back the node $\sigma(u') = u$ from which it was derived in the original tree T . We say that an isotype labeling $\beta' : V(T') \rightarrow [r]$ of T' is *transitory* if along each directed edge (u', v') of T' either the isotype changes or u' and v' correspond to two distinct nodes of T . More formally, we have the following definition.

Definition E.1. Let T' be a refinement of a tree T whose leaves are labeled by isotypes $\mathbf{b}$. Then, an isotype labeling β' of T' is *transitory* provided (i) $\beta'(v'_0) = 1$ where v'_0 is the root

of T' , (ii) $\beta'(v') = b_{\sigma(v')}$ for each leaf $v' \in L(T')$, (iii) $\beta'(u') \leq \beta(v')$ for each edge (u', v') of T' , and (iv) $\beta'(u') = \beta'(v')$ only if $\sigma(u') \neq \sigma(v')$ for each edge (u', v') of T' .

Importantly, among the set of optimal solutions (T', β') to each MPTR problem instance $(T, \mathbf{b}, \mathbf{P})$ there exist solutions where β' is transitory.

Lemma E.2. Let $(T, \mathbf{b}, \mathbf{P})$ be an MPTR problem instance. There exist an optimal solution (T', β') where β' is transitory.

Proof. We prove this by contradiction. Let (T', β') be an optimal solution where β' is not transitory. First, observe that it holds that $\beta'(u') \leq \beta(v')$ for each edge (u', v') of T' . To see why, if there were an edge (u', v') such that $\beta'(u') > \beta(v')$ then $\text{CSR}(T', \beta', \mathbf{P}) = -\infty$ as $\log p_{s,t} = -\infty$ if $s > t$. However, setting $\beta'(u') = 1$ for nodes u' not in $L(T')$ would result in log-likelihood greater than $-\infty$. Since (T', β') is a feasible solution to MPTR respecting irreversibility of isotype transitions, it means that condition (iv) of Definition E.1 is violated. Let (u', v') be an edge such that $\beta'(u') = \beta'(v')$ and $\sigma(u') = \sigma(v')$. We can contract this edge, retaining the isotype labeling β' for the remaining nodes, such that the resulting tree remains a refinement of T and the objective value remains unchanged as $\log p_{s,s} = 0$. Repeating this procedure for all edges (u', v') such that $\beta'(u') = \beta'(v')$ and $\sigma(u') = \sigma(v')$ results in (T'', β'') , where T'' is a refinement of T labeled by β'' , with the same optimal score as (T', β') . Clearly, (T'', β'') is transitory, proving the lemma. QED.

E.3 SUPPLEMENTARY METHODS

E.3.1 Tree refinement

We solve an instance $(T, \mathbf{b}, \mathbf{P})$ of the MPTR problem (Fig. E.1a) by reducing it to a graph problem. Given an instance $(T, \mathbf{b}, \mathbf{P})$ of the MPTR, we construct a directed graph $G_{T,\mathbf{b}}$, called the expansion graph, with nodes $V(G_{T,\mathbf{b}}) \subseteq V(T) \times [r]$ and edges $E(G_{T,\mathbf{b}})$. At a high level, nodes of $V(G_{T,\mathbf{b}})$ are of the form (u, s) where $u \in V(T)$ is a node of the input tree T and $s \in [r]$ is an isotype state. Formally, we have the following definition.

Definition E.2. A directed graph $G_{T,\mathbf{b}}$ is an *expansion graph* of a rooted tree T whose leaves are labeled by isotypes $\mathbf{b}$ provided $V(G_{T,\mathbf{b}}) = \bigcup_{u \in V(T)} X(u)$ where

$$X(u) = \begin{cases} \{(u, b_u)\}, & \text{if } u \in L(T), \\ \{(u, s) \mid s \in \{1, \dots, \max\{b_v \mid v \in L(T_u)\}\}\}, & \text{if } u \in V(T) \setminus L(T), \end{cases} \quad (\text{E.9})$$

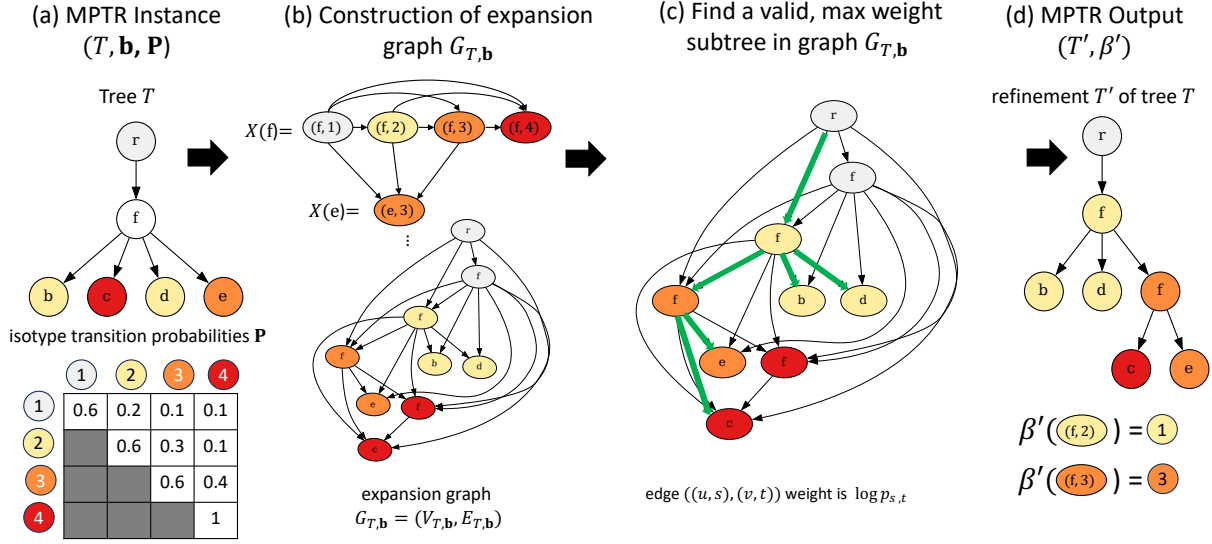


Figure E.1: **Algorithm for solving the MPTR problem** (a) An instance $(T, \mathbf{b}, \mathbf{P})$ of the MPTR problem. (b) To construct the expansion graph $G_{T,\mathbf{b}}$ for tree T whose leaves have isotypes $\mathbf{b}$, each original node u in $V(T)$ corresponds to a set $X(u)$ of nodes in $G_{T,\mathbf{b}}$. Edges are added to capture all transitory refinements of tree T . (c) We use the expansion graph $G_{T,\mathbf{b}}$ with weighted edges to find a valid, maximum weight subtree in $G_{T,\mathbf{b}}$, depicted in green. (d) This selected subtree is an optimal solution (T', β') to the MPTR problem instance $(T, \mathbf{b}, \mathbf{P})$.

and $E(G_{T,\mathbf{b}}) = \{((u,s), (v,t)) \mid (u,v) \in E(T), s \leq t\} \cup \{((u,s), (u,t)) \mid u \in V(T), s < t\}$.

In the above definition $X(u)$ is the set of nodes of $G_{T,\mathbf{b}}$ corresponding to node u of T , accounting for the fact that leaves u of T retain their isotype state in any refinement T' of T . On the other hand, internal nodes u of T may be subject to EXPAND operations such that the corresponding nodes of T' are assigned isotypes s ranging from state 1 to the maximum isotype state among all descendant leaves of u in T . The edges of $G_{T,\mathbf{b}}$ respect the irreversibility property of isotypes as well as the parental relationships of nodes of T . See Fig. E.1c for an example expansion graph $G_{T,\mathbf{b}}$.

We now define constrained subtrees, termed valid, of the expansion graph $G_{T,\mathbf{b}}$.

Definition E.3. A subtree T' of $G_{T,\mathbf{b}}$ is *valid* provided (i) T' is rooted at $(v_0, 1)$ where v_0 is the root of T and (ii) there is a unique edge $((u,s), (v,t))$ in $E(T')$ for each edge (u,v) of T .

We now show that the set of valid subtrees of $G_{T,\mathbf{b}}$ corresponding to trees T' with isotype labelings β' is equivalent to the set composed of pairs (T', β') where T' is a refinement of T and β' is a transitory isotype labeling of T' .

Lemma E.3. Let T' be a refinement of T whose leaves are labeled by isotypes $\mathbf{b}$ and let

β' be an isotype labeling of T' . Then, β' is transitory if and only if (T', β') induces a valid subtree of $G_{T, \mathbf{b}}$.

Proof. ($\Rightarrow$) Let β' be a transitory isotype labeling of T' . We start by showing that (T', β') induce a connected subtree of $G_{T, \mathbf{b}}$. First, let u' be a node of T' labeled by isotype $\beta(u')$. We claim that $(u', \beta(u')) \in X(u)$. We distinguish the two cases. First, $u' \in L(T')$. Let $u = \sigma(u')$ be the original leaf node u of T . Since β' is transitory, we have $\beta(u') = b_{\sigma(u')} = b_u$. Hence, $(u', \beta(u')) \in X(u)$ for each leaf node $u' \in L(T')$. Second, $u' \in V(T') \setminus L(T')$. Let $u = \sigma(u')$ be the original internal node u of T . Suppose for a contradiction $(u', \beta(u')) \notin X(u)$. This means that $\beta'(u') > \max\{b_v \in L(T_u)\}$. As such, there would be an edge (u'', v'') such that $\beta'(u'') > \beta'(v'')$ where u'' is a node in the subtree $T'_{u'}$ rooted at node u' . However, this would mean that β' would violate condition (iii) of Definition E.1, a contradiction. Thus, $(u', \beta(u')) \in X(u)$ for each internal node $u' \in V(T' \setminus L(T'))$. Hence, $(u', \beta(u')) \in V(G_{T, \mathbf{b}})$.

We now prove that each edge (u', v') of T' whose incident nodes are labeled by $(\beta'(u'), \beta'(v'))$ corresponds to an edge $((u', \beta'(u')), (v', \beta'(v')))$ of $G_{T, \mathbf{b}}$. This follows directly from conditions (iii) and (iv) of Definition E.1 and the definition of $E(G_{T, \mathbf{b}})$ in Definition E.2. This implies that the subgraph of $G_{T, \mathbf{b}}$ induced by (T', β') is a (connected) subtree of $G_{T, \mathbf{b}}$.

We now must show that this induced subtree of $G_{T, \mathbf{b}}$ is valid. By condition (i) of Definition E.1, we have that $\beta'(v'_0) = 1$ for the root v'_0 of T' . As such, the induced subtree of $G_{T, \mathbf{b}}$ is rooted at $(v'_0, 1)$. Finally, we must show there is a unique edge $((u, s), (v, t))$ in the induced subtree of $G_{T, \mathbf{b}}$ for each original edge (u, v) of T . This follows from the fact that T' is a refinement of T . Thus the subgraph of $G_{T, \mathbf{b}}$ induced by (T', β') is a valid subtree of $G_{T, \mathbf{b}}$.

($\Leftarrow$) Consider a valid subtree of $G_{T, \mathbf{b}}$, resulting in a tree T' and isotype labeling β' . To see why T' is a refinement of T , observe that edges $((u, s), (u, t))$ correspond to an EXPAND operation on node u of T . It remains to show that β' is transitory. By condition (i) of Definition E.3, we have that the root of T' is labeled by state 1, satisfying condition (i) of Definition E.1. Conditions (ii) and (iii) of Definition E.1 are met by construction of $G_{T, \mathbf{b}}$. Finally, condition (iv) of Definition E.1 follows from condition (ii) of Definition E.3. Hence, the isotype labeling β' of T' is transitory. QED.

The following key proposition follows from the previous two lemmas.

Proposition E.1. Let $G_{T, \mathbf{b}}$ be an expansion graph of a rooted tree T whose leaves are labeled by isotypes $\mathbf{b}$. Then, given isotype transition probabilities $\mathbf{P}$, a valid subtree (T', β') of $G_{T, \mathbf{b}}$ maximizing $\sum_{(u', v') \in E(T')} \log p_{\beta'(u'), \beta'(v')}$ is an optimal solution to MPTR instance $(T, \mathbf{b}, \mathbf{P})$.

To find such a valid subtree with maximum log-likelihood, we formulate the following MILP based on a multi-commodity flow formulation for modeling connectivity. We make use of two sets of decision variables. The first is $f_{(u,s),(v,t)}^t \in \mathbb{R}_{\geq 0}$, which represents the amount of flow on edge (u, v) designated for sink $q \in L(T)$. The second is $x_{(u,s),(v,t)} \in \{0, 1\}$, which indicates if edge (u, v) has non-zero flow.

$$\min \sum_{((u,s),(v,t)) \in E(G_{T,\mathbf{b}})} x_{(u,s),(v,t)} \log p_{s,t} \quad (\text{E.10})$$

s.t.

$$\sum_{(v,t) \in \eta^+((u,s))} f_{(u,s),(v,t)}^q = \sum_{(v,t) \in \eta^-((u,s))} f_{(v,t),(u,s)}^q, \quad \forall v \in V(T) \setminus L(T),$$

$$(v, s) \in V(G_{T,\mathbf{b}}) \setminus \{(v_0, 1)\}, q \in L(T), \quad (\text{E.11})$$

$$\sum_{(u,s) \in \eta^-((q,b_q))} f_{(u,s),(q,b_q)}^q = 1, \quad \forall q \in L(T), \quad (\text{E.12})$$

$$\sum_{(v,t) \in \eta^+((v_0,1))} f_{(v_0,1),(v,t)}^q = 1, \quad \forall q \in L(T), \quad (\text{E.13})$$

$$f_{(u,s),(v,t)}^q \leq x_{(u,s),(v,t)}, \quad \forall q \in L(T), ((u,s),(v,t)) \in E(G_{T,\mathbf{b}}), \quad (\text{E.14})$$

$$\sum_{(u,s) \in X(u)} \sum_{(v,t) \in X(v)} x_{(u,s),(v,t)} = 1, \quad \forall (u,v) \in E(T), \quad (\text{E.15})$$

$$0 \leq f_{(u,s),(v,t)}^q \leq 1, \quad \forall q \in L(T), ((u,s),(v,t)) \in E(G_{T,\mathbf{b}}), \quad (\text{E.16})$$

$$x_{(u,s),(v,t)} \in \{0, 1\}, \quad \forall ((u,s),(v,t)) \in E(G_{T,\mathbf{b}}), \quad (\text{E.17})$$

where $\eta^+((u,s))$ is the set of direct successors of node (u,s) in graph $E(G_{T,\mathbf{b}})$ and $\eta^-((u,s))$ is the set of direct predecessors of node (u,s) .

Constraints (E.11), (E.12), (E.13) enforce flow conservation and ensure that each terminal receives one unit of flow. Below is a description of each of the above constraints. Constraint (E.14) links the flow variables to the choice of edges in the resulting refinement. Finally, constraint (E.15) ensures that refined tree T' can be obtained from tree T via a series of EXPAND operations.

E.3.2 Maximum likelihood estimate of isotype transition probabilities

Given a forest $T_1, \dots, T_k$ of lineage trees correspondingly labeled by isotypes $\beta_1, \dots, \beta_k$, we seek the maximum likelihood estimate of isotype transition probabilities $\mathbf{P}^*$.

$$\mathbf{P}^* = \underset{\mathbf{P}}{\operatorname{argmax}} \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{\sum_{j=1}^k N_{j,s,t}} \quad (\text{E.18})$$

subject to

$$\sum_{t \in [r]} p_{s,t} = 1, \quad \forall s \in [r]. \quad (\text{E.19})$$

We solve this constrained optimization problem using Lagrange multipliers λ_s for each state s . We first take the log of likelihood $\prod_{j=1}^k \text{CSR}(T_j, \beta_j, \mathbf{P})$ with respect to isotype transition probabilities $\mathbf{P}$.

$$\log \prod_{j=1}^k \text{CSR}(T_j, \beta_j, \mathbf{P}) = \log \prod_{(s,t) \in [r] \times [r]} p_{s,t}^{\sum_{j=1}^k N_{j,s,t}} \quad (\text{E.20})$$

$$= \sum_{(s,t) \in [r] \times [r]} \left(\sum_{j=1}^k N_{j,s,t} \right) \log p_{s,t}. \quad (\text{E.21})$$

To our log-likelihood, we add the term $\lambda_s \left(\sum_{s \in [r]} p_{s,t} - 1 \right)$ for each isotype s , resulting in new objective

$$\mathcal{L}(\mathbf{P}, \lambda_1, \dots, \lambda_r) = \left[\sum_{(s,t) \in [r] \times [r]} \left(\sum_{j=1}^k N_{j,s,t} \right) \log p_{s,t} + \sum_{s \in [r]} \lambda_s \left(\sum_{t \in [r]} p_{s,t} - 1 \right) \right] \quad (\text{E.22})$$

Then, we set the partial derivative of $\mathcal{L}(\mathbf{P}, \lambda_1, \dots, \lambda_r)$ with respect to each parameter $p_{s,t}$ and λ_s and solve the resulting system of equations. For each λ_s , we obtain our constraint,

$$\frac{\partial \mathcal{L}}{\partial \lambda_s} = 0 = \left(\sum_{t \in [r]} p_{s,t} - 1 \right) \quad (\text{E.23})$$

$$\sum_{t \in [r]} p_{s,t} = 1 \quad (\text{E.24})$$

For each parameter $p_{s,t}$, we set the partial derivative to 0 and solve for $p_{s,t}$ as a function of λ_s .

$$\frac{\partial \mathcal{L}}{\partial p_{s,t}} = 0 = \frac{\sum_{j=1}^k N_{j,s,t}}{p_{s,t}} - \lambda_s \quad (\text{E.25})$$

$$\lambda_s = \frac{\sum_{j=1}^k N_{j,s,t}}{p_{s,t}} \quad (\text{E.26})$$

$$p_{s,t} = \frac{\sum_{j=1}^k N_{j,s,t}}{\lambda_s} \quad (\text{E.27})$$

Given the constraint (E.23), we have that

$$\sum_{t \in [r]} p_{s,t} = \frac{\sum_{t \in [r]} \sum_{j=1}^k N_{j,s,t}}{\lambda_s} = 1, \quad (\text{E.28})$$

and

$$\lambda_s = \sum_{t \in [r]} \sum_{j=1}^k N_{j,s,t}. \quad (\text{E.29})$$

This yields the following maximum likelihood estimate $p_{s,t}^*$,

$$p_{s,t}^* = \frac{\sum_{j=1}^k N_{j,s,t}}{\sum_{t \in [r]} \sum_{j=1}^k N_{j,s,t}} \quad (\text{E.30})$$

Lastly, to account for unobserved isotype transitions where isotype $s \leq t$, we add a pseudocount of 1, resulting in updated isotype transition probabilities

$$p_{s,t}^* = \frac{\sum_{j=1}^k N_{j,s,t} + 1}{\sum_{t \in [r]} \left(\sum_{j=1}^k N_{j,s,t} + 1 \right)}. \quad (\text{E.31})$$

In the main text, we additionally use the shorthand $N_{s,t} = \sum_{j=1}^k N_{j,s,t}$.

E.4 SIMULATION DETAILS

We designed *in silico* experiments to evaluate TRIBAL with known ground-truth isotype transition probabilities $\mathbf{P}$ and lineage trees T labeled by sequences α and isotypes β . Specifically, we used an existing BCR phylogenetic simulator [64] that models SHM (Appendix E.4.1 but not CSR. We generated isotype transition probabilities $\mathbf{P}$ with $r = 7$ isotypes (as in mice) under two different models of CSR (Appendix E.4.2). Briefly, both CSR models assume the probability of not transitioning is higher than the probability of transitioning, but in the *sequential model* there is clear preference for transitions to the next contiguous isotype, while in the *direct model* the probabilities of contiguous and non-contiguous class are similar (Fig. E.2) Given $\mathbf{P}$, we evolved isotype characters down each ground truth lineage tree T .

We generated 5 replications of each CSR model for $k = 75$ clonotypes and $n \in \{35, 65\}$ cells per clonotype, resulting in 20 *in silico* experiments, yielding a total of 1500 ground truth lineage trees. We generated our *in silico* experiments to evaluate all aspects of TRIBAL while benchmarking against existing methods including dnapars [5], dnaml [5] and IgPhyML [61].

E.4.1 SHM simulation and benchmarking

The Davidsen and Matsen SHM simulator models the generation of B cell lineage trees via a Poisson branching process with selection towards BCRs with increased affinity [64]. We used the provided Docker Hub image container ¹⁵ to generate our ground truth B cell lineage trees T and sequence labels α . In addition, we used the provided benchmarking pipeline to run dnapars [5], dnaml [5] and IgPhyML [61]. Below is the command to generate our *in silico* experiments for $n \in \{35, 65\}$ cells and $k = 75$ clonotypes and run comparison methods.

```
simulate
  --igphym1
  --dnapars
  --dnaml
  --selection
  --target_dist=5
  --target_count=100
  --carry_cap=1000
```

¹⁵krdav/bcr-phylo-benchmark

```

--T=35
--lambda=2.0
--lambda0=0.365
--n={n}
--nsim={k}
--random_naive=sequence_data/AbPair_naive_seqs.fa

```

E.4.2 CSR simulation

After generating each ground truth B cell lineage tree T as described above, we then evolved isotype characters down each tree T using two different models for class switch recombination to obtain ground truth isotypes β . First, we describe the two different CSR models that we used to generate ground truth isotype transition probabilities $\mathbf{P}$. Then, we describe the generation of these isotype transition probability matrices under these two models.

We grouped each isotype transition probability $p_{s,t}$ where $s \leq t$ into one of three categories: (i) *stay*, (ii) *next*, and (iii) *jump* (Fig E.2a). In *stay*, the B cell does not undergo any class switching and the isotype does not change. In *next*, a B cell class switches to the next contiguous heavy chain locus. In *jump*, the B cell class switches by jumping to an isotype heavy chain constant locus that is not contiguous.

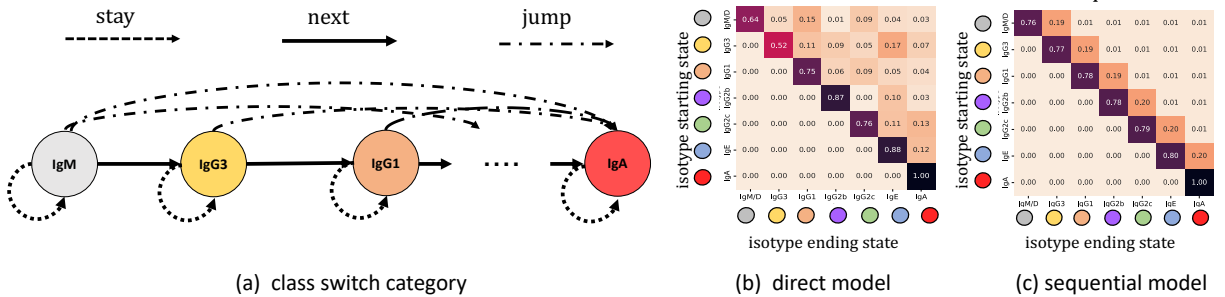


Figure E.2: **Class switch recombination models for *in silico* experiments.** a) Examples of different isotype transition probability parameter groups. (b) Examples of simulated isotype transition probabilities $\mathbf{P}$ for the direct model of CSR. In the direct model, when a B cell class switches is no systematic preference for transition to the *next* sequential state or *jumping* to a non-contiguous isotype. (c) In the sequential model, a B cell undergoing CSR has a strong affinity for the *next* contiguous heavy chain locus.

Next, we describe how we generated ground truth isotype transition probabilities $\mathbf{P}$ under both direct and sequential CSR models. To simulate isotype transition probabil-

ities with direct switching, we randomly sampled a probability of transitioning $1 - \theta \in \{0.1, 0.15, \dots, 0.35\}$. We then set the initial isotype transition probabilities as

$$p'_{s,t} = \begin{cases} 0, & \text{if } s > t, \\ \min(\theta + \epsilon, \tau), & \text{if } s = t, \\ \min(\frac{1-\theta}{r-s} + \epsilon, \tau), & \text{if } s < t, \end{cases} \quad (\text{E.32})$$

where we add Gaussian noise $\epsilon \sim \mathcal{N}(\mu, \sigma)$ with mean $\mu = 0.05$ and standard deviation $\sigma = 0.025$ to each parameter. To avoid negative transition probabilities we set $\tau = 0.01$. Fig. E.2b shows an example of a simulated isotype transition probability matrix under the direct CSR model.

$$p'_{s,t} = \begin{cases} 0, & \text{if } s > t, \\ \min(\theta + \epsilon, \tau), & \text{if } s = t, \\ \min(1 - \theta + \epsilon, \tau), & \text{if } t = s + 1, \\ \tau, & \text{otherwise.} \end{cases} \quad (\text{E.33})$$

We then set parameter $p'_{s,t} := p'_{s,t} / \sum_{s \in [r]} p'_{s,t}$ to ensure each row in the isotype transition probability matrix $\mathbf{P}$ sums to 1. Fig. E.2b shows an example of a simulated isotype transition probability matrix under the direct model. Fig. E.2c shows an example of a simulated isotype transition probability matrix under the sequential CSR model.

E.4.3 Inference using TRIBAL

We ran TRIBAL in two ways, referred to as TRIBAL and TRIBAL-NO REFINEMENT, in order to assess the importance of the tree refinement stage of our algorithm. As the naming convention implies, the main difference between TRIBAL and TRIBAL-NO REFINEMENT, is that in TRIBAL-NO REFINEMENT the input trees are not refined and the isotypes $\hat{\beta}$ are inferred using the Sankoff [188] algorithm with weights $w_{s,t}^{(\ell)} = -\log p_{s,t}^{(\ell)}$. All other steps of TRIBAL algorithm remain the same.

Due to large input sets $\mathcal{T}_j$ for some simulated clonotypes j , we sample 50 trees from $\mathcal{T}_j$ for consideration of candidate lineage tree $T_j^{(\ell)}$ within each iteration ℓ . We additionally include the previous optimal lineage tree $T^{(\ell-1)}$ of iteration $\ell - 1$ in the sampled trees for each clonotype j to ensure convergence.

We used a convergence threshold of 0.5 and a maximum of 10 iterations per restart. A total of 5 restarts were performed by iterating through $\theta \in \{0.55, 0.65, 0.75, 0.85, 0.95\}$ for each restart.

E.4.4 Performance metrics

To evaluate performance of isotype transition probability inference, $\hat{\mathbf{P}}$, we utilized *Kullback-Leibler (KL) divergence*. To assess accuracy of lineage tree inference, we used normalized Robinson-Foulds (RF) distance to assess accuracy of the topology of the inferred tree $\hat{T}$, most recent common ancestor (MRCA) distance to assess accuracy of the inferred sequences $\hat{\alpha}$, and Class Switch Recombination (CSR) error to assess accuracy of the inferred isotypes $\hat{\beta}$.

Kullback-Leibler (KL) divergence. To evaluate accuracy of isotype transition probability inference, we used *Kullback-Leibler (KL) divergence* [186] to compare the inferred transition probability distribution $\hat{\mathbf{p}}_s$ of each isotype s to the simulated ground truth distribution $\mathbf{p}_s$. KL divergence D_{KL} is defined as

$$D_{\text{KL}}(\hat{\mathbf{p}}_s || \mathbf{p}_s) = \sum_{q \in [r]} \hat{p}_{s,t} \log(\hat{p}_{s,t}/p_{s,t}) \quad (\text{E.34})$$

The lower the KL divergence, the more similar the two distributions.

Normalized Robinson-Foulds (RF) distance. To assess the accuracy of topology of the inferred B cell lineage tree $\hat{T}$ with respect to simulated ground truth tree T , we used *normalized Robinson-Foulds (RF) distance*. For this metric, we treat both trees as unrooted. For an unrooted tree, if you remove an edge (but not its endpoints), it defines a bipartition of the leaf set [2]. Doing this for every edge in tree T yields a set $B(T)$ of bipartitions. RF distance is defined as the size of the symmetric difference between bipartitions $B(T)$ and $B(\hat{T})$ [187]. We then normalize this by the total number of bipartitions in each tree. Thus, normalized RF is computed as follows

$$\text{normalized RF}(T, \hat{T}) = \frac{|B(T) \Delta B(\hat{T})|}{|B(T)| + |B(\hat{T})|}. \quad (\text{E.35})$$

Most Recent Common Ancestor (MRCA) distance. To assess the accuracy of the inferred ancestral sequence reconstruction $\hat{\alpha}$ with respect to simulated ground truth α ,

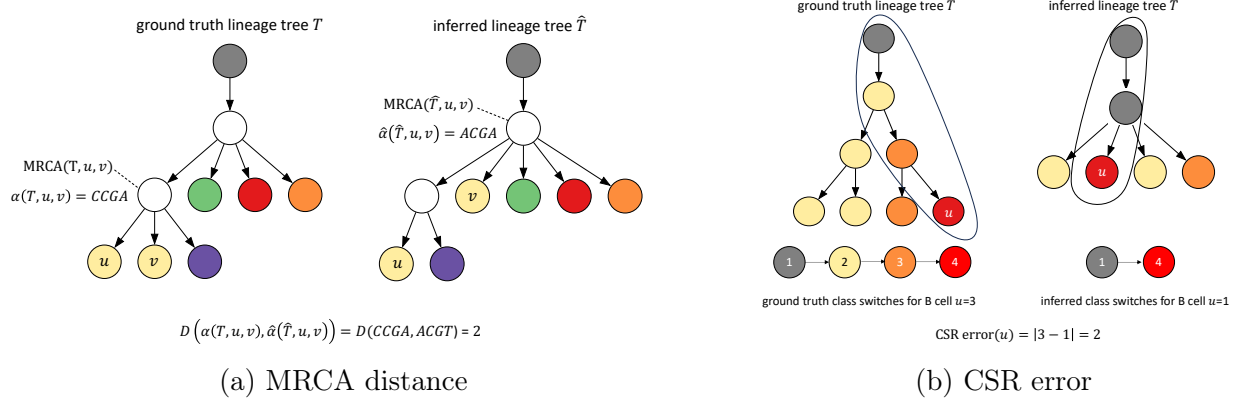


Figure E.3: **Performance metrics for B cell lineage tree inference.** (a) An example calculation for MRCA distance leaves u and v . (b) An example calculation of CSR error for lineage u .

we used a metric called *Most Recent Common Ancestor (MRCA) distance* introduced by Davidsen and Matsen [64]. For any two simulated B cells (leaves), the MRCA distance is the Hamming distance between the MRCA sequences of these two B cells in both the ground truth and inferred lineage trees. This distance is then averaged over all pairs of simulated B cells. A graphical depiction of this metric is shown in Fig. E.3a.

More formally,

$$\text{MRCA distance}(\alpha, \hat{\alpha}) = \frac{2}{n(n-1)m} \sum_{u,v \in L(T)} D(\hat{\alpha}(\hat{T}, u, v), \alpha(T, u, v)), \quad (\text{E.36})$$

where in a slight abuse of notation $\alpha(T, u, v)$ is the sequence of the most recent common ancestor (MRCA) of nodes u and v in lineage tree T and m is the length of MSA.

Class switch recombination (CSR) error. We assessed the accuracy of isotype inference by a new metric called *CSR error*, which is computed for each B cell i and clonotype j and is the absolute difference between the number of ground-truth class switches and inferred number of class switches that occurred along its evolutionary path from the root (Fig. E.3b). Since dnaml, dnapars and IgPhyML do not infer isotypes for internal nodes, we pair these methods with the Sankoff algorithm [188] using $w_{s,t}$ equals 1 if $s = t$, 0 if $s < t$ and ∞ otherwise.

E.4.5 Average clade entropy for a leaf labeling

We describe a metric used to assess the average entropy contained within a leaf-labeling of the clades of a tree. First, we introduce some notation. Let Σ be an alphabet. Let clade u of tree T be the subtree T_u rooted at node u . Let $\delta(u) \subseteq L(T)$ be the subset of leaves that are descendants of node u . Let $\ell : L(T) \rightarrow \Sigma$ be a leaf labeling. Given a clade u and leaf-labeling ℓ , the entropy of a clade with respect to its leaf labels is defined as

$$H(u, \ell) = - \sum_{s \in [\Sigma]} p(s) \log p(s), \quad (\text{E.37})$$

where $p(s) = \sum_{v \in \delta(u)} \mathbf{1}(\ell(v) = s) / |\delta(u)|$. The average clade entropy $\bar{H}$ is computed over all clades except the leaves $L(T)$ and the root r as follows

$$\bar{H}(T, \ell) = \frac{\sum_{u \in \bar{V}} H(u, \ell)}{|\bar{V}|}, \quad (\text{E.38})$$

where $\bar{V} = V(T) \setminus (\{r\} \cup L(T))$ is the set of non-trivial clades.

E.5 SUPPLEMENTARY RESULTS

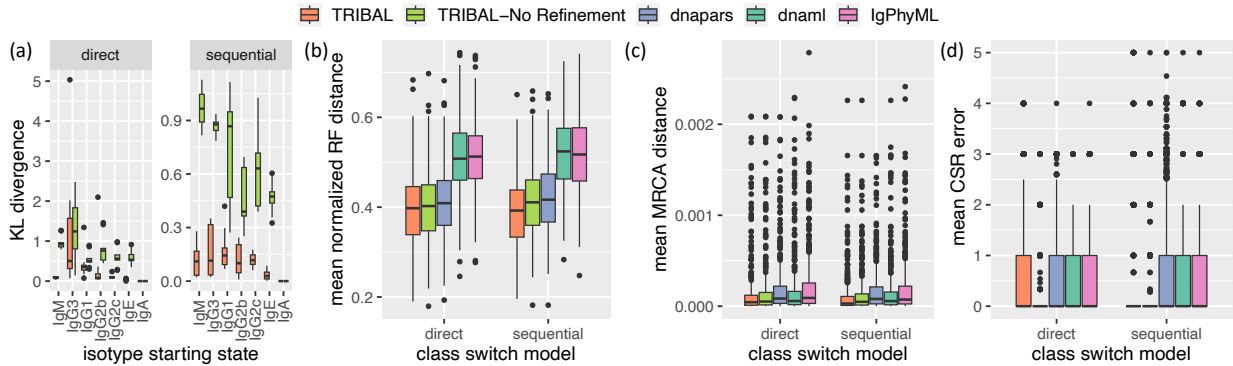


Figure E.4: **Simulations results for $k = 75$ clonotypes and $n = 65$ cells per clonotype.**

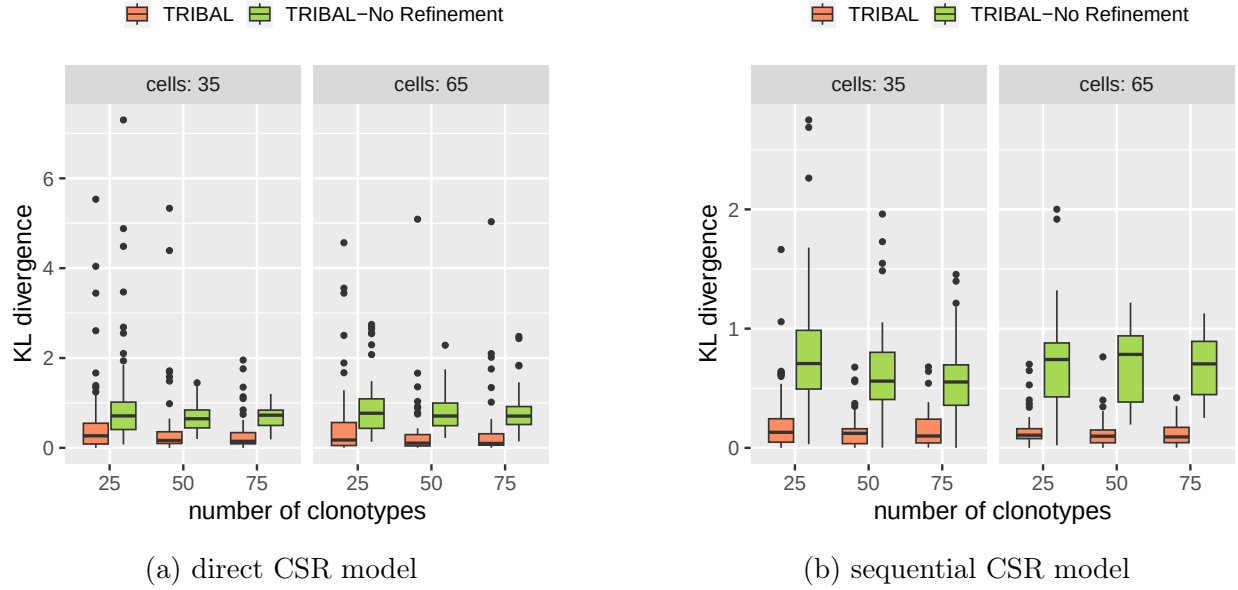


Figure E.5: KL divergence from ground truth isotype transition probabilities aggregated over all isotype starting states, except IgA, by isotype starting state with varying the number k clonotypes, the number n of cells and CSR model.

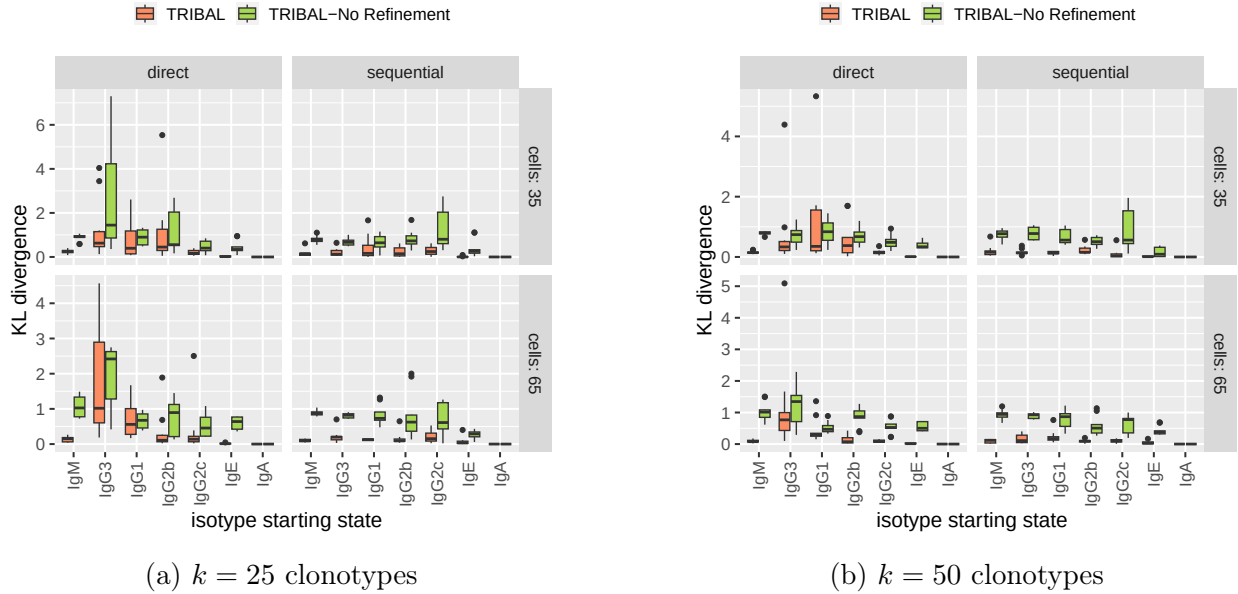


Figure E.6: KL divergence from ground truth isotype transition probabilities by isotype starting state for $k \in \{25, 50\}$ clonotypes

dataset	clonotypes k	total cells n	median cells per clonotype	max cells per clonotype	median distinct isotypes per clonotype
NP-KLH-1	167	1776	7	89	3
NP-KLH-2a	70	537	6	32	2
NP-KLH-2b	58	357	5	21	2

Table E.1: Summary of NP-KLH mouse scRNA-seq datasets

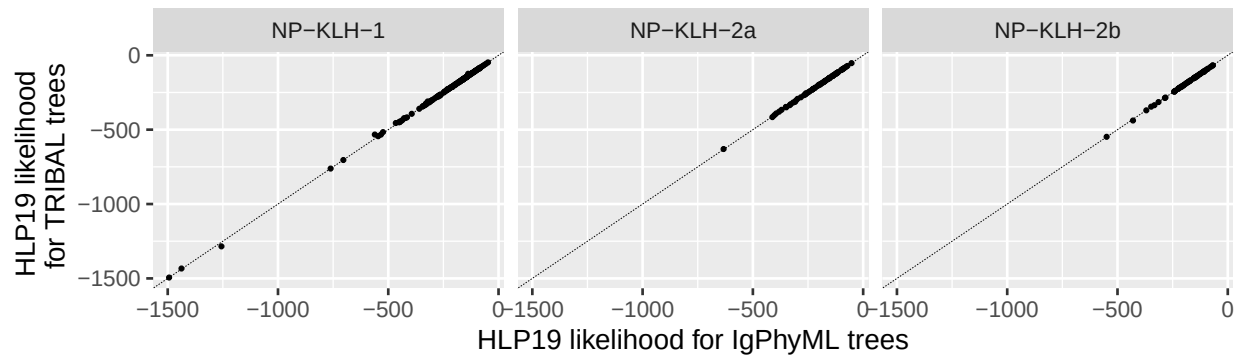


Figure E.7: Comparison of HLP19 likelihood computed for IgPhyML and TRIBAL inferred B cell lineage trees for NP-KLH datasets.

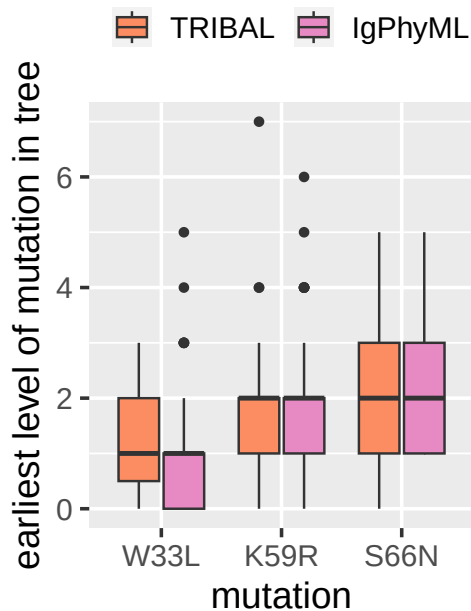


Figure E.8: Earliest observed level of mutation in a B cell lineage tree. Level 0 represents the MRCA.

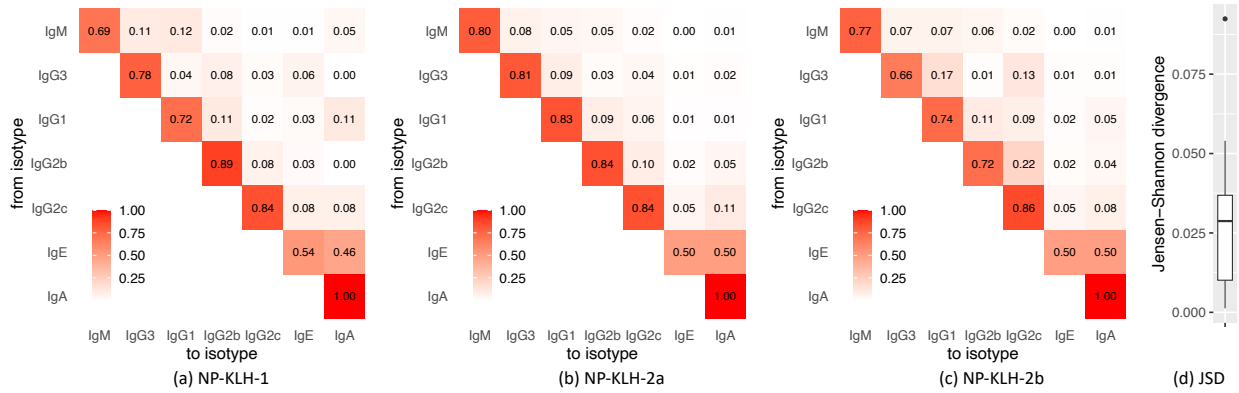


Figure E.9: **TRIBAL** inferred isotype transition probabilities for NP-KLH. (a) Isotype transition probabilities for NP-KLH-1. (b) Isotype transition probabilities for NP-KLH-2a. (c) Isotype transition probabilities for NP-KLH-2b. (d) The distribution of Jensen-Shannon divergence (JSD) for pairwise comparisons of rows of the inferred isotype transition probability matrices for IgM through Ig2c. IgE was excluded from comparison due to a lack of observed B cells within each dataset to yield informative estimates. IgA is excluded as the inference of this row is trivial.

dataset	clonotypes k	total cells n	median cells per clonotype	max cells per clonotype	median distinct isotypes per clonotype
mouse 1	24	224	7.5	31	2
mouse 2	15	218	7	81	3
mouse 3	15	157	7	39	3

Table E.2: **Summary of ABC mouse scRNA-seq datasets**

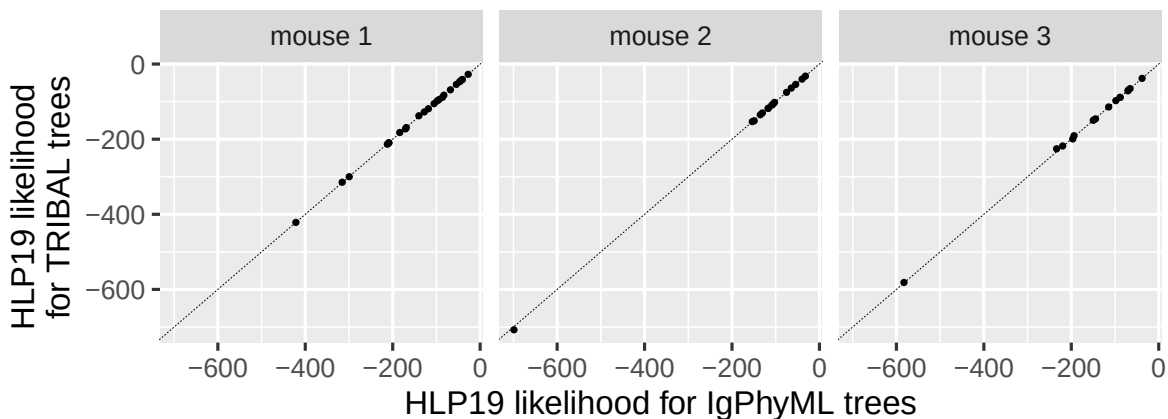


Figure E.10: **Scatterplot** comparing HLP19 likelihood for IgPhyML trees to the HLP19 likelihood computed for TRIBAL trees for ABC datasets.

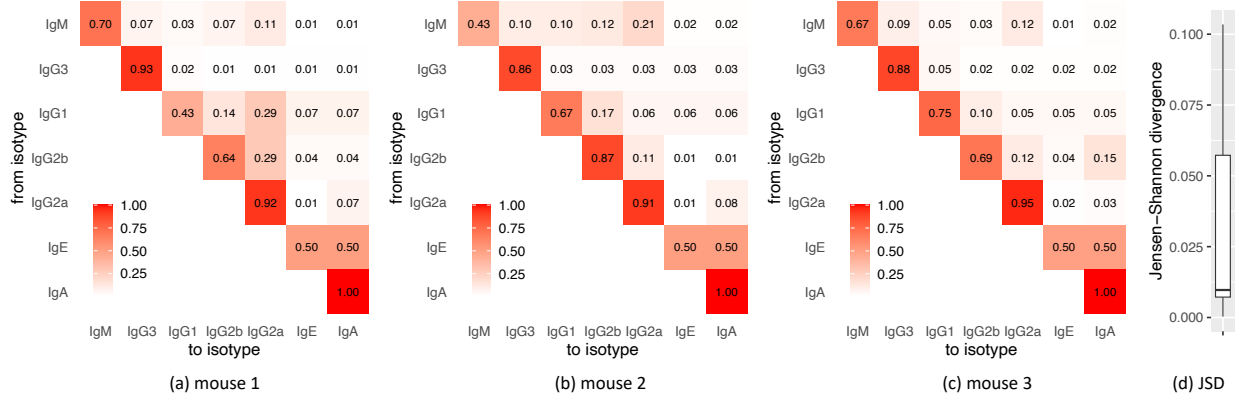


Figure E.11: **TRIBAL** inferred isotype transition probabilities for ABC datasets. (a) Isotype transition probabilities for Mouse 1. (b) Isotype transition probabilities for Mouse 2. (c) Isotype transition probabilities for NP-Mouse 3. (d) The distribution of Jensen-Shannon divergence (JSD) for pairwise comparisons of rows of the inferred isotype transition probability matrices for IgM through Ig2c. IgE was excluded from comparison due to a lack of observed B cells within each dataset to yield informative estimates. IgA is excluded as the inference of this row is trivial.

REFERENCES

- [1] D. M. Hillis, C. Moritz, and B. K. Mable, *Molecular Systematics*. Sinauer Associates, 1996.
- [2] T. Warnow, *Computational Phylogenetics: an Introduction to Designing Methods for Phylogeny Estimation*. Cambridge University Press, 2017.
- [3] C. Semple and M. Steel, *Phylogenetics*. Oxford University Press, 2003.
- [4] J. Felsenstein, “A likelihood approach to character weighting and what it tells us about parsimony and compatibility,” *Biological Journal of the Linnean Society*, vol. 16, no. 3, pp. 183–196, 1981.
- [5] J. Felsenstein, “PHYLP (phylogeny inference package) version 3.6. distributed by the author,” <http://www.evolution.gs.washington.edu/phylip.html>, 2004.
- [6] J. Hadfield, C. Megill, S. M. Bell, J. Huddleston, B. Potter, C. Callender, P. Sagulenko, T. Bedford, and R. A. Neher, “Nextstrain: real-time tracking of pathogen evolution,” *Bioinformatics*, vol. 34, no. 23, pp. 4121–4123, 2018.
- [7] T. Li, D. Liu, Y. Yang, J. Guo, Y. Feng, X. Zhang, S. Cheng, and J. Feng, “Phylogenetic supertree reveals detailed evolution of SARS-CoV-2,” *Scientific Reports*, vol. 10, no. 1, p. 22366, 2020.
- [8] P. Forster, L. Forster, C. Renfrew, and M. Forster, “Phylogenetic network analysis of SARS-CoV-2 genomes,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 17, pp. 9241–9243, 2020.
- [9] S. W. Attwood, S. C. Hill, D. M. Aanensen, T. R. Connor, and O. G. Pybus, “Phylogenetic and phylodynamic approaches to understanding and combating the early SARS-CoV-2 pandemic,” *Nature Reviews Genetics*, vol. 23, no. 9, pp. 547–562, 2022.
- [10] N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito et al., “Tumour evolution inferred by single-cell sequencing,” *Nature*, vol. 472, no. 7341, pp. 90–94, 2011.
- [11] M. Pellegrino et al., “High-throughput single-cell DNA sequencing of acute myeloid leukemia tumors with droplet microfluidics,” *Genome Research*, vol. 28, no. 9, pp. 1345–1352, 2018.
- [12] M. L. Leung, A. Davis, R. Gao, A. Casasent, Y. Wang, E. Sei, E. Vilar, D. Maru, S. Kopetz, and N. E. Navin, “Single-cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer,” *Genome Research*, vol. 27, no. 8, pp. 1287–1299, 2017.

- [13] Y. Wang and N. E. Navin, “Advances and applications of single-cell sequencing technologies,” *Molecular cell*, vol. 58, no. 4, pp. 598–609, 2015.
- [14] D. E. Wagner and A. M. Klein, “Lineage tracing meets single-cell omics: opportunities and challenges,” *Nature Reviews Genetics*, vol. 21, no. 7, pp. 410–427, 2020.
- [15] M. B. Woodworth, K. M. Girsakis, and C. A. Walsh, “Building a lineage from single cells: genetic techniques for cell lineage tracking,” *Nature Reviews Genetics*, vol. 18, no. 4, pp. 230–244, 2017.
- [16] J. Kuipers et al., “Advances in understanding tumour evolution through single-cell sequencing,” *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, vol. 1867, no. 2, pp. 127–138, 2017.
- [17] K. E. Neu, Q. Tang, P. C. Wilson, and A. A. Khan, “Single-cell genomics: approaches and utility in immunology,” *Trends in Immunology*, vol. 38, no. 2, pp. 140–149, 2017.
- [18] P. C. Nowell, “The clonal evolution of tumor cell populations: Acquired genetic lability permits stepwise selection of variant sublines and underlies tumor progression.” *Science*, vol. 194, no. 4260, pp. 23–28, 1976.
- [19] G. Stanta and S. Bonin, “Overview on clinical relevance of intra-tumor heterogeneity,” *Frontiers in Medicine*, vol. 5, p. 85, 2018.
- [20] A. Marusyk and K. Polyak, “Tumor heterogeneity: causes and consequences,” *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, vol. 1805, no. 1, pp. 105–117, 2010.
- [21] A. Marusyk, M. Janiszewska, and K. Polyak, “Intratumor heterogeneity: the rosetta stone of therapy resistance,” *Cancer cell*, vol. 37, no. 4, pp. 471–484, 2020.
- [22] S. Venkatesan and C. Swanton, “Tumor evolutionary principles: how intratumor heterogeneity influences cancer treatment and outcome,” *American Society of Clinical Oncology Educational Book*, vol. 36, pp. e141–e149, 2016.
- [23] D. C. Minussi et al., “Breast tumours maintain a reservoir of subclonal diversity during expansion,” *Nature*, vol. 592, no. 7853, pp. 302–308, 2021.
- [24] E. Laks et al., “Clonal decomposition and DNA replication states defined by scaled single-cell genome sequencing,” *Cell*, vol. 179, no. 5, pp. 1207–1221, 2019.
- [25] H. Zahn et al., “Scalable whole-genome single-cell library preparation without preamplification,” *Nature Methods*, vol. 14, no. 2, p. 167, 2017.
- [26] T. Funnell et al., “Single-cell genomic variation induced by mutational processes in cancer,” *Nature*, vol. 612, no. 7938, pp. 106–115, 2022.
- [27] S. Ivanovic and M. El-Kebir, “Modeling and predicting cancer clonal evolution with reinforcement learning,” *Genome Research*, vol. 33, no. 7, pp. 1078–1088, 2023.

- [28] S. Christensen, M. D. Leiserson, and M. El-Kebir, “PhySigs: phylogenetic inference of mutational signature dynamics,” in *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2020*. World Scientific, 2019, pp. 226–237.
- [29] M. Gerstung et al., “The evolutionary history of 2,658 cancers,” *Nature*, vol. 578, no. 7793, pp. 122–128, 2020.
- [30] K. Morita et al., “Clonal evolution of acute myeloid leukemia revealed by high-throughput single-cell genomics,” *Nature Communications*, vol. 11, no. 1, pp. 1–17, 2020.
- [31] Y. Rubanova, R. Shi, C. F. Harrigan, R. Li, J. Wintersinger, N. Sahin, A. G. Deshwar, and Q. D. Morris, “Reconstructing evolutionary trajectories of mutation signature activities in cancer using TrackSig,” *Nature Communications*, vol. 11, no. 1, p. 731, 2020.
- [32] N. Amirouchene-Angelozzi, C. Swanton, and A. Bardelli, “Tumor evolution as a therapeutic target,” *Cancer Discovery*, vol. 7, no. 8, pp. 805–817, 2017.
- [33] C. Abbosh et al., “Phylogenetic ctDNA analysis depicts early-stage lung cancer evolution,” *Nature*, vol. 545, no. 7655, pp. 446–451, 2017.
- [34] 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, July 2016.
- [35] A. G. Deshwar, S. Vembu, C. K. Yung, G. H. Jang, L. Stein, and Q. Morris, “PhyloWGS: Reconstructing subclonal composition and evolution from whole-genome sequencing of tumors,” *Genome Biology*, vol. 16, no. 1, p. 35, Feb. 2015.
- [36] S. Malikic, A. W. McPherson, N. Donmez, and C. S. Sahinalp, “Clonality inference in multiple tumor samples using phylogeny,” *Bioinformatics*, vol. 31, no. 9, pp. 1349–1356, 2015.
- [37] Y. Qi, D. Pradhan, and M. El-Kebir, “Implications of non-uniqueness in phylogenetic deconvolution of bulk DNA samples of tumors,” *Algorithms for Molecular Biology*, vol. 14, no. 1, pp. 23–14, Sep. 2019.
- [38] K. Jahn, J. Kuipers, and N. Beerenwinkel, “Tree inference for single-cell data,” *Genome Biology*, vol. 17, no. 1, p. 86, 2016.
- [39] S. Malikic et al., “PhISCS: a combinatorial approach for subperfect tumor phylogeny reconstruction via integrative use of single-cell and bulk sequencing data,” *Genome Research*, vol. 29, no. 11, pp. 1860–1877, Oct. 2019.
- [40] A. Roth et al., “Clonal genotype and population structure inference from single-cell tumor sequencing,” *Nature Methods*, vol. 13, no. 7, pp. 573–576, 2016.

- [41] E. M. Ross and F. Markowetz, “OncoNEM: inferring tumor evolution from single-cell sequencing data,” *Genome Biology*, vol. 17, no. 1, pp. 1–14, 2016.
- [42] M. El-Kebir, “SPHyR: tumor phylogeny estimation from single-cell sequencing data under loss and error,” *Bioinformatics*, vol. 34, no. 17, pp. i671–i679, 2018.
- [43] H. Zafar et al., “SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data,” *Genome Research*, vol. 29, no. 11, pp. 1847–1859, 2019.
- [44] F. Wang et al., “MEDALT: single-cell copy number lineage tracing enabling gene discovery,” *Genome Biology*, vol. 22, no. 1, pp. 1–22, 2021.
- [45] M. Markowska, T. Cakała, B. Miasojedow, B. Aybey, D. Juraeva, J. Mazur, E. Ross, E. Staub, and E. Szczurek, “CONET: Copy number event tree model of evolutionary tumor history for single-cell data,” *Genome Biology*, vol. 23, no. 1, pp. 1–35, 2022.
- [46] T. L. Kaufmann et al., “MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution,” *Genome Biology*, vol. 23, no. 1, p. 241, 2022.
- [47] K. T. Schmid, B. Höllbacher, C. Cruceanu, A. Böttcher, H. Lickert, E. B. Binder, F. J. Theis, and M. Heinig, “scPower accelerates and optimizes the design of multi-sample single cell transcriptomic studies,” *Nature Communications*, vol. 12, no. 1, p. 6625, 2021.
- [48] V. Svensson, K. N. Natarajan, L.-H. Ly, R. J. Miragaia, C. Labalette, I. C. Macaulay, A. Cvejic, and S. A. Teichmann, “Power analysis of single-cell RNA-sequencing experiments,” *Nature Methods*, vol. 14, no. 4, pp. 381–387, 2017.
- [49] D. Abrams, P. Kumar, R. Karuturi, and J. George, “A computational method to aid the design and analysis of single cell RNA-seq experiments for cell type identification,” *BMC Bioinformatics*, vol. 20, no. 11, pp. 1–6, 2019.
- [50] A. Davis, R. Gao, and N. E. Navin, “SCOPIT: sample size calculations for single-cell sequencing experiments,” *BMC Bioinformatics*, vol. 20, no. 1, p. 566, 2019.
- [51] S. L. Wolock et al., “Scrublet: computational identification of cell doublets in single-cell transcriptomic data,” *Cell Systems*, vol. 8, no. 4, pp. 281–291, 2019.
- [52] C. S. McGinnis et al., “DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors,” *Cell Systems*, vol. 8, no. 4, pp. 329–337, 2019.
- [53] A. Davis, R. Gao, and N. Navin, “Tumor evolution: Linear, branching, neutral or punctuated?” *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, vol. 1867, no. 2, pp. 151–161, 2017.
- [54] K. Murphy and C. Weaver, *Janeway’s Immunobiology*. Garland science, 2016.

- [55] E. Meffre, R. Casellas, and M. C. Nussenzweig, “Antibody regulation of B cell development,” *Nature Immunology*, vol. 1, no. 5, pp. 379–385, 2000.
- [56] G. D. Victora and M. C. Nussenzweig, “Germinal centers,” *Annu Rev Immunol*, vol. 30, no. 1, pp. 429–457, 2012.
- [57] J. Stavnezer, J. E. Guikema, and C. E. Schrader, “Mechanism and regulation of class switch recombination,” *Annual Review of Immunology*, vol. 26, p. 261, 2008.
- [58] A. J. Drummond and A. Rambaut, “BEAST: Bayesian evolutionary analysis by sampling trees,” *BMC Evolutionary Biology*, vol. 7, no. 1, pp. 1–8, 2007.
- [59] 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.
- [60] 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.
- [61] K. B. Hoehn, G. Lunter, and O. G. Pybus, “A phylogenetic codon substitution model for antibody lineages,” *Genetics*, vol. 206, no. 1, pp. 417–427, 2017.
- [62] K. B. Hoehn, J. A. Vander Heiden, J. Q. Zhou, G. Lunter, O. G. Pybus, and S. H. Kleinstein, “Repertoire-wide phylogenetic models of B cell molecular evolution reveal evolutionary signatures of aging and vaccination,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 664–22 672, 2019.
- [63] W. S. DeWitt III, L. Mesin, G. D. Victora, V. N. Minin, and F. A. Matsen IV, “Using genotype abundance to improve phylogenetic inference,” *Molecular Biology and Evolution*, vol. 35, no. 5, pp. 1253–1265, 2018.
- [64] K. Davidsen and F. A. Matsen IV, “Benchmarking tree and ancestral sequence inference for B cell receptor sequences,” *Frontiers in Immunology*, vol. 9, p. 2451, 2018.
- [65] K. B. Hoehn, O. G. Pybus, and S. H. Kleinstein, “Phylogenetic analysis of migration, differentiation, and class switching in B cells,” *PLoS Computational Biology*, vol. 18, no. 4, p. e1009885, 2022.
- [66] M. G. Jones, A. Khodaverdian, J. J. Quinn, M. M. Chan, J. A. Hussmann, R. Wang, C. Xu, J. S. Weissman, and N. Yosef, “Inference of single-cell phylogenies from lineage tracing data using Cassiopeia,” *Genome Biology*, vol. 21, no. 1, pp. 1–27, 2020.
- [67] F. M. Davis, B. Lloyd-Lewis, O. B. Harris, S. Kozar, D. J. Winton, L. Muresan, and C. J. Watson, “Single-cell lineage tracing in the mammary gland reveals stochastic clonal dispersion of stem/progenitor cell progeny,” *Nature Communications*, vol. 7, no. 1, p. 13053, 2016.

- [68] P. Sashittal, H. Schmidt, M. Chan, and B. J. Raphael, “Startle: a star homoplasy approach for CRISPR-Cas9 lineage tracing,” *Cell Systems*, vol. 14, no. 12, pp. 1113–1121, 2023.
- [69] L. L. Weber, C. Zhang, I. Ochoa, and M. El-Kebir, “Phertilizer: Growing a clonal tree from ultra-low coverage single-cell DNA sequencing of tumors,” *PLoS Computational Biology*, vol. 19, no. 10, p. e1011544, 2023.
- [70] N. Abdollahi, L. Jeusset, A. de Septenville, F. Davi, and J. S. Bernardes, “Reconstructing B cell lineage trees with minimum spanning tree and genotype abundances,” *BMC Bioinformatics*, vol. 24, no. 1, p. 70, 2023.
- [71] S. Canzar, K. E. Neu, Q. Tang, P. C. Wilson, and A. A. Khan, “BASIC: BCR assembly from single cells,” *Bioinformatics*, vol. 33, no. 3, pp. 425–427, 2017.
- [72] M. El-Kebir, G. Satas, and B. J. Raphael, “Inferring parsimonious migration histories for metastatic cancers,” *Nature Genetics*, vol. 50, no. 5, pp. 718–726, 2018.
- [73] M. Slatkin and W. P. Maddison, “A cladistic measure of gene flow inferred from the phylogenies of alleles,” *Genetics*, vol. 123, no. 3, pp. 603–613, 1989.
- [74] L. L. Weber, N. Aguse, N. Chia, and M. El-Kebir, “PhyDOSE: Design of follow-up single-cell sequencing experiments of tumors,” *PLoS Computational Biology*, vol. 16, no. 10, p. e1008240, 2020.
- [75] M. El-Kebir, L. Oesper, H. Acheson-Field, and B. J. Raphael, “Reconstruction of clonal trees and tumor composition from multi-sample sequencing data,” *Bioinformatics*, vol. 31, no. 12, pp. i62–i70, June 2015.
- [76] S. Malikic, A. W. McPherson, N. Donmez, and C. S. Sahinalp, “Clonality inference in multiple tumor samples using phylogeny,” *Bioinformatics*, Jan 2015.
- [77] K. Yuan, T. Sakoparnig, F. Markowetz, and N. Beerenwinkel, “BitPhylogeny: a probabilistic framework for reconstructing intra-tumor phylogenies,” *Genome Biology*, vol. 16, no. 1, p. 36, Feb. 2015.
- [78] V. Popic, R. Salari, I. Hajirasouliha, D. Kashef-Haghighi, R. B. West, and S. Batzoglou, “Fast and scalable inference of multi-sample cancer lineages,” *Genome Biology*, vol. 16, no. 1, p. 91, May 2015.
- [79] E. M. Ross and F. Markowetz, “OncoNEM: inferring tumor evolution from single-cell sequencing data,” *Genome Biology*, vol. 17, no. 1, p. 69, 2016.
- [80] H. Zafar, A. Tzen, N. Navin, K. Chen, and L. Nakhleh, “SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models,” *Genome Biology*, vol. 18, no. 1, p. 178, Sep. 2017.
- [81] N. E. Navin, “Cancer genomics: one cell at a time,” *Genome Biology*, vol. 15, no. 8, p. 452, 2014.

- [82] S. Malikic, K. Jahn, J. Kuipers, S. C. Sahinalp, and N. Beerenwinkel, “Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data,” *Nature Communications*, vol. 10, no. 1, pp. 1–12, June 2019.
- [83] Y. Kuboki et al., “Single-cell sequencing defines genetic heterogeneity in pancreatic cancer precursor lesions,” *The Journal of Pathology*, vol. 247, no. 3, pp. 347–356, Mar. 2019.
- [84] M. L. Leung, A. Davis, R. Gao, A. Casasent, Y. Wang, E. Sei, E. Sanchez, D. Maru, S. Kopetz, and N. E. Navin, “Single cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer,” *Genome Research*, p. gr.209973.116, May 2017.
- [85] C. Gawad, W. Koh, and S. R. Quake, “Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 50, pp. 17 947–17 952, 2014.
- [86] A. McPherson et al., “Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer,” *Nature Genetics*, May 2016.
- [87] C. Kim, R. Gao, E. Sei, R. Brandt, J. Hartman, T. Hatschek, N. Crosetto, T. Foukakis, and N. E. Navin, “Chemoresistance evolution in triple-negative breast cancer delineated by single-cell sequencing,” *Cell*, vol. 173, no. 4, pp. 879–893, 2018.
- [88] K. Morita et al., “Clonal evolution of acute myeloid leukemia revealed by high-throughput single-cell genomics,” *bioRxiv*, 2020. [Online]. Available: <https://www.biorxiv.org/content/early/2020/02/07/2020.02.07.925743>
- [89] M. Jamal-Hanjani et al., “Tracking the evolution of non-small-cell lung cancer,” *New England Journal of Medicine*, vol. 376, no. 22, pp. 2109–2121, 2017.
- [90] N. Bolli, H. Avet-Loiseau, D. C. Wedge, P. Van Loo, L. B. Alexandrov, P. J. Campbell, N. C. Munshi et al., “Heterogeneity of genomic evolution and mutational profiles in multiple myeloma,” *Nature Communications*, vol. 5, Jan. 2014.
- [91] S. C. Dentre, D. C. Wedge, and P. Van Loo, “Principles of Reconstructing the Subclonal Architecture of Cancers.” *Cold Spring Harbor Perspectives in Medicine*, vol. 7, no. 8, p. a026625, Aug. 2017.
- [92] P. J. Stephens et al., “The landscape of cancer genes and mutational processes in breast cancer,” *Nature*, vol. 486, no. 7403, pp. 400–404, June 2012.
- [93] B. Levin, “A Representation for Multinomial Cumulative Distribution Functions,” *The Annals of Statistics*, vol. 9, no. 5, pp. 1123–1126, Sep. 1981.
- [94] Y. Fu, C. Li, S. Lu, W. Zhou, F. Tang, X. S. Xie, and Y. Huang, “Uniform and accurate single-cell sequencing based on emulsion whole-genome amplification.” *Proceedings of the National Academy of Sciences*, vol. 112, no. 38, pp. 11 923–11 928, Sep. 2015.

- [95] H. Prüfer, “Neuer beweis eines satzes uber permutationen,” *Arch Math Phys*, vol. 27, pp. 742–4, 1918.
- [96] A. Roth et al., “PyClone: statistical inference of clonal population structure in cancer.” *Nature Methods*, vol. 11, no. 4, pp. 396–398, Apr. 2014.
- [97] L. D. Brown, T. T. Cai, and A. DasGupta, “Interval estimation for a binomial proportion,” *Statistical Science*, pp. 101–117, 2001.
- [98] DECIBIO INSIGHTS, “Mission Bio launches single-cell DNA analysis platform at ashg 2017,” 2017. [Online]. Available: <https://www.decibio.com/2017/10/17/mission-bio-launches-single-cell-dna-analysis-platform-ashg-2017/>
- [99] Bioinformatics, “ASHG 2017: New single-cell, CRISPR and NGS products highlight lab technology’s progress,” 2017. [Online]. Available: <https://bioinfoinc.com/digest/ashg-2017-new-single-cell-crispr-ngs-products-highlight-lab-technologys-progress/>
- [100] N. Aguse, Y. Qi, and M. El-Kebir, “Summarizing the solution space in tumor phylogeny inference by multiple consensus trees,” *Bioinformatics*, vol. 35, no. 14, pp. i408–i416, July 2019.
- [101] L. L. Weber, P. Sashittal, and M. El-Kebir, “doubletD: detecting doublets in single-cell DNA sequencing data,” *Bioinformatics*, vol. 37, no. Supplement_1, pp. i214–i221, 2021.
- [102] C. F. De Bourcy et al., “A quantitative comparison of single-cell whole genome amplification methods,” *PLOS One*, vol. 9, no. 8, p. e105585, 2014.
- [103] H. Liu et al., “Improving single-cell encapsulation efficiency and reliability through neutral buoyancy of suspension,” *Micromachines*, vol. 11, no. 1, p. 94, 2020.
- [104] N. E. Navin and K. Chen, “Genotyping tumor clones from single-cell data,” *Nature Methods*, vol. 13, no. 7, pp. 555–556, 2016.
- [105] H. Zafar et al., “Computational approaches for inferring tumor evolution from single-cell genomic data,” *Current Opinion in Systems Biology*, vol. 7, pp. 16–25, 2018.
- [106] L. A. Miles et al., “Single-cell mutation analysis of clonal evolution in myeloid malignancies,” *Nature*, vol. 587, no. 7834, pp. 477–482, 2020.
- [107] S. Salehi et al., “Clonal fitness inferred from time-series modelling of single-cell cancer genomes,” *Nature*, vol. 595, no. 7868, pp. 585–590, 2021.
- [108] J. Kuipers et al., “Single-cell sequencing data reveal widespread recurrence and loss of mutational hits in the life histories of tumors,” *Genome Research*, vol. 27, no. 11, pp. 1885–1894, Nov. 2017.

- [109] Y. Wu, “Accurate and efficient cell lineage tree inference from noisy single cell data: the maximum likelihood perfect phylogeny approach,” *Bioinformatics*, vol. 36, no. 3, pp. 742–750, 2020.
- [110] E. A. DePasquale et al., “DoubletDecon: deconvoluting doublets from single-cell RNA-sequencing data,” *Cell Reports*, vol. 29, no. 6, pp. 1718–1727, 2019.
- [111] N. M. Xi and J. J. Li, “Benchmarking computational doublet-detection methods for single-cell RNA sequencing data,” *Cell Systems*, 2020.
- [112] L. J. Luquette et al., “Identification of somatic mutations in single cell DNA-seq using a spatial model of allelic imbalance,” *Nature Communications*, vol. 10, no. 1, p. 3908, Dec. 2019.
- [113] S. Zaccaria and B. J. Raphael, “Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL,” *Nature Biotechnology*, Sep. 2020.
- [114] B. Hwang et al., “Single-cell RNA sequencing technologies and bioinformatics pipelines,” *Experimental & Molecular Medicine*, vol. 50, no. 8, pp. 1–14, 2018.
- [115] G. Chen et al., “Single-cell RNA-seq technologies and related computational data analysis,” *Frontiers in Genetics*, vol. 10, p. 317, 2019.
- [116] M. Gerstung et al., “Reliable detection of subclonal single-nucleotide variants in tumour cell populations,” *Nature Communications*, vol. 3, no. 1, pp. 1–8, 2012.
- [117] H. Zafar et al., “Monovar: single-nucleotide variant detection in single cells,” *Nature Methods*, vol. 13, no. 6, pp. 505–507, 2016.
- [118] D. Posada, “CellCoal: coalescent simulation of single-cell sequencing samples,” *Molecular Biology and Evolution*, vol. 37, no. 5, pp. 1535–1542, 2020.
- [119] D. Lähnemann, J. Köster, U. Fischer, A. Borkhardt, A. C. McHardy, and A. Schönhuth, “Accurate and scalable variant calling from single cell DNA sequencing data with ProSolo,” *Nature Communications*, vol. 12, no. 1, p. 6744, 2021.
- [120] C. Gawad et al., “Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 50, pp. 17947–17952, 2014.
- [121] A. McPherson et al., “Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer,” *Nature Genetics*, vol. 48, no. 7, p. 758, 2016.
- [122] G. Satas et al., “Scarlet: Single-cell tumor phylogeny inference with copy-number constrained mutation losses,” *Cell Systems*, vol. 10, no. 4, pp. 323–332, 2020.
- [123] Mission Bio, “Performance of the Tapestry platform for single-cell targeted DNA sequencing,” Tech. Rep., 2019. [Online]. Available: https://missionbio.com/resources/technical-notes/platform_white_paper/

- [124] S. Malikic et al., “Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data,” *Nature Communications*, vol. 10, no. 1, pp. 1–12, 2019.
- [125] L. L. Weber and M. El-Kebir, “Distinguishing linear and branched evolution given single-cell DNA sequencing data of tumors,” *Algorithms for Molecular Biology*, vol. 16, pp. 1–12, 2021.
- [126] L. L. Weber and M. El-Kebir, “Phyolin: Identifying a linear perfect phylogeny in single-cell dna sequencing data of tumors,” *Leibniz international proceedings in informatics*, vol. 172, 2020.
- [127] M. El-Kebir, L. Oesper, H. Acheson-Field, and B. J. Raphael, “Reconstruction of clonal trees and tumor composition from multi-sample sequencing data,” *Bioinformatics*, vol. 31, no. 12, pp. i62–i70, 2015.
- [128] E. S. Azer, M. H. Ebrahimabadi, S. Malikić, R. Khardon, and S. C. Sahinalp, “Tumor phylogeny topology inference via deep learning,” *iScience*, vol. 23, no. 11, p. 101655, 2020.
- [129] D. Gusfield, *ReCombinatorics: the algorithmics of ancestral recombination graphs and explicit phylogenetic networks*. MIT press, 2014.
- [130] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [131] D. Chen, O. Eulenstein, D. Fernandez-Baca, and M. Sanderson, “Minimum-flip supertrees: complexity and algorithms,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 3, no. 2, pp. 165–173, 2006.
- [132] M. Yannakakis, “Computing the minimum fill-in is NP-complete,” *SIAM Journal on Algebraic Discrete Methods*, vol. 2, no. 1, pp. 77–79, 1981.
- [133] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach Third Edition*. Pearson, 2010.
- [134] R. R. Hudson, “Generating samples under a Wright–Fisher neutral model of genetic variation,” *Bioinformatics*, vol. 18, no. 2, pp. 337–338, 2002.
- [135] A. Schuh et al., “Monitoring chronic lymphocytic leukemia progression by whole genome sequencing reveals heterogeneous clonal evolution patterns,” *Blood, The Journal of the American Society of Hematology*, vol. 120, no. 20, pp. 4191–4196, 2012.
- [136] X. Fu, H. Lei, Y. Tao, and R. Schwartz, “Reconstructing tumor clonal lineage trees incorporating single-nucleotide variants, copy number alterations and structural variations,” *Bioinformatics*, vol. 38, no. Supplement_1, pp. i125–i133, 2022.

- [137] J. Kannan, L. Mathews, Z. Wu, N. S. Young, and S. Gao, “CAISC: A software to integrate copy number variations and single nucleotide mutations for genetic heterogeneity profiling and subclone detection by single-cell RNA sequencing,” *BMC Bioinformatics*, vol. 23, no. 3, pp. 1–17, 2022.
- [138] A. Roth et al., “Clonal genotype and population structure inference from single-cell tumor sequencing,” *Nature Methods*, vol. 13, no. 7, pp. 573–576, 2016.
- [139] S. Zaccaria and B. J. Raphael, “Characterizing allele-and haplotype-specific copy numbers in single cells with chisel,” *Nature Biotechnology*, vol. 39, no. 2, pp. 207–214, 2021.
- [140] Y. Liu, M. Edrisi, Z. Yan, H. A. Ogilvie, and L. Nakhleh, “NestedBD: Bayesian inference of phylogenetic trees from single-cell copy number profiles under a birth-death model,” *Algorithms for Molecular Biology*, vol. 19, no. 1, p. 18, 2024.
- [141] A. Kozlov, J. M. Alves, A. Stamatakis, and D. Posada, “CellPhy: accurate and fast probabilistic inference of single-cell phylogenies from scDNA-seq data,” *Genome Biology*, vol. 23, no. 1, pp. 1–30, 2022.
- [142] S. Kang, N. Borgsmüller, M. Valecha, J. Kuipers, J. M. Alves, S. Prado-López, D. Chantada, N. Beerenwinkel, D. Posada, and E. Szczurek, “SIEVE: joint inference of single-nucleotide variants and cell phylogeny from single-cell dna sequencing data,” *Genome Biology*, vol. 23, no. 1, p. 248, 2022.
- [143] K. Chen, J. C. Moravec, A. Gavryushkin, D. Welch, and A. J. Drummond, “Accounting for errors in data improves divergence time estimates in single-cell cancer evolution,” *Molecular Biology and Evolution*, vol. 39, no. 8, p. msac143, 2022.
- [144] S. Milite, R. Bergamin, L. Patruno, N. Calonaci, and G. Caravagna, “A Bayesian method to cluster single-cell RNA sequencing data using copy number alterations,” *Bioinformatics*, vol. 38, no. 9, pp. 2512–2518, 2022.
- [145] Z. Zhou, B. Xu, A. Minn, and N. R. Zhang, “DENDRO: genetic heterogeneity profiling and subclone detection by single-cell RNA sequencing,” *Genome Biology*, vol. 21, no. 1, pp. 1–15, 2020.
- [146] M. A. Myers, S. Zaccaria, and B. J. Raphael, “Identifying tumor clones in sparse single-cell mutation data,” *Bioinformatics*, vol. 36, no. Supplement_1, pp. i186–i193, 2020.
- [147] H. Rozhoňová, D. Danciu, S. Stark, G. Rättsch, A. Kahles, and K.-V. Lehmann, “SE-CEDO: SNV-based subclone detection using ultra-low coverage single-cell DNA sequencing,” *Bioinformatics*, vol. 38, no. 18, pp. 4293–4300, 2022.
- [148] M. Kimura, “The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations,” *Genetics*, vol. 61, no. 4, p. 893, 1969.
- [149] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

- [150] L. Dollo, “Les lois de l’évolution,” *Bulletin de la Société belge de géologie, de paléontologie et d’hydrologie*, vol. 7, pp. 164–166, 1893.
- [151] E. Cerami et al., “The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data,” *Cancer Discovery*, vol. 2, no. 5, pp. 401–404, 2012.
- [152] J. Gao et al., “Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal,” *Science Signaling*, vol. 6, no. 269, pp. pl1–pl1, 2013.
- [153] Z. Sondka, S. Bamford, C. G. Cole, S. A. Ward, I. Dunham, and S. A. Forbes, “The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers,” *Nature Reviews Cancer*, vol. 18, no. 11, pp. 696–705, 2018.
- [154] G. A. Van der Auwera and B. D. O’Connor, *Genomics in the cloud: using Docker, GATK, and WDL in Terra*. O’Reilly Media, 2020.
- [155] A. P. G. Consortium et al., “AACR project GENIE: powering precision medicine through an international consortium,” *Cancer Discovery*, vol. 7, no. 8, pp. 818–831, 2017.
- [156] A. Heravi-Moussavi et al., “Recurrent somatic DICER1 mutations in nonepithelial ovarian cancers,” *New England Journal of Medicine*, vol. 366, no. 3, pp. 234–242, 2012.
- [157] P. M. Schnepp, M. Chen, E. T. Keller, and X. Zhou, “SNV identification from single-cell RNA sequencing data,” *Human Molecular Genetics*, vol. 28, no. 21, pp. 3569–3583, 2019.
- [158] F. Liu, Y. Zhang, L. Zhang, Z. Li, Q. Fang, R. Gao, and Z. Zhang, “Systematic comparative analysis of single-nucleotide variant detection methods from single-cell RNA sequencing data,” *Genome Biology*, vol. 20, no. 1, pp. 1–15, 2019.
- [159] J. Dou et al., “Single-nucleotide variant calling in single-cell sequencing data with Monopogen,” *Nature Biotechnology*, pp. 1–10, 2023.
- [160] F. Muyas, C. M. Sauer, J. E. Valle-Inclán, R. Li, R. Rahbari, T. J. Mitchell, S. Hormoz, and I. Cortés-Ciriano, “De novo detection of somatic mutations in high-throughput single-cell profiling data sets,” *Nature Biotechnology*, pp. 1–10, 2023.
- [161] A. R. Massarat, A. Sen, J. Jauregui, S. T. Tyndale, Y. Fu, G. Erikson, and G. McVicker, “Discovering single nucleotide variants and indels from bulk and single-cell ATAC-seq,” *Nucleic Acids Research*, vol. 49, no. 14, pp. 7986–7994, 2021.
- [162] G. Quinones-Valdez, T. Fu, T. W. Chan, and X. Xiao, “scAllele: A versatile tool for the detection and analysis of variants in scRNA-seq,” *Science Advances*, vol. 8, no. 35, p. eabn6398, 2022.

- [163] W. Gasper, F. Rossi, M. Ligorio, and D. Gherzi, “Variant calling enhances the identification of cancer cells in single-cell RNA sequencing data,” *PLoS Computational Biology*, vol. 18, no. 10, p. e1010576, 2022.
- [164] D. Ramazzotti, F. Angaroni, D. Maspero, G. Ascolani, I. Castiglioni, R. Piazza, M. Antoniotti, and A. Graudenzi, “Variant calling from scRNA-seq data allows the assessment of cellular identity in patient-derived cell lines,” *Nature Communications*, vol. 13, no. 1, p. 2718, 2022.
- [165] M. El-Kebir, Q. Morris, L. Oesper, and S. C. Sahinalp, “Emerging topics in cancer evolution,” in *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2022*. World Scientific, 2021, pp. 397–401.
- [166] P. Sashittal, S. Zaccaria, and M. El-Kebir, “Parsimonious clone tree integration in cancer,” *Algorithms for Molecular Biology*, vol. 17, no. 1, pp. 1–14, 2022.
- [167] A. Antonello, et al., “Computational validation of clonal and subclonal copy number alterations from bulk tumor sequencing using CNAqc,” *Genome Biology*, vol. 25, no. 1, pp. 1–40, 2024.
- [168] P. Sashittal, H. Zhang, C. A. Iacobuzio-Donahue, and B. J. Raphael, “ConDoR: tumor phylogeny inference with a copy-number constrained mutation loss model,” *Genome Biology*, vol. 24, no. 1, p. 272, 2023.
- [169] E. Sollier, J. Kuipers, K. Takahashi, N. Beerenwinkel, and K. Jahn, “COMPASS: joint copy number and mutation phylogeny reconstruction from amplicon single-cell sequencing data,” *Nature Communications*, vol. 14, no. 1, p. 4921, 2023.
- [170] L. Zhang, H. W. Bass, J. Irianto, and X. Mallory, “Integrating SNVs and CNAs on a phylogenetic tree from single-cell DNA sequencing data,” *Genome Research*, vol. 33, no. 11, pp. 2002–2017, 2023.
- [171] S. Ivanovic and M. El-Kebir, “Evolution-Aware Deep Reinforcement Learning for Single-Cell DNA Copy Number Calling,” *bioRxiv*, pp. 2024–03, 2024.
- [172] J. Demeulemeester, S. C. Dentro, M. Gerstung, and P. Van Loo, “Biallelic mutations in cancer genomes reveal local mutational determinants,” *Nature Genetics*, vol. 54, no. 2, pp. 128–133, 2022.
- [173] M. Kimura and J. F. Crow, “The number of alleles that can be maintained in a finite population,” *Genetics*, vol. 49, no. 4, p. 725, 1964.
- [174] G. Satas, S. Zaccaria, M. El-Kebir, and B. J. Raphael, “DeCiFering the elusive cancer cell fraction in tumor heterogeneity and evolution,” *Cell Systems*, vol. 12, no. 10, pp. 1004–1018, 2021.
- [175] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.

- [176] L. Weber, D. Reiman, M. Roddur, Y. Qi, M. El-Kebir, and A. Khan, “TRIBAL: Tree inference of B cell clonal lineages,” *bioRxiv*, 2023.
- [177] L. L. Weber, D. Reiman, M. S. Roddur, Y. Qi, M. El-Kebir, and A. A. Khan, “TRIBAL: Tree inference of B cell clonal lineages,” in *International Conference on Research in Computational Molecular Biology*. Springer, 2024, pp. 364–367.
- [178] S. Tonegawa, “Somatic generation of antibody diversity,” *Nature*, vol. 302, no. 5909, pp. 575–581, 1983.
- [179] D. Reiman, G. K. Manakkat Vijay, H. Xu, A. Sonin, D. Chen, N. Salomonis, H. Singh, and A. A. Khan, “Pseudocell Tracer—a method for inferring dynamic trajectories using scRNAseq and its application to B cells undergoing immunoglobulin class switch recombination,” *PLoS Computational Biology*, vol. 17, no. 5, p. e1008094, 2021.
- [180] G. X. Y. Zheng et al., “Massively parallel digital transcriptional profiling of single cells,” *Nature Communications*, vol. 8, no. 1, p. 14049, 2017.
- [181] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [182] F. Horns, C. Vollmers, D. Croote, S. F. Mackey, G. E. Swan, C. L. Dekker, M. M. Davis, and S. R. Quake, “Lineage tracing of human B cells reveals the in vivo landscape of human antibody class switching,” *Elife*, vol. 5, p. e16578, 2016.
- [183] O. Ronen, J. R. Rohlicek, and M. Ostendorf, “Parameter estimation of dependence tree models using the EM algorithm,” *IEEE Signal Processing Letters*, vol. 2, no. 8, pp. 157–159, 1995.
- [184] C. Chow and C. Liu, “Approximating discrete probability distributions with dependence trees,” *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.
- [185] S. Sridhar, F. Lam, G. E. Blelloch, R. Ravi, and R. Schwartz, “Mixed integer linear programming for maximum-parsimony phylogeny inference,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 3, pp. 323–331, 2008.
- [186] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [187] D. F. Robinson and L. R. Foulds, “Comparison of phylogenetic trees,” *Mathematical Biosciences*, vol. 53, no. 1-2, pp. 131–147, 1981.
- [188] D. Sankoff, “Minimal mutation trees of sequences,” *SIAM Journal on Applied Mathematics*, vol. 28, no. 1, pp. 35–42, 1975.
- [189] A. Cumano and K. Rajewsky, “Clonal recruitment and somatic mutation in the generation of immunological memory to the hapten NP.” *The EMBO Journal*, vol. 5, no. 10, pp. 2459–2468, 1986.

- [190] D. Chen et al., “Coupled analysis of transcriptome and BCR mutations reveals role of OXPHOS in affinity maturation,” *Nature Immunology*, vol. 22, no. 7, pp. 904–913, 2021.
- [191] E. Stephenson et al., “Single-cell multi-omics analysis of the immune response in COVID-19,” *Nature Medicine*, vol. 27, no. 5, pp. 904–916, 2021.
- [192] D. Allen, T. Simon, F. Sablitzky, K. Rajewsky, and A. Cumano, “Antibody engineering for the analysis of affinity maturation of an anti-hapten response.” *The EMBO Journal*, vol. 7, no. 7, pp. 1995–2001, 1988.
- [193] A. A. Weiser, N. Wittenbrink, L. Zhang, A. I. Schmelzer, A. Valai, and M. Or-Guil, “Affinity maturation of B cells involves not only a few but a whole spectrum of relevant mutations,” *International Immunology*, vol. 23, no. 5, pp. 345–356, 2011.
- [194] K. Furukawa, A. Akasako-Furukawa, H. Shirai, H. Nakamura, and T. Azuma, “Junctional amino acids determine the maturation pathway of an antibody,” *Immunity*, vol. 11, no. 3, pp. 329–338, 1999.
- [195] J. Jacob, J. Przylepa, C. Miller, and G. Kelsoe, “In situ studies of the primary immune response to (4-hydroxy-3-nitrophenyl) acetyl. iii. the kinetics of V region mutation and selection in germinal center B cells.” *The Journal of Experimental Medicine*, vol. 178, no. 4, pp. 1293–1307, 1993.
- [196] K. M. Nickerson et al., “Age-associated B cells are heterogeneous and dynamic drivers of autoimmunity in mice,” *Journal of Experimental Medicine*, vol. 220, no. 5, p. e20221346, 2023.
- [197] J. C. Ng, G. Montamat Garcia, A. T. Stewart, P. Blair, C. Mauri, D. K. Dunn-Walters, and F. Fraternali, “sciCSR infers B cell state transition and predicts class-switch recombination dynamics using single-cell transcriptomic data,” *Nature Methods*, pp. 1–12, 2023.
- [198] J. Tan et al., “A LAIR1 insertion generates broadly reactive antibodies against malaria variant antigens,” *Nature*, vol. 529, no. 7584, pp. 105–109, 2016.
- [199] C. Lupo, N. Spisak, A. M. Walczak, and T. Mora, “Learning the statistics and landscape of somatic mutation-induced insertions and deletions in antibodies,” *PLOS Computational Biology*, vol. 18, no. 6, p. e1010167, 2022.
- [200] R. M. Karp, *Reducibility among Combinatorial Problems*. Springer, 1972, pp. 85–103.
- [201] C. E. B. Owen, “Parameter estimation for the beta distribution,” 2008.
- [202] M. A. Lodato et al., “Somatic mutation in single human neurons tracks developmental and transcriptional history,” *Science*, vol. 350, no. 6256, pp. 94–98, 2015.

- [203] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the surprising behavior of distance metrics in high dimensional space,” in *International Conference on Database Theory*. Springer, 2001, pp. 420–434.
- [204] H.-P. Kriegel, P. Kröger, and A. Zimek, “Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,” *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, pp. 1–58, 2009.
- [205] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [206] A. Damle, V. Minden, and L. Ying, “Simple, direct and efficient multi-way spectral clustering,” *Information and Inference: A Journal of the IMA*, vol. 8, no. 1, pp. 181–203, 2019.
- [207] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS: Ordering points to identify the clustering structure,” *ACM Sigmod Record*, vol. 28, no. 2, pp. 49–60, 1999.
- [208] W. McLaren, L. Gil, S. E. Hunt, H. S. Riat, G. R. Ritchie, A. Thormann, P. Flicek, and F. Cunningham, “The Ensembl variant effect predictor,” *Genome Biology*, vol. 17, no. 1, pp. 1–14, 2016.
- [209] R. Vaser, S. Adusumalli, S. N. Leng, M. Sikic, and P. C. Ng, “SIFT missense predictions for genomes,” *Nature Protocols*, vol. 11, no. 1, pp. 1–9, 2016.
- [210] I. Adzhubei, D. M. Jordan, and S. R. Sunyaev, “Predicting functional effect of human missense mutations using PolyPhen-2,” *Current Protocols in Human Genetics*, vol. 76, no. 1, pp. 7–20, 2013.
- [211] L. R. Foulds and R. L. Graham, “The Steiner problem in phylogeny is NP-complete,” *Advances in Applied Mathematics*, vol. 3, no. 1, pp. 43–49, 1982.