Phertilizer: Growing a Clonal Tree from Ultra-low Coverage Single-cell DNA Sequencing of Tumors

Leah L. Weber[1,*], Chuanyi Zhang[2,*], Idoia Ochoa[2,3,†], and Mohammed El-Kebir[1,4,†]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL 61801, USA
[2]Department of Electrical & Computer Engineering, University of Illinois at Urbana-Champaign, IL 61801, USA
[3]Department of Electrical and Electronics Engineering, University of Navarra, Donostia, Spain
[4]Cancer Center at Illinois, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
[*]Joint first authorship
[†]Correspondence: {idoia,melkebir}@illinois.edu

Abstract

Emerging ultra-low coverage single-cell DNA sequencing (scDNA-seq) technologies have enabled high resolution evolutionary studies of copy number aberrations (CNAs) within tumors. While these sequencing technologies are well suited for identifying CNAs due to the uniformity of sequencing coverage, the sparsity of coverage poses challenges for the study of single-nucleotide variants (SNVs). In order to maximize the utility of increasingly available ultra-low coverage scDNA-seq data and obtain a comprehensive understanding of tumor evolution, it is important to also analyze the evolution of SNVs from the same set of tumor cells.

We present PHERTILIZER, a method to infer a clonal tree from ultra-low coverage scDNA-seq data of a tumor. Based on a probabilistic model, our method recursively partitions the data by identifying key evolutionary events in the history of the tumor. We demonstrate the performance of PHERTILIZER on simulated data as well as on two real datasets, finding that PHERTILIZER effectively utilizes the copy-number signal inherent in the data to more accurately uncover clonal structure and genotypes compared to previous methods.

Availability: https://github.com/elkebir-group/phertilizer

Acknowledgements. We thank the Navin Lab and Darlan Minussi for their assistance with the ACT data. Additionally, we thank the Shah Lab, including Daniel Lai, Robert Reinert, and Andrew McPherson, for their assistance with the DLP+ data. M.E-K. was supported by the National Science Foundation (CCF-2046488) as well as funding from the Cancer Center at Illinois. I.O. was supported by a Gipuzkoa Fellows grant from the Basque Government, a Ramon y Cajal Grant from Spain, and a grant from the Spanish Ministry of Science and Innovation (PID2021-126718OA-I00). This work used resources, services, and support provided via the Greg Gulick Honorary Research Award Opportunity supported by a gift from Amazon Web Services.

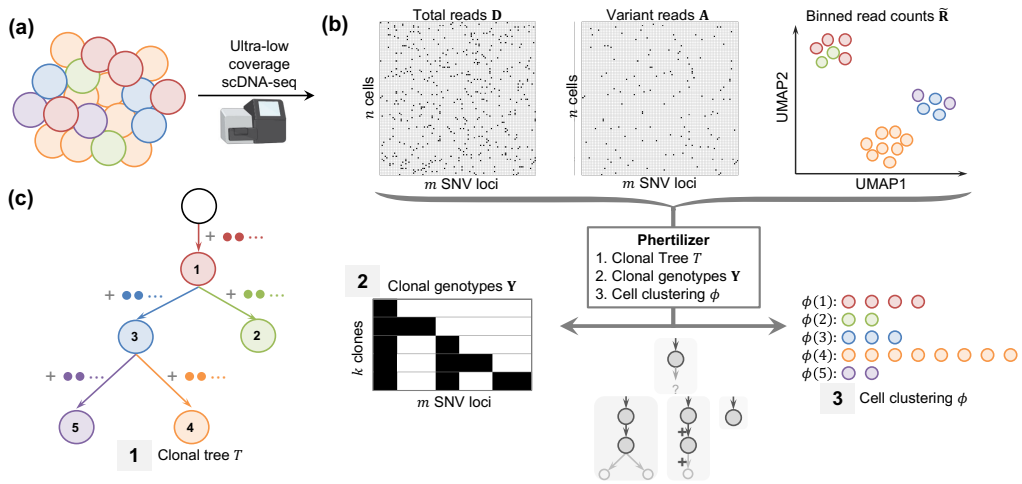


Figure 1: **PHERTILIZER infers a clonal tree T, clonal genotypes $\mathbf{Y}$ and a cell clustering ϕ given ultra-low coverage single-cell sequencing data**. (a) A tumor consists of clones with distinct genotypes. (b) Ultra-low coverage scDNA-seq produces total read counts $\mathbf{D} \in \mathbb{N}^{n \times m}$ 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 $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$.

1 Introduction

Cancer results from an evolutionary process that yields a heterogeneous tumor composed of multiple subpopulations of cells, or *clones*, with distinct sets of somatic mutations [1] (Fig. 1a). These mutations include single-nucleotide variants (SNVs) that alter a single base and copy-number aberrations (CNAs) that amplify or delete large genomic regions. Over the last decade, new developments in single-cell DNA sequencing (scDNA-seq) methods have helped uncover a wealth of insights regarding intra-tumor heterogeneity and cancer evolution [2–5]. In particular, the ongoing development and application of high-throughput, ultra-low coverage scDNA-seq technologies ($< 1\times$), such as direct library preparation (DLP+) [6] and acoustic cell tagmentation (ACT) [7], have paved the way for enriching our understanding of the role CNAs play in cancer progression and tumor evolution [6–8].

The advantage these ultra-low coverage scDNA-seq technologies have over other high-throughput scDNA-seq methods ($> 1\times$), like Mission Bio Tapestry [9], 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 [10] and CASIC [11], 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 [12–26]. One exception in the medium to high coverage scDNA-seq regime is SCARLET [27], 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 [28] 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 [4] 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 [29] 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 methods 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 [6]. 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 PHERTILIZER, 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, PHERTILIZER seeks to grow a clonal tree with maximum posterior probability by recursively inferring elementary clonal trees as building blocks. Our simulations demonstrate that PHERTILIZER accurately infers phylogenies and cell clusters when the number of cells matches current practice. In particular, PHERTILIZER outperforms a current method [28] for simultaneously clustering SNVs and cells as well as another commonly used *ad hoc* approach [6]. On real data, we find that PHERTILIZER effectively utilizes the copy-number signal inherent in the data to uncover clonal structure, yielding high-fidelity clonal genotypes.

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

Definition 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 [11] 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 A.1. Third, similarly to methods such as SCITE [12], SciCloneFit [15] and SPhyR [14], 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 [30] defined as follows.

Definition 3. A tree T with nodes $\{v_1, \ldots, v_k\}$ rooted at node v_1 is a *clonal tree* for clonal genotypes $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_k]^\top$ 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 A.2 that describes the generation of variant read counts $\mathbf{A}$ and the binned read count embedding $\tilde{\mathbf{R}}$. This model 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 (Fig. S1). Importantly, while Definitions 1 to 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 A.2.4). 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 A.3 and Fig. S2 for details and in particular Definition 4. This leads to the following problem.

Problem 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 $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, c, \alpha)$.

3 Methods

To solve the CTI problem, PHERTILIZER maintains a set $\mathcal{T}$ of candidate trees throughout three phases: (i) initialization, (ii) enumeration, 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 enumeration phase, 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 $T \in \mathcal{T}$(Fig. 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. 2). The key idea is that each elementary

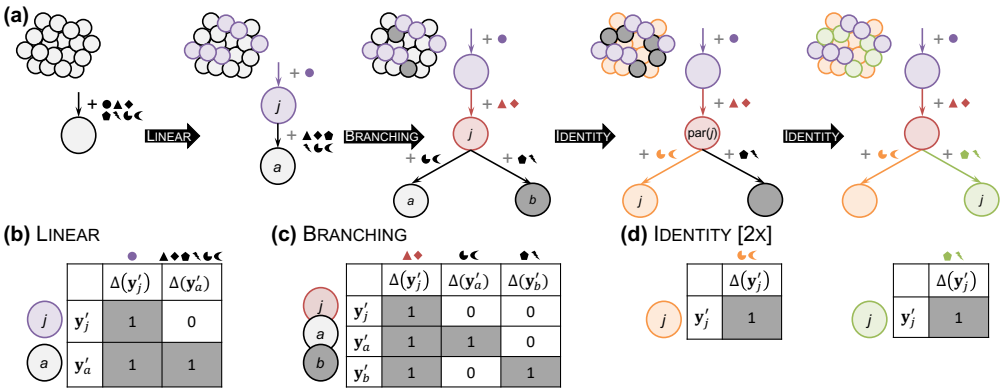


Figure 2: **PHERTILIZER solves the CTI problem by enumerating clonal trees using three elementary operations in a recursive fashion.** (a) Each operation yields a new clonal tree T' by extending a leaf v_j of the previous clonal tree T and reassigning its SNVs $\Delta(\mathbf{y}_j)$ and cells $\phi(j)$. The resulting clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' are constrained as depicted: (b) `Linear`, (c) `Branching`, and (d) `Identity`.

tree operation breaks down the CTI problem into smaller subproblems. Intuitively, a `Linear` operation (Fig. 2a) replaces a leaf node with a two node linear subtree, while a `Branching` operation (Fig. 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 [31]. The `Identity` (Fig. 2c) operation does not modify the tree.

These operations are defined more formally in Appendix A.4. While the specifics vary slightly, both `Linear` (Appendix A.4.1) and `Branching` (Appendix A.4.2) 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 [32], we incorporate the binned read count embedding $\tilde{\mathbf{R}}$ and variant read counts $\mathbf{A}$ into a single feature [11]. We then use this feature as input to the normalized cut algorithm [32] 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. We terminate this process upon convergence or when a maximum number of iterations is reached. 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 (Appendix A.3) and meet additional regularization criteria (Appendix A.4.3).

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 A.4.4). 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. PHERTILIZER is implemented in Python 3, open source (BSD-3-Clause), and available at `https://github.com/elkebir-group/phertilizer`.

4 Results

4.1 Simulation study

Overview. To assess the performance of PHERTILIZER 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 B.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 [33], 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. 3a and S6 for examples and to Appendix B.1.4 for more formal definitions.

We benchmarked against SBMCLONE [28] 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 [6, 7]. In BASELINE, cells are first clustered into clones from the read count embedding $\tilde{\mathbf{R}}$. 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 B.1.1). Since there exists no standalone method that performs cell clustering, genotyping and tree inference for ultra-low coverage scDNA-seq, we paired SBMCLONE and BASELINE with SCITE [12].

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 B 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. 3b, Fig. S8). 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. 3b, Fig. S8). PHERTILIZER achieved the highest overall cell accuracy (median: 0.82) in

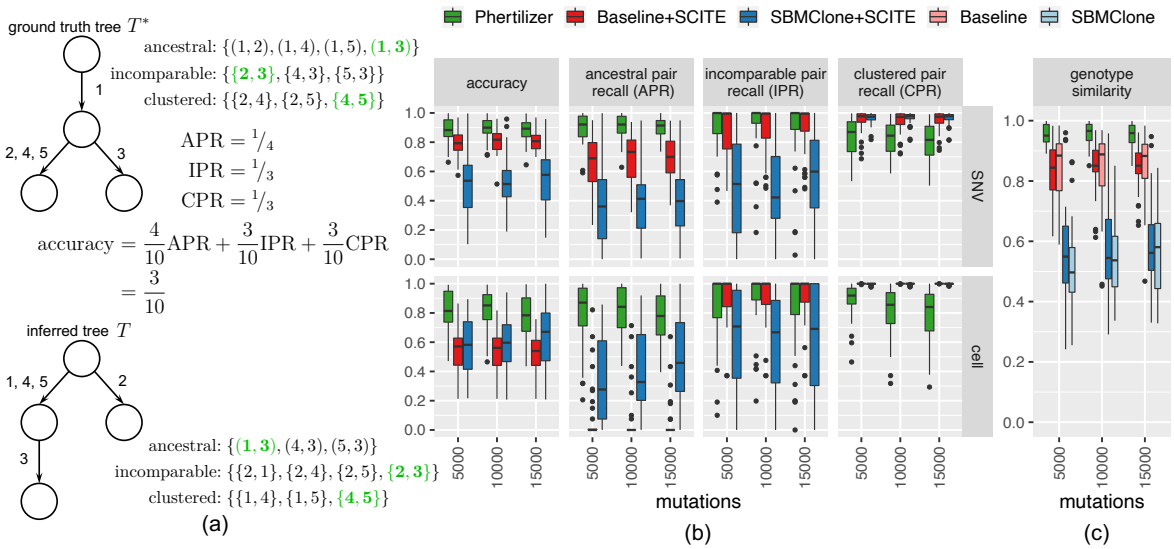


Figure 3: **PHERTILIZER outperforms BASELINE and SBMCLONE on simulated data.** We show aggregated results for $n \in \{1000, 2000\}$ cells, $k \in \{5, 9\}$ clones and coverage $g = 0.05\times$. (a) Example of SNV accuracy, ancestral pair recall (APR), clustered pair recall (CPR), and incomparable pair recall (IPR) metrics. See Fig. S6 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.

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 or each simulated cell against its ground truth genotype, it captures the interplay between cell clustering and clonal genotyping. Given that PHERTILIZER achieved the highest performance on the clonal tree inference and cell clustering metrics, we would expect PHERTILIZER to have the highest genotype similarity. Indeed, Fig. 3c demonstrates that to be true since PHERTILIZER 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 $g = 0.01\times$, PHERTILIZER maintained top

performance on both cell placement and genotype similarity but BASELINE+SCITE was competitive with PHERTILIZER on SNV placement (Fig. S7). For the highest coverage $g = 0.1\times$, PHERTILIZER 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. S9). 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 PHERTILIZER was $460\,\mathrm{s}$, $45.9\,\mathrm{s}$ for BASELINE+SCITE and $101\,\mathrm{s}$ for SBMCLONE+SCITE (Fig. S12).

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 PHERTILIZER still outperformed SBMCLONE+SCITE but that for coverage $g = 0.01\times$ our performance were slightly worse than simulations with CNAs (Fig. S10), implying CNA features aid inference when sequencing coverage is extremely sparse. For the second, simulations were generated under a Dollo [34] evolutionary model with $k = 9$, $m = 15000$, coverage $g \in \{0.01\times, 0.05\times, 0.01\times\}$. We found that PHERTILIZER still outperformed BASELINE+SCITE and SBMCLONE+SCITE, having maintained high scores on all performance metrics (Fig. S11) with the exception of cell APR for the lowest coverage (Fig. S11a).

In summary, we conclude that given the high level of accuracy obtained by PHERTILIZER 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.

4.2 High-grade serous ovarian cancer patient

Utilizing a grid search over input parameters(Appendix B.2.1), we ran PHERTILIZER on $n = 890$ DLP+ sequenced cells from three clonally-related cancer cells lines sourced from the same high-grade serous ovarian cancer patient [6]. We used the variant and total read counts $\mathbf{A}, \mathbf{D}$ for $m = 14{,}068$ SNVs and derived binned read counts $\mathbf{R}$ for $b = 6{,}207$ bins (bin width of 500 KB) from data reported by Laks et al. [6]. The average sequencing coverage for these data was $0.25\times$. Utilizing an approach similar to the BASELINE method described above, Laks et al. [6] 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. 4a). We annotated inferred trees with mutations in cancer-related genes in the cBioPortal [35,36] and Cancer Gene Census (CGC) [37] from COSMIC v97 (see Appendix B.2.2).

As shown in Fig. 4b, PHERTILIZER inferred a clonal tree with 13 nodes and 8 clones. We found that PHERTILIZER's tree closely aligned with the Laks et al. [6] tree, with both approaches correctly identifying three major clades corresponding to the three distinct cell lines of origin (Fig. 4c). Additionally, driver genes *TP53*, *SUGCT*, and *MYH9* are identified as clonal in both the Laks et.al [6] inferred tree (Fig 4a) and the PHERTILIZER clonal tree (Fig 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 $\mathrm{CMB}(i, M) = \sum_{q \in M} \mathbf{1}\{a_{iq} > 0\} / \sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}$. In words, $\mathrm{CMB}(i, M)$ 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 $\mathrm{CMB}(i, M_j)$ to be high, although the value will depend on copy number. By contrast, for cells placed outside of clade j, we expect $\mathrm{CMB}(i, M_j)$ to be low. More details on CMB are provided in Appendix B.2.3 and Fig. S13.

Fig. 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. [6] and PHERTILIZER, respectively. For cells placed outside of a specified clade, the reported tree by Laks et al. [6] 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

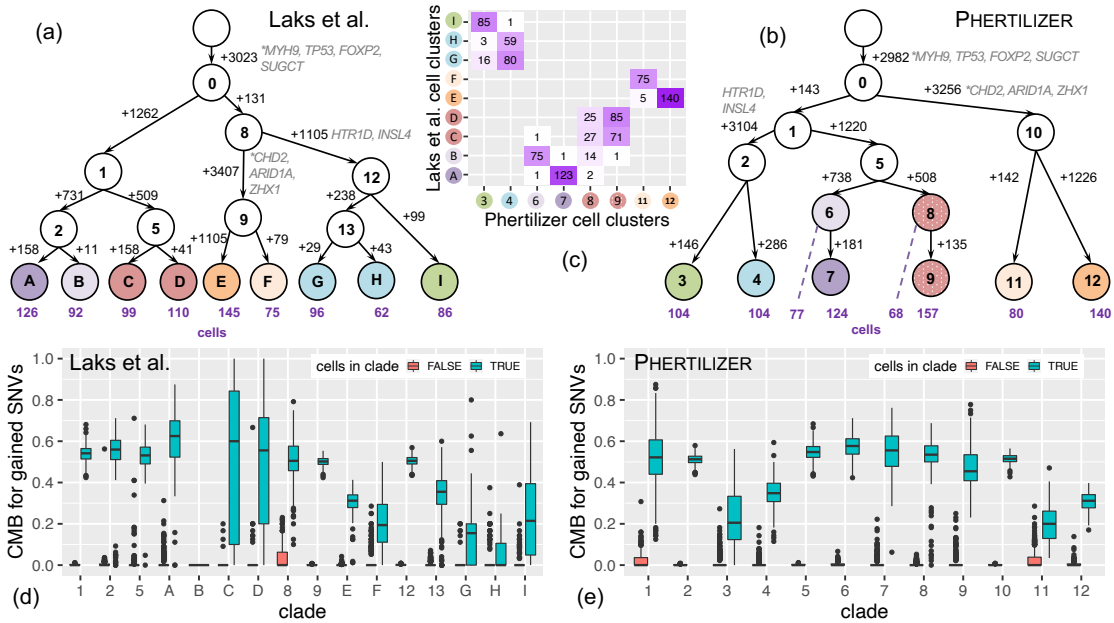


Figure 4: **PHERTILIZER improves upon the SNV phylogeny previously inferred by Laks et al. [6]** (a) The clonal tree inferred by Laks et al. [6] with edges labeled by SNV gains and cell numbers shown below leaf nodes. (b) The clonal tree inferred by PHERTILIZER with edges labeled by SNV gains and cell 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. [6] cell placement and PHERTILIZER 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. [6] and PHERTILIZER clonal tree.

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. [6] 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. S14a). In comparison, PHERTILIZER uses both copy number and SNV signal, resulting in the Laks et al. [6] 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. 4d) with PHERTILIZER's inferred clade 4 (Fig. 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. [6] 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. [6] 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. S14a), 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. S14b),

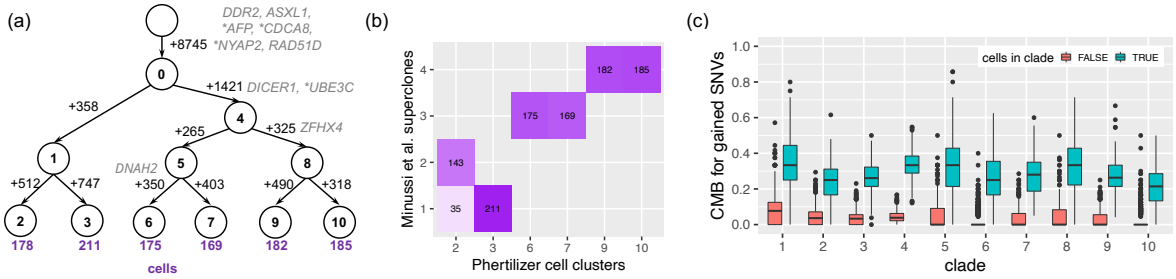


Figure 5: **PHERTILIZER infers clonal tree for breast cancer tumor TN1.** (a) The tree inferred by PHERTILIZER with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs ('*': stop-gain variant). (b) A mapping between PHERTILIZER's cell clusters and the Minussi et al. [7] superclones. (c) The cell mutational burden (CMB) comparison between cells within (blue) and outside of (red) each clade in the inferred clonal tree.

we also observed a clear separation between the CMB distributions for cells placed within and outside of clades 8 and 9 (Fig. 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. [6] 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. [6], 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.

4.3 Eight triple negative breast tumors

We applied PHERTILIZER to eight triple negative breast tumors sequenced via ACT [7], labeled TN1 to TN8. After dimensionality reduction of normalized and GC-bias corrected binned read counts $\mathbf{R}$, Minussi et al. [7] 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 [38]. See Appendix B.2.5 for further details on data processing. Table S1 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. [7] superclones and subclones, as well as the number of inferred clones by SBMCLONE. 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. [7] 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. S16) provides no evidence in support of these trees. SBMCLONE 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) and TN2 (Fig. S17).

For tumor TN1, PHERTILIZER inferred a branching tree with 11 nodes and 6 clones (Fig. 5a). We also

identified a subclonal missense SNV in driver gene *DICER1* associated with tumorigenesis and poor prognosis [39,40]. We noted good concordance between the Minussi et al. [7] superclones and the PHERTILIZER cell clusters, with the exception of 35 cells that appear as outliers in superclone 1 (Fig. 5b, Fig. S15). 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. [7] using bulk whole exome sequencing. We used CMB to assess the performance of SNV and cell placement (Fig. 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 [7] superclone 4 (Fig. S15). 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 (Section 4.2) is expected due to the drop in sequencing coverage 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. S17). Two of PHERTILIZER's cell clusters directly agreed with Minussi et al. [7] 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. S18). 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. S19).

5 Discussion

Ultra-low coverage scDNA-seq has greatly enhanced our ability to study tumor evolution from a copy number perspective [7,17]. 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. Third, the infinite sites model used in this work is often violated due to copy-number deletions. While we demonstrate robustness to such violations, a future direction is to use the Dollo evolutionary model [14,34]. Fourth, our model lacks an explicit placement of CNA events on the tree. Tree reconciliation methods, such as PACTION [41], can now be applied to integrate an SNV clonal tree generated by PHERTILIZER and a CNA tree to obtain a joint tree. Finally, beyond SNVs and CNAs, we plan to support structural variants and integrate other omics modalities such as methylation and transcription [42].

References

[1] Nowell, P. C. The clonal evolution of tumor cell populations: Acquired genetic lability permits stepwise selection of variant sublines and underlies tumor progression. *Science* **194**, 23–28 (1976).

[2] Morita, K. *et al.* Clonal evolution of acute myeloid leukemia revealed by high-throughput single-cell genomics. *Nature Communications* **11**, 1–17 (2020).

[3] Baslan, T. *et al.* Novel insights into breast cancer copy number genetic heterogeneity revealed by single-cell genome sequencing. *eLife* **9**, e51480 (2020).

[4] Kim, C. *et al.* Chemoresistance evolution in triple-negative breast cancer delineated by single-cell sequencing. *Cell* **173**, 879–893 (2018).

[5] Miles, L. A. *et al.* Single-cell mutation analysis of clonal evolution in myeloid malignancies. *Nature* **587**, 477–482 (2020).

[6] Laks, E. *et al.* Clonal decomposition and dna replication states defined by scaled single-cell genome sequencing. *Cell* **179**, 1207–1221 (2019).

[7] Minussi, D. C. *et al.* Breast tumours maintain a reservoir of subclonal diversity during expansion. *Nature* **592**, 302–308 (2021).

[8] Zahn, H. *et al.* Scalable whole-genome single-cell library preparation without preamplification. *Nature Methods* **14**, 167–173 (2017).

[9] Pellegrino, M. *et al.* High-throughput single-cell DNA sequencing of acute myeloid leukemia tumors with droplet microfluidics. *Genome Research* **28**, 1345–1352 (2018).

[10] Fu, X., Lei, H., Tao, Y. & Schwartz, R. Reconstructing tumor clonal lineage trees incorporating single-nucleotide variants, copy number alterations and structural variations. *Bioinformatics* **38**, i125–i133 (2022).

[11] Kannan, J., Mathews, L., Wu, Z., Young, N. S. & Gao, S. 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* **23**, 1–17 (2022).

[12] Jahn, K., Kuipers, J. & Beerenwinkel, N. Tree inference for single-cell data. *Genome Biology* **17**, 1–17 (2016).

[13] Malikic, S. *et al.* PhISCS: a combinatorial approach for subperfect tumor phylogeny reconstruction via integrative use of single-cell and bulk sequencing data. *Genome Research* **29**, 1860–1877 (2019).

[14] El-Kebir, M. SPhyR: tumor phylogeny estimation from single-cell sequencing data under loss and error. *Bioinformatics* **34**, i671–i679 (2018).

[15] Zafar, H., Navin, N., Chen, K. & Nakhleh, L. SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data. *Genome Research* **29**, 1847–1859 (2019).

[16] Roth, A. *et al.* Clonal genotype and population structure inference from single-cell tumor sequencing. *Nature Methods* **13**, 573–576 (2016).

[17] Zaccaria, S. & Raphael, B. J. Characterizing allele-and haplotype-specific copy numbers in single cells with chisel. *Nature Biotechnology* **39**, 207–214 (2021).

[18] Markowska, M. *et al.* CONET: Copy number event tree model of evolutionary tumor history for single-cell data. *Genome Biology* **23**, 1–35 (2022).

[19] Liu, Y., Edrisi, M., Ogilvie, H. & Nakhleh, L. NestedBD: Bayesian inference of phylogenetic trees from single-cell DNA copy number profile data under a birth-death model. *bioRxiv* (2022).

[20] Wang, F. *et al.* MEDALT: single-cell copy number lineage tracing enabling gene discovery. *Genome Biology* **22**, 1–22 (2021).

[21] Kaufmann, T. L. *et al.* MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution. *Genome biology* **23**, 241 (2022).

[22] Kozlov, A., Alves, J. M., Stamatakis, A. & Posada, D. CellPhy: accurate and fast probabilistic inference of single-cell phylogenies from scDNA-seq data. *Genome biology* **23**, 1–30 (2022).

[23] Kang, S. *et al.* SIEVE: joint inference of single-nucleotide variants and cell phylogeny from single-cell dna sequencing data. *Genome Biology* **23**, 248 (2022).

[24] Chen, K., Moravec, J. C., Gavryushkin, A., Welch, D. & Drummond, A. J. Accounting for errors in data improves divergence time estimates in single-cell cancer evolution. *Molecular biology and evolution* **39**, msac143 (2022).

[25] Milite, S., Bergamin, R., Patruno, L., Calonaci, N. & Caravagna, G. A Bayesian method to cluster single-cell RNA sequencing data using copy number alterations. *Bioinformatics* **38**, 2512–2518 (2022).

[26] Zhou, Z., Xu, B., Minn, A. & Zhang, N. R. DENDRO: genetic heterogeneity profiling and subclone detection by single-cell RNA sequencing. *Genome biology* **21**, 1–15 (2020).

[27] Satas, G., Zaccaria, S., Mon, G. & Raphael, B. J. SCARLET: single-cell tumor phylogeny inference with copy-number constrained mutation losses. *Cell Systems* **10**, 323–332 (2020).

[28] Myers, M. A., Zaccaria, S. & Raphael, B. J. Identifying tumor clones in sparse single-cell mutation data. *Bioinformatics* **36**, i186–i193 (2020).

[29] Rozhoňová, H. *et al.* SECEDO: SNV-based subclone detection using ultra-low coverage single-cell DNA sequencing. *Bioinformatics* **38**, 4293–4300 (2022).

[30] Kimura, M. The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. *Genetics* **61**, 893 (1969).

[31] Davis, A., Gao, R. & Navin, N. Tumor evolution: Linear, branching, neutral or punctuated? *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer* **1867**, 151–161 (2017).

[32] Shi, J. & Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* **22**, 888–905 (2000).

[33] El-Kebir, M., Oesper, L., Acheson-Field, H. & Raphael, B. J. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. *Bioinformatics* **31**, i62–i70 (2015).

[34] Dollo, L. Les lois de l'évolution. *Bulletin de la Société belge de géologie, de paléontologie et d'hydrologie* **7**, 164–166 (1893).

[35] Cerami, E. *et al.* The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer Discovery* **2**, 401–404 (2012).

[36] Gao, J. *et al.* Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Science Signaling* **6**, pl1–pl1 (2013).

[37] Sondka, Z. *et al.* The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers. *Nature Reviews Cancer* **18**, 696–705 (2018).

[38] Van der Auwera, G. A. & O'Connor, B. D. *Genomics in the cloud: using Docker, GATK, and WDL in Terra* (O'Reilly Media, 2020).

[39] Consortium, A. P. G. *et al.* AACR project GENIE: powering precision medicine through an international consortium. *Cancer discovery* **7**, 818–831 (2017).

[40] Heravi-Moussavi, A. *et al.* Recurrent somatic DICER1 mutations in nonepithelial ovarian cancers. *New England Journal of Medicine* **366**, 234–242 (2012).

[41] Sashittal, P., Zaccaria, S. & El-Kebir, M. Parsimonious clone tree integration in cancer. *Algorithms for Molecular Biology* **17**, 1–14 (2022).

[42] El-Kebir, M., Morris, Q., Oesper, L. & Sahinalp, S. C. Emerging topics in cancer evolution. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2022*, 397–401 (World Scientific, 2021).

[43] McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[44] Weber, L. L., Sashittal, P. & El-Kebir, M. doubletD: detecting doublets in single-cell DNA sequencing data. *Bioinformatics* **37**, i214–i221 (2021).

[45] De Bourcy, C. F. *et al.* A quantitative comparison of single-cell whole genome amplification methods. *PloS one* **9**, e105585 (2014).

[46] Aggarwal, C. C., Hinneburg, A. & Keim, D. A. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, 420–434 (Springer, 2001).

[47] Kriegel, H.-P., Kröger, P. & Zimek, A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data* **3**, 1–58 (2009).

[48] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[49] Damle, A., Minden, V. & Ying, L. Simple, direct and efficient multi-way spectral clustering. *Information and Inference: A Journal of the IMA* **8**, 181–203 (2019).

[50] Ankerst, M., Breunig, M. M., Kriegel, H.-P. & Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* **28**, 49–60 (1999).

[51] McLaren, W. *et al.* The Ensembl variant effect predictor. *Genome biology* **17**, 1–14 (2016).

[52] Vaser, R., Adusumalli, S., Leng, S. N., Sikic, M. & Ng, P. C. Sift missense predictions for genomes. *Nature protocols* **11**, 1–9 (2016).

[53] Adzhubei, I., Jordan, D. M. & Sunyaev, S. R. Predicting functional effect of human missense mutations using PolyPhen-2. *Current protocols in human genetics* **76**, 7–20 (2013).

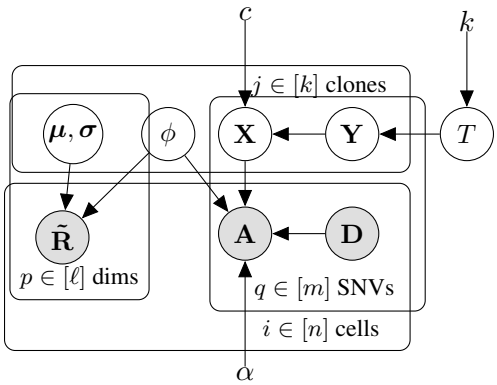


Figure S1: **Plate diagram.** The observed data consists of variant and total read counts $\mathbf{A}, \mathbf{D}$ and binned read counts $\tilde{\mathbf{R}}$, generated from an unobserved cell clustering ϕ and clonal tree T of k clones with clonal genotypes $\mathbf{Y}$. Given hyperparameters $\theta = (c, \alpha)$ and observed data $(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}})$, we seek $(T, \mathbf{Y}, \phi)$ with maximum posterior probability $P(T, \mathbf{Y}, \phi, \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta)$, marginalizing over latent variables $\mathbf{X}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

A Supplementary methods

A.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

$$r'_{ip} = \frac{r_{ip}}{\sum_{p'=1}^{b} r_{ip'}}. \tag{1}$$

Next, we use UMAP [43] 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 $\ell = 2$.

A.2 Generative model

Our generative model (Fig. S1) 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 $P(T \mid k)$. We generate clonal genotypes $\mathbf{Y}$ for tree T under the infinite sites model with $P(\mathbf{Y} \mid T)$ uniformly distributed. We model the cell clustering $\phi : [k] \to 2^{[n]}$ 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.

The generative model has three remaining components: the (i) latent variant allele frequency (VAF) model, (ii) the variant read count model and (iii) the binned read count embedding model. First, the latent VAFs $\mathbf{X} \in \mathbb{Q}^{k \times m}$ are dependent on the clonal genotypes $\mathbf{Y}$. Thus, whenever clonal genotype $y_{jq} = 0$ for SNV q of clone j, the latent VAF $x_{jq} = 0$. Otherwise, x_{jq} is the ratio of chromosomal copies that harbor SNV q to the total number of chromosomal copies of clone j. Appendix A.2.1 contains the precise definition of $P(\mathbf{X} \mid \mathbf{Y}, \theta)$. Second, given latent VAFs $\mathbf{X}$, total read counts $\mathbf{D}$, cell clustering ϕ and the probability α of

misreading a single nucleotide during sequencing, we model variant read counts with a binomial distribution that accounts for sequencing errors. Appendix A.2.2 provides the derivation of $P(\mathbf{A} \mid \mathbf{D}, \mathbf{X}, \phi, \theta)$. Third, 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 $p \in [\ell]$. Appendix A.2.3 contains the definition of $P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma})$.

This model 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 (Fig. S1). Given hyperparameters $\theta = (c, \alpha)$ and observed data $(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}})$, we seek to identify latent variables $(T, \mathbf{Y}, \phi)$ with maximum posterior probability $P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta)$. We show in Appendix A.2.4 that this can be approximated as $P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma})$ taking $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ as the maximum *a posteriori* estimates.

A.2.1 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

$$S = \{s/t \mid 1 \leq s \leq t \leq c\}. \tag{2}$$

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

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

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 all possible combinations of $\bar{a}_{jq}$ and $\bar{b}_{jq}$ are equally likely. Using independence among clones $[k]$ and SNV loci $[m]$ when given clonal genotypes $\mathbf{Y}$, we have that

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

A.2.2 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 [44, 45], the lack of preamplification in the latest generation of ultra-low scDNA-seq technology restricts errors to base calling errors that occur during sequencing [6, 7]. Let α be the probability of misreading a single nucleotide during sequencing, next we derive $P(\mathbf{A} \mid \mathbf{D}, \mathbf{X}, \phi, \theta)$.

$$P(a_{iq} \mid d_{iq}, \mathbf{x}_{\cdot q}, \phi, \theta) = \text{Binom}\left(a_{iq};\, d_{iq},\, x_{jq}(1 - \alpha) + \frac{(1 - x_{jq})\alpha}{3}\right), \tag{5}$$

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

$$P(a_{iq} \mid d_{iq}, \mathbf{x}_{\cdot q}, \phi, \theta) = \prod_{j=1}^{k} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta)^{\mathbf{1}\{i \in \phi(j)\}} \tag{6}$$

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

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

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

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

A.2.3 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 [46, 47] 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

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

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

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

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

A.2.4 Posterior probability

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

$$\arg\max_{T, \mathbf{Y}, \phi} P(T, \mathbf{Y}, \phi \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta). \tag{13}$$

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

$$\underset{T,\mathbf{Y},\phi}{\arg\max} \int_{\boldsymbol{\mu}} \int_{\boldsymbol{\sigma}} P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta)\, d\boldsymbol{\mu}\ d\boldsymbol{\sigma}. \tag{14}$$

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

$$\underset{T,\mathbf{Y},\phi,\boldsymbol{\mu},\boldsymbol{\sigma}}{\arg\max}\ P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta). \tag{15}$$

We begin by applying Bayes' rule, obtaining

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

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

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

Using uniformity of $P(T \mid \theta)$, $P(\mathbf{Y} \mid T, \theta)$, $P(\phi \mid \theta)$, $P(\boldsymbol{\mu}, \boldsymbol{\sigma} \mid \theta)$ and dropping $P(\mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}} \mid \theta)$ from the denominator, we obtain

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

Using the law of conditional probability, we get

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

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

$$\begin{aligned} P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) &\propto P(\mathbf{A}, \tilde{\mathbf{R}} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) && (20)\\ &= P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) P(\tilde{\mathbf{R}} \mid \mathbf{D}, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta) && (21)\\ &= P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}). && (22) \end{aligned}$$

We now focus on $P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta)$, which equals

$$\begin{aligned} P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) &= \prod_{j=1}^{k} \prod_{i=1}^{n} \prod_{q=1}^{m} P(a_{iq} \mid d_{iq}, y_{jq}, \phi, \theta)^{\mathbf{1}\{i \in \phi(j)\}} && (23)\\ &= \prod_{j=1}^{k} \prod_{i=1}^{n} \prod_{q=1}^{m} \left[\sum_{x_{jq} \in S \cup \{0\}} P(a_{iq}, x_{jq} \mid d_{iq}, y_{jq}, \phi, \theta) \right]^{\mathbf{1}\{i \in \phi(j)\}} && (24)\\ &= \prod_{j=1}^{k} \prod_{i=1}^{n} \prod_{q=1}^{m} \left[\sum_{x_{jq} \in S \cup \{0\}} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta) P(x_{jq} \mid y_{jq}, \theta) \right]^{\mathbf{1}\{i \in \phi(j)\}}. && (25) \end{aligned}$$

Bringing everything together, we get

$$
\begin{aligned}
&P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) && (26)\\
&\propto P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) && (27)\\
&= \left(\prod_{j=1}^{k} \prod_{i=1}^{n} \prod_{q=1}^{m} \left[\sum_{x_{jq} \in S \cup \{0\}} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta) P(x_{jq} \mid y_{jq}, \theta) \right]^{\mathbf{1}\{i \in \phi(j)\}} \right) \\
&\quad \cdot \left(\prod_{j=1}^{k} \prod_{i=1}^{n} \prod_{p=1}^{\ell} \mathcal{N}(\mu_{jp}, \sigma_{jp}^2)^{\mathbf{1}\{i \in \phi(j)\}} \right). && (28)
\end{aligned}
$$

We now take the logarithm, yielding

$$
\begin{aligned}
&P(T, \mathbf{Y}, \phi, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid \mathbf{A}, \mathbf{D}, \tilde{\mathbf{R}}, \theta) && (29)\\
&\propto \log P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) && (30)\\
&= \log P(\mathbf{A} \mid \mathbf{D}, \mathbf{Y}, \phi, \theta) + \log P(\tilde{\mathbf{R}} \mid \phi, \boldsymbol{\mu}, \boldsymbol{\sigma}) && (31)\\
&= \sum_{j=1}^{k} \sum_{i=1}^{n} \sum_{q=1}^{m} \mathbf{1}\{i \in \phi(j)\} \log \sum_{x_{jq} \in S \cup \{0\}} P(a_{iq} \mid d_{iq}, x_{jq}, \phi, \theta) P(x_{jq} \mid y_{jq}, \theta) \\
&\quad + \sum_{j=1}^{k} \sum_{i=1}^{n} \sum_{p=1}^{\ell} \mathbf{1}\{i \in \phi(j)\} \log \mathcal{N}(\mu_{jp}, \sigma_{jp}^2). && (32)
\end{aligned}
$$

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.

A.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

$$
O(N, M) = \sum_{i \in N} \sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}. \tag{33}
$$

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 4. Given parameter $t \in \mathbb{N}$, a clone comprised of cells N and newly introduced SNVs M is *detectable* provided (i) $\operatorname{median}\{O(i, M) \mid i \in N\} \geq t$ and (ii) $\operatorname{median}\{O(N, q) \mid q \in M\} \geq t$.

We refer to Fig. S2 for an example.

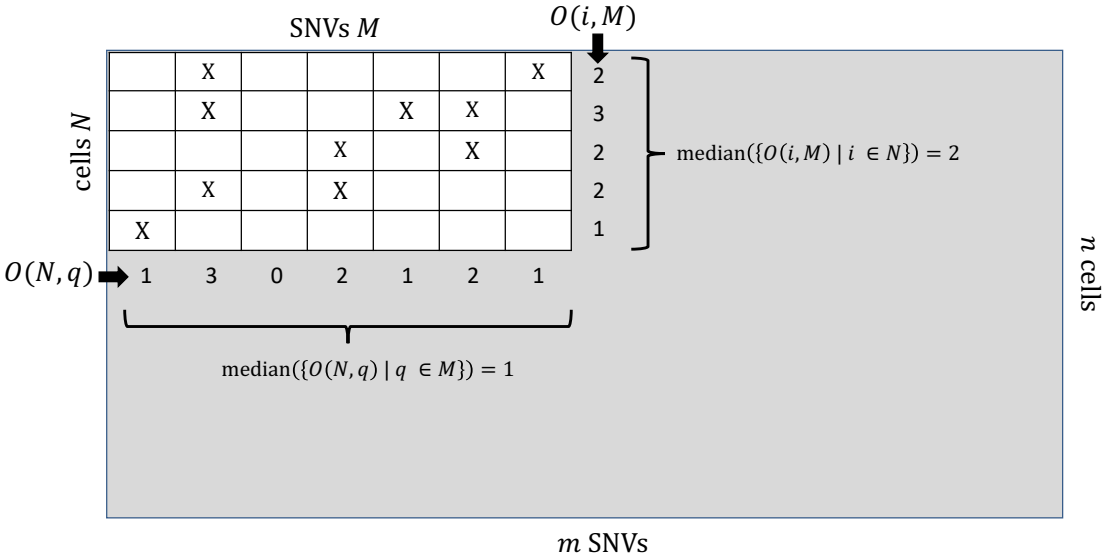


Figure S2: Visual depiction of the detectability of a clone with subset $N \subseteq [n]$ of cells and subset $M \subseteq [m]$ of SNVs. Here, 'X' indicates the presence of a mapped read for SNV locus q in cell i. Observations $O(N, q)$ is the marginal sum of observations over cells N for each SNV $q \in M$. 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 4).

A.4 PHERTILIZER

At the heart of PHERTILIZER is the definition of three elementary tree operations: `Branching`, `Linear` and `Identity`. The input to each 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. 2b). 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. 2c). 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. 2d).

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 (i) the `Linear` elementary operation (Appendix A.4.1) (ii) the `Branching` elementary operation (Appendix A.4.2), (iii) a number of regularization steps to avoid overfitting the data (Appendix A.4.3) and (iv) a post-processing phase to improve clonal genotyping and cell clustering in the inferred clonal tree (Appendix A.4.4).

A.4.1 `Linear`

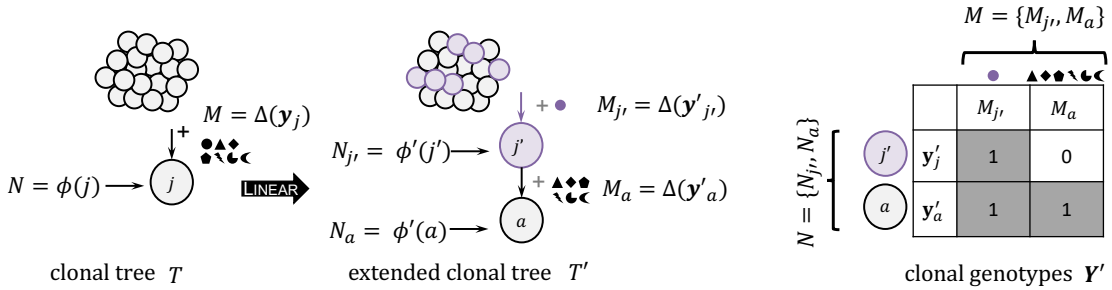


Figure S3: A graphical depiction of a `Linear` elementary tree operation. The cells $N = \phi(j)$ and the SNVs $M = \Delta(\mathbf{y}_j)$ associated with clone j of tree T are partitioned into $N = \{N_{j'}, N_a\}$ and $M = \{M_{j'}, M_a\}$, respectively. These partitions are then used to update clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' associated with extended tree T'.

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. S3) 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 [32]. 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 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. S3). 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 $\{M_{j'}, M_a\}$. Gven the SNV partition $\{M_{j'}, M_a\}$, we use the normalized cut algorithm [32] 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 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 defind as

$$w_{ii'} = e^{\frac{-\|f_i - f_{i'}\|_2^2}{s_f}} \cdot \begin{cases} e^{\frac{-\|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2}{s_c}}, & \text{if } \|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}_{i'}}\|_2^2 < \tau, \\ 0, & \text{otherwise,} \end{cases} \tag{34}$$

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 [32], 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` [48] spectral clustering implementation as it solves the normalized cut problem when the number of clusters is 2. We choose the `cluster-qr` [49] 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 $\mathbf{a}'_{\cdot q}$ and $\mathbf{d}'_{\cdot q}$ 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 $\log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N_{j'}, q \in M', \theta)$ where $M' \in \{M_{j'}, M_a\}$. These probabilities are computed as follows.

$$\log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N_{j'}, q \in M_{j'}, \theta) = \sum_{i \in N_{j'}} \log P(a'_{iq} \mid d'_{iq}, y_{j'q} = 1, \theta), \tag{35}$$

$$\log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N_{j'}, q \in M_a, \theta) = \sum_{i \in N_{j'}} \log P(a'_{iq} \mid d'_{iq}, y_{j'q} = 0, \theta). \tag{36}$$

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 only an `Identity` operation can be performed. See Appendix A.4.3 for details on this criteria. The recursion step is necessary because the normalized cut algorithm is a recursive partitioning algorithm. Unlike `Branching` (Appendix A.4.2), 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.

A.4.2 `Branching`

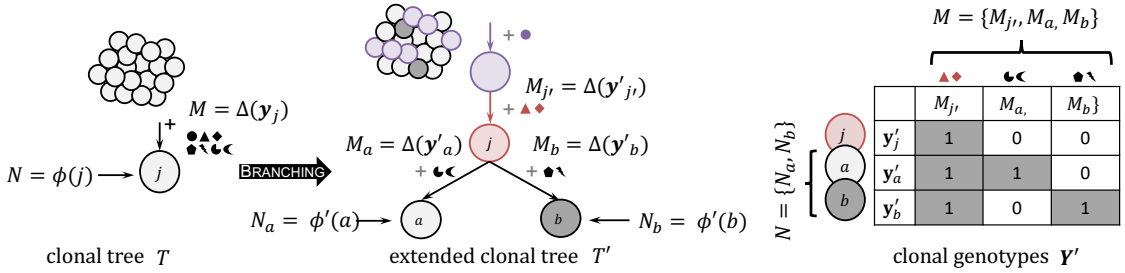


Figure S4: A graphical depiction of a `Branching` elementary tree operation. The cells $N = \phi(j)$ and the SNVs $M = \Delta(\mathbf{y}_j)$ associated with clone j of tree T are partitioned into $N = \{N_a, N_b\}$ and $M = \{M_{j'}, M_a, M_b\}$, respectively. These partitions are then used to update clonal genotypes $\mathbf{Y}'$ and cell clustering ϕ' associated with extended tree T'.

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. S4) 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 [32]. 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 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. S4). 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 $\{M_{j'}, M_a, M_b\}$. Gven the SNV partition $\{M_{j'}, M_a, M_b\}$, we use the normalized cut algorithm [32] to find a two part cell partition $\{N_a, N_b\}$ 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 is set as $\mathbf{f}_i = [f_{ia}, f_{ib}]^\top$, where

$$f_{ia} = \frac{\sum_{q \in M_a} \mathbf{1}\{a_{iq} > 0\}}{\sum_{q \in M_a} \mathbf{1}\{d_{iq} > 0\}}. \tag{37}$$

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.

$$w_{ii'} = e^{\frac{-\|\mathbf{f}_i - \mathbf{f}_{i'}\|_2^2}{s_f}} \cdot \begin{cases} e^{\frac{-\|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_{i'}\|_2^2}{s_c}}, & \text{if } \|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}_{i'}}\|_2^2 < \tau, \\ 0, & \text{otherwise,} \end{cases} \tag{38}$$

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 [32], 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` [48] spectral clustering implementation as it solves the normalized cut problem when the number of clusters is 2. We choose the `cluster-qr` [49] 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 $\mathbf{a}'_{\cdot q}$ and $\mathbf{d}'_{\cdot q}$ 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 $\log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N = \{N_a, N_b\}, q \in M', \theta)$ where $M' \in \{M_{j'}, M_a, M_b\}$. These probabilities are computed as follows.

$$\log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N = \{N_a, N_b\}, q \in M_{j'}, \theta) = \sum_{i \in N} \log P(a'_{iq} \mid d'_{iq}, y_{j'q} = 1, \theta), \tag{39}$$

$$\begin{aligned} \log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N = \{N_a, N_b\}, q \in M_a, \theta) = &\sum_{i \in N_a} \log P(a'_{iq} \mid d'_{iq}, y_{aq} = 1, \theta) \\ &+ \sum_{i \in N_b} \log P(a'_{iq} \mid d'_{iq}, y_{bq} = 0, \theta), \end{aligned} \tag{40}$$

$$\begin{aligned} \log P(\mathbf{a}'_{\cdot q} \mid \mathbf{d}'_{\cdot q}, N = \{N_a, N_b\}, q \in M_b, \theta) = &\sum_{i \in N_b} \log P(a'_{iq} \mid d'_{iq}, y_{bq} = 1, \theta) \\ &+ \sum_{i \in N_a} \log P(a'_{iq} \mid d'_{iq}, y_{aq} = 0, \theta). \end{aligned} \tag{41}$$

In other words, we compare the probabilities of assigning SNV q to either $M_{j'}$, M_a or M_a. 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.

A.4.3 `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 A.4.2) and nodes v_j, v_a for `Linear` (Appendix A.4.1) are detectable (Appendix A.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

$$\mathrm{CMB}(i, M) = \frac{\sum_{q \in M} \mathbf{1}\{a_{iq} > 0\}}{\sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}}. \tag{42}$$

More details and intuition on the CMB is provided in Appendix B.2.3.

We define a quality check $\mathrm{qc}(N, M)$ as $\mathrm{median}\{\mathrm{CMB}(i, M) \mid i \in [N]\}$. 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 A.4.1) we expect $\mathrm{qc}(N, M_{j'})$ 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 $\mathrm{qc}(N_{j'}, M_a)$ 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 A.4.2).

$$\begin{aligned}
&\mathrm{qc}(N, M_{j'}) \geq 0.15, \\
&\mathrm{qc}(N_a, M_b) \leq 0.05, \\
&\mathrm{qc}(N_b, M_a)) \leq 0.05.
\end{aligned}$$

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.

A.4.4 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.

$$\begin{aligned}\log P(\mathbf{a}_{\cdot q} \mid \mathbf{d}_{\cdot q}, \phi, q \in \Delta(\mathbf{y}_j), T, \theta) = & \sum_{v_{j'} \in \mathrm{desc}(v_j) \cup \{v_j\}} \sum_{i \in \phi(j')} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 1, \theta) \\ & + \sum_{v_{j'} \in \mathrm{anc}(v_j)} \sum_{i \in \phi(j')} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 0, \theta),\end{aligned} \tag{43}$$

where $\mathrm{desc}(v_j)$ is the set of descendant nodes and $\mathrm{anc}(v_j)$ 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.

$$\begin{aligned}\log P(\mathbf{a}_{\cdot q} \mid \mathbf{d}_{\cdot q}, \mathbf{Y}, i \in \phi(j), T, \theta) = & \sum_{v_{j'} \in \mathrm{anc}(v_j) \cup \{v_j\}} \sum_{q \in \Delta(\mathbf{y}_{j'})} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 1, \theta) \\ & + \sum_{v_{j'} \in \mathrm{desc}(v_j)} \sum_{q \in \Delta(\mathbf{y}_{j'})} \log P(a_{iq} \mid d_{iq}, y_{j'q} = 0, \theta).\end{aligned} \tag{44}$$

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 alele frequency (BVAF), which is as follows for a given SNV q and node v_j

$$\mathrm{BVAF}(q, v_j) = \frac{\sum_{v_{j'} \in \mathrm{desc}(v_j) \cup \{v_j\}} \sum_{i \in \phi(j')} \mathbf{1}\{a_{iq} > 0\}}{\sum_{v_{j'} \in \mathrm{desc}(v_j) \cup \{v_j\}} \sum_{i \in \phi(j')} \mathbf{1}\{d_{iq} > 0\}}. \tag{45}$$

In words, $\mathrm{BVAF}(q, v_j)$ is the fraction of cells in clade v_j with mapped variant reads at SNV q. We compute $\mathrm{BVAF}(q, v_j)$ 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 $\mathrm{BVAF}(q, v_j) \leq t'$ 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.

B Supplementary results

B.1 Simulation study

Below we provide pertinent details related to our simulation study and supplemental figures.

- Appendix B.1.1 provides additional details for the BASELINE comparison method
- Appendix B.1.2 provides additional details on our simulation study design
- Appendix B.1.3 provides the specification of runtime parameters for methods analyzed in the simulation study
- Appendix B.1.4 provides additional details on the performance metrics used to evaluation the results of our simulation study
- Appendix B.1.5 contains supplemental results figures for our simulation study

B.1.1 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 A.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 [50] 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. S5 shows good concordance of clonal genotypes using this approach with that of the inferred clonal genotypes on a DLP+ dataset [6].

B.1.2 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 heterzygous diploid at every locus in the genome. We also varied the SNV evolutionary model, using either a Dollo (SNV loss, [34]) or an infinite sites model (without SNV loss, [30]). 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.

B.1.3 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 \
```

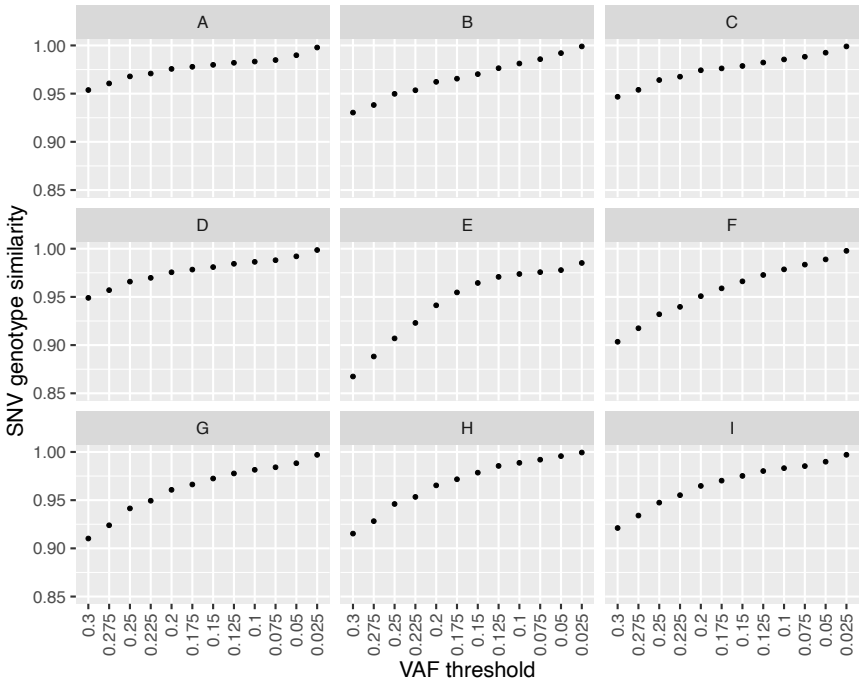


Figure S5: Genotype similarity between inferred genotypes per cell cluster using the BASELINE method and varying the variant allele frequency (VAF) threshold and inferred clonal genotypes per cell cluster by Laks et al. [6]

```
    --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}
```

B.1.4 Performance metrics

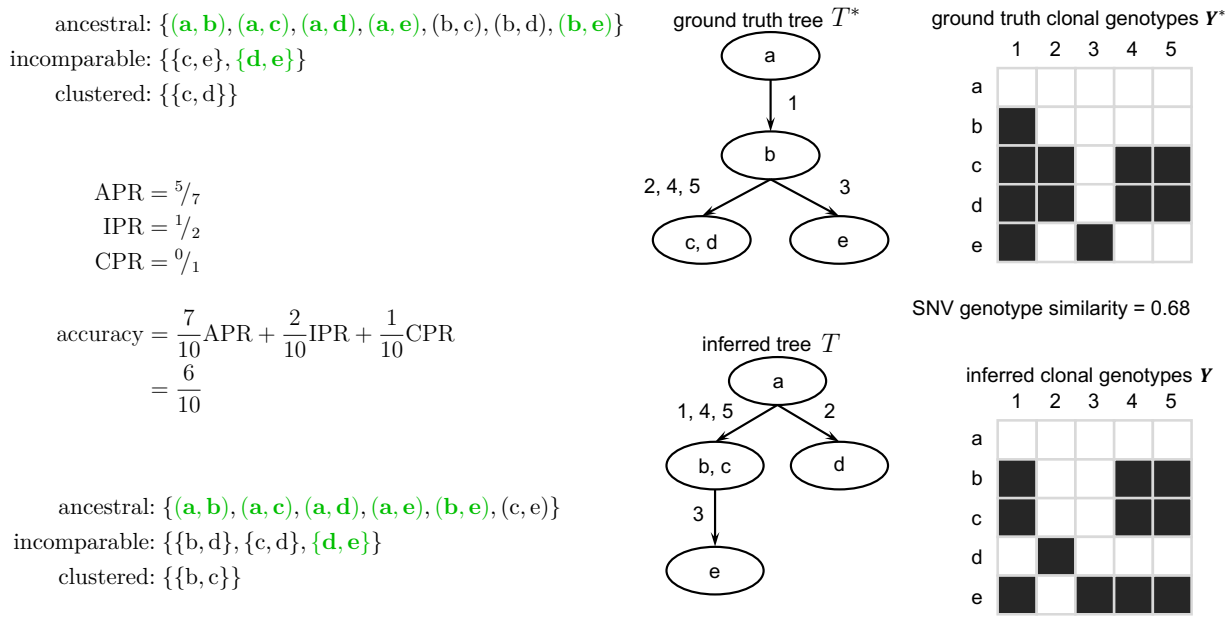


Figure S6: Example for ancestral pair recall (APR), clustered pair recall (CPR), incomparable pair recall (IPR), and accuracy for cells. To the right of the trees, we show the ground truth and inferred genotypes $\mathbf{Y}^*$ and $\mathbf{Y}$ projected down to individual cells. Main Text Fig. 3a provides an example of APR, CPR, IPR and accuracy metrics for SNVs.

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 [33], 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.

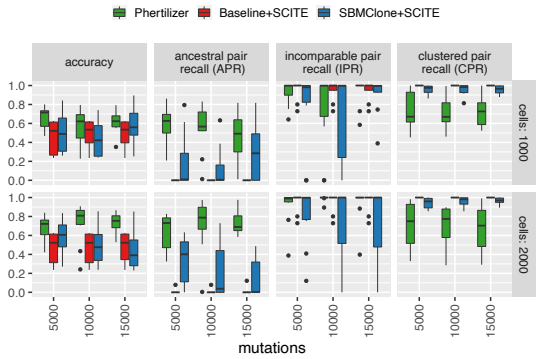
APR, IPR, CPR. For any two SNVs $q \neq q'$ there are three possible placements in the tree: (i) *ancestral*: q is gained on a node that is distinct and ancestral to the node where q' is gained; (ii) *clustered*: q and q' are both gained on the same node; and (iii) *incomparable*: q and q' are gained 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. 3a and Fig. S6 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

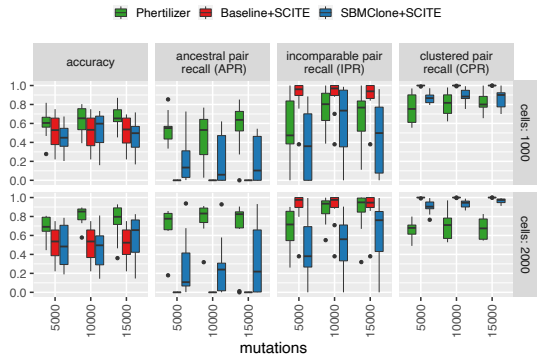
$1/(nm)\sum_{i=1}^{n}\sum_{q=1}^{m}\mathbb{1}(y_{\bar{\phi}^*(i),j} = y^*_{\bar{\phi}(i),q})$ where $\bar{\phi}(i)$ ($\bar{\phi}^*(i)$) is the unique clone j such that $i \in \phi(j)$ ($i \in \phi^*(j)$), given an inferred genotype $\hat{\mathbf{y}}_i \in \{0,1\}^m$ and a ground truth genotype $\mathbf{y}^*_i$. Thus, a genotype similarity of 1 implies that every sequenced cell was correctly genotyped. Fig. S6 above provides a visualization for the computation of this metric.

B.1.5 Supplemental simulation study figures

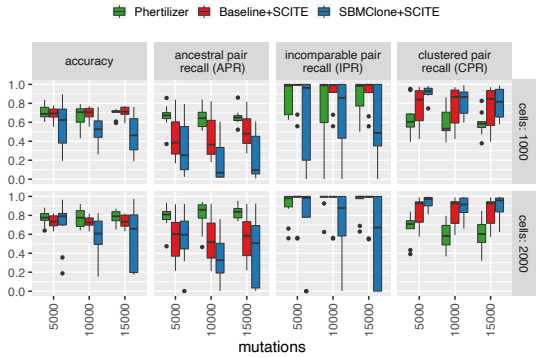
- Fig. S7 shows supplemental simulation study results for coverage $0.01\times$
- Fig. S8 shows supplemental simulation study results for coverage $0.05\times$
- Fig. S9 shows supplemental simulation study results for coverage $0.1\times$
- Fig. S10 shows heterozygous diploid simulation study results
- Fig. S11 shows Dollo evolutionary model simulation study results
- Fig. S12 shows a running time comparison for the simulation study



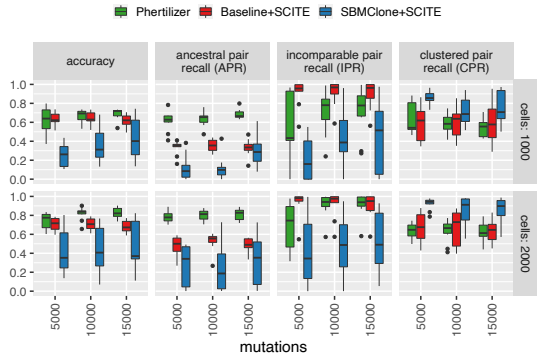
(a) cell placement metrics for $k = 5$ clones



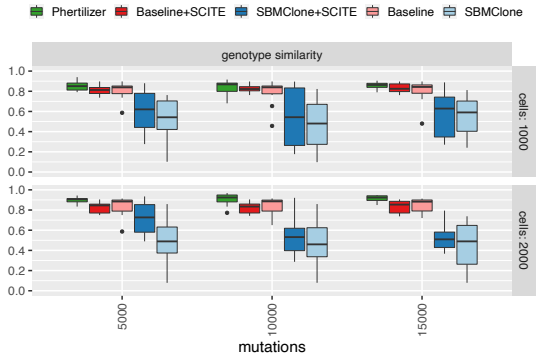
(b) cell placement metrics for $k = 9$ clones



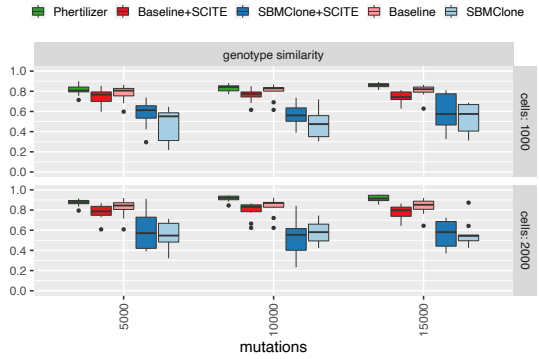
(c) SNV placement metrics for $k = 5$ clones



(d) SNV placement metrics for $k = 9$ clones

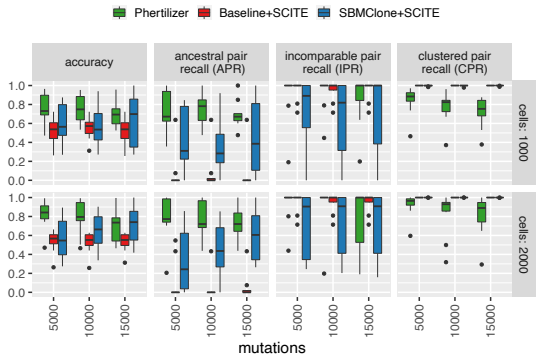


(e) genotype similarity for $k = 5$ clones

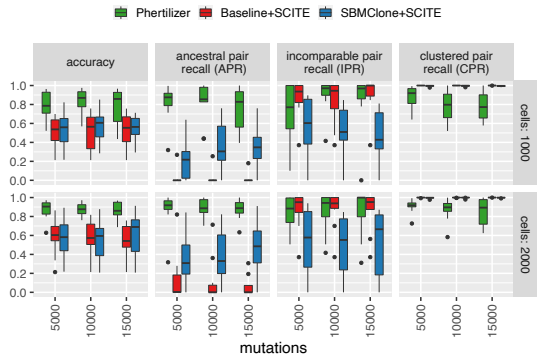


(f) genotype similarity for $k = 9$ clones

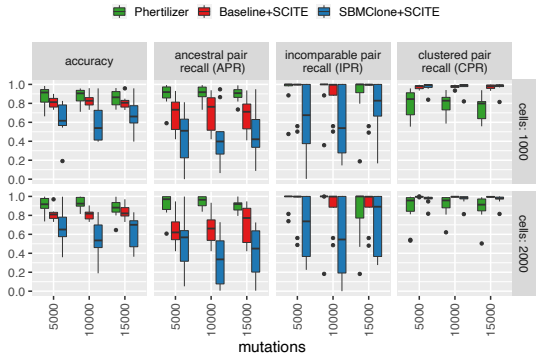
Figure S7: Simulation results for coverage $g = 0.01\times$



(a) cell placement metrics for $k = 5$ clones

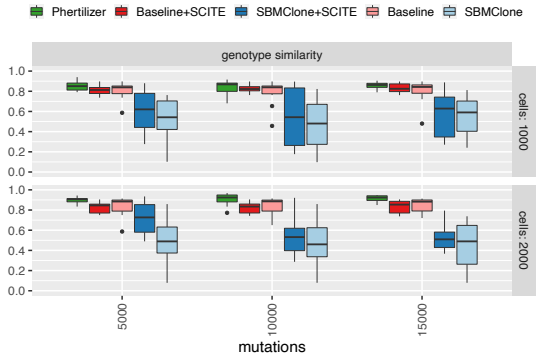


(b) cell placement metrics for $k = 9$ clones

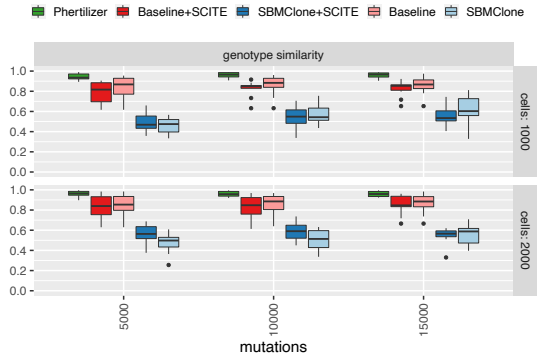


(c) SNV placement metrics for $k = 5$ clones

(d) SNV placement metrics for $k = 9$ clones

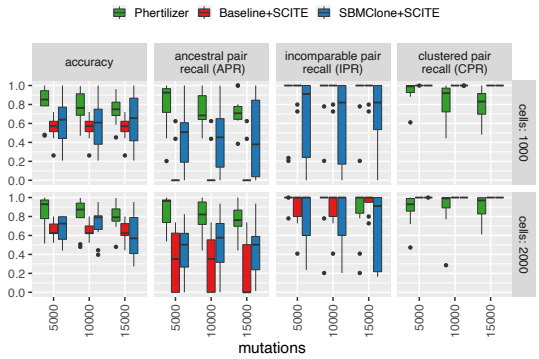


(e) genotype similarity for $k = 5$ clones

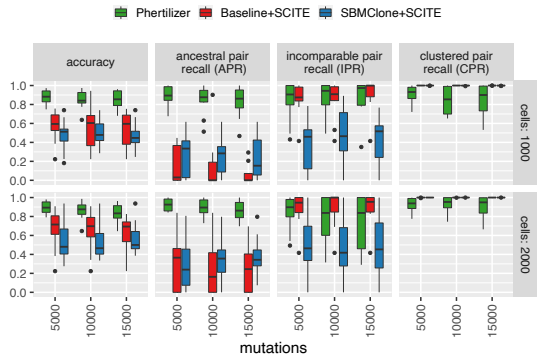


(f) genotype similarity for $k = 9$ clones

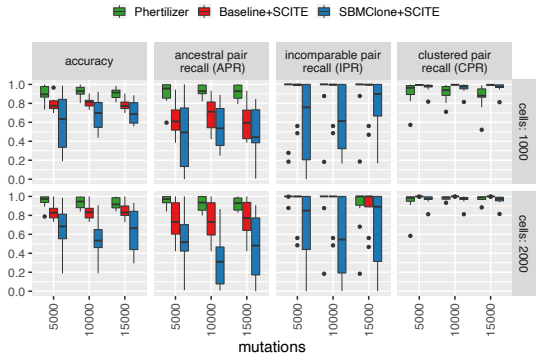
Figure S8: Simulation results for coverage $g = 0.05\times$.



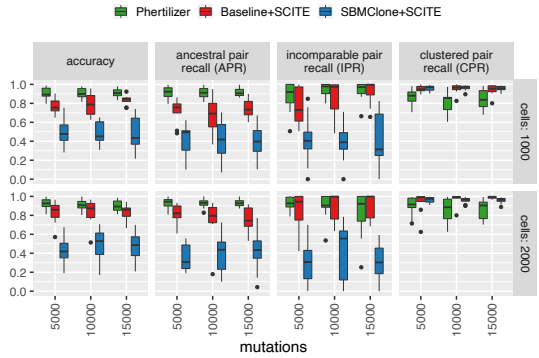
(a) cell placement metrics for $k = 5$ clones



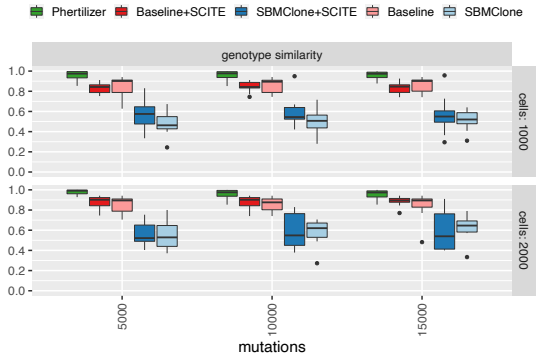
(b) cell placement metrics for $k = 9$ clones



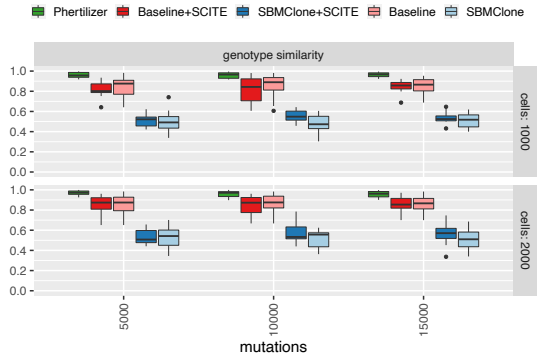
(c) SNV placement metrics for $k = 5$ clones



(d) SNV placement metrics for $k = 9$ clones

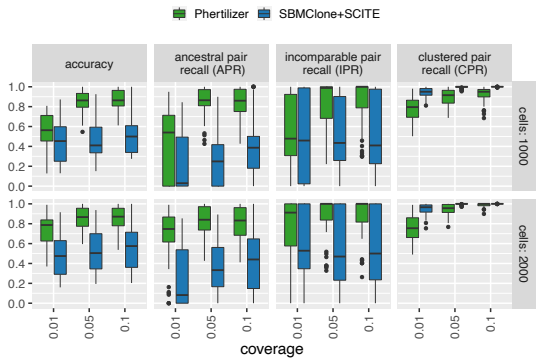


(e) genotype similarity for $k = 5$ clones

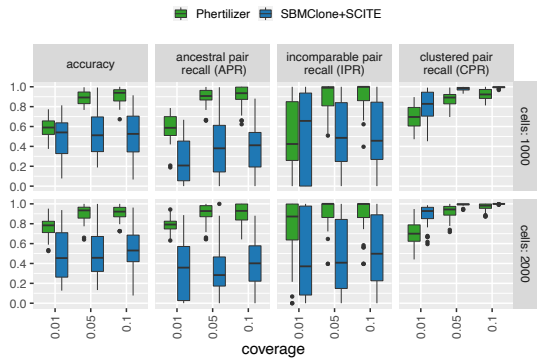


(f) genotype similarity for $k = 9$ clones

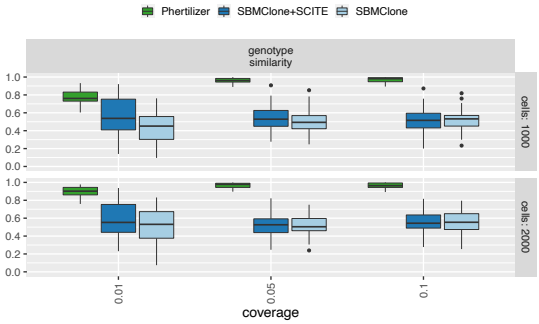
Figure S9: Simulation results for coverage $g = 0.1\times$.



(a) cell placement metrics

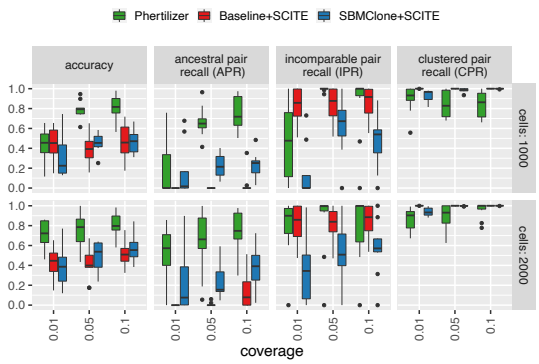


(b) SNV placement metrics

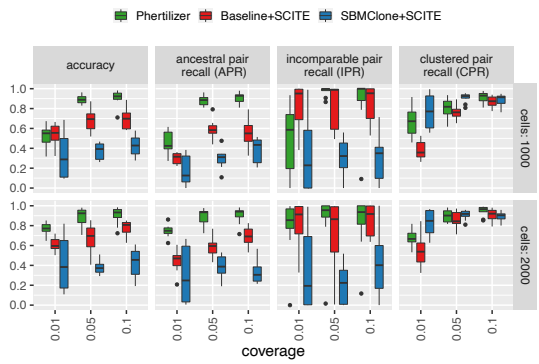


(c) genotype similarity

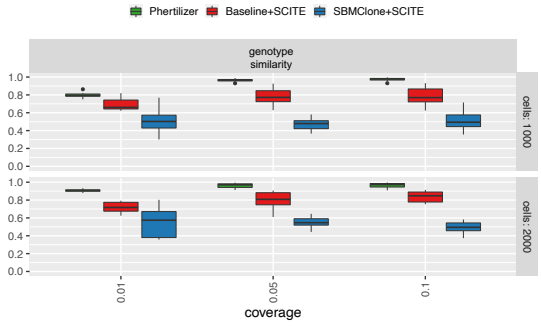
Figure S10: Heterozygous diploid simulation results aggregrated over $k \in \{5, 9\}$ clones and $m \in \{5000, 10000, 15000\}$ SNVs. BASELINE+SCITE was excluded from comparison because the lack of CNAs in the data resulted in only a single clone being identified in the read count embedding space for each instance.



(a) cell placement metrics



(b) SNV placement metrics



(c) genotype similarity

Figure S11: Dollo evolutionary model simulation results for $k = 9$ clones and $m = 15000$ SNVs.

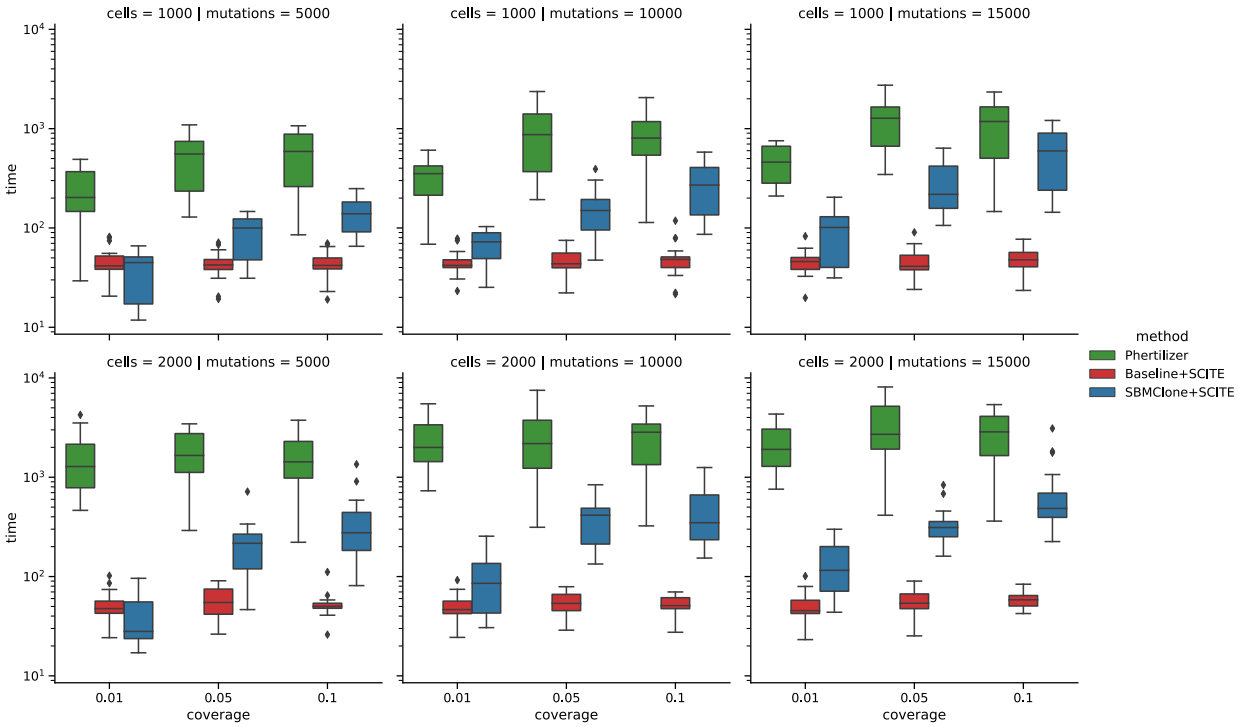


Figure S12: Running time (in seconds) on simulation data.

B.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.

- Appendix B.2.1 provides details of the PHERTILIZER runtime parameters used for experimental data.
- Appendix B.2.2 contains amplifying information on how cancer related genes were placed on inferred trees.
- Appendix refsupp: cell mutational burden provides additional description and analysis of the cell mutational burden metric
- Appendix B.2.4 contains supplementary results for the high-grade serous ovarian cancer cells sequenced with DLP+
- Appendix B.2.5 provides the data processing details for the triple negative breast cancer tumors sequenced via ACT
- Appendix B.2.6 contains supplemental results and figures for the triple negative breast cancer tumors sequenced via ACT

B.2.1 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 A.3) and the lower bound for the quality check qc $\in \{0.05, 0.075\}$ (Appendix A.4.3). 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.

B.2.2 Placement of driver genes on inferred trees

We annotated inferred clonal trees with cancer-related genes listed in the Cancer Gene Census (CGC) [37] from COSMIC v97 and cBioPortal [35,36]. 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 [51], and looked for missense variants that are predicted to be damaging or deleterious by both SIFT [52] and PolyPhen [53]. 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.

B.2.3 Cell mutational burden (CMB)

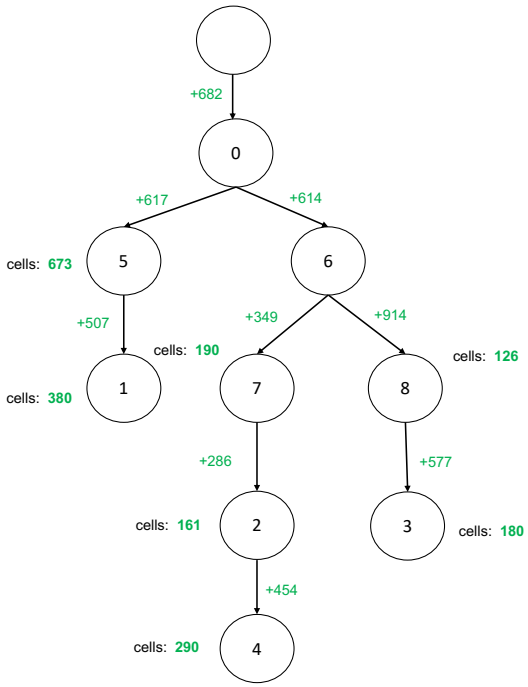
To assess the quality of each inferred clade for real data, we developed a performance metric called *cell mutational burden* (CMB) defined as

$$\mathrm{CMB}(i, M) = \sum_{q \in M} \mathbf{1}\{a_{iq} > 0\} / \sum_{q \in M} \mathbf{1}\{d_{iq} > 0\}. \tag{46}$$

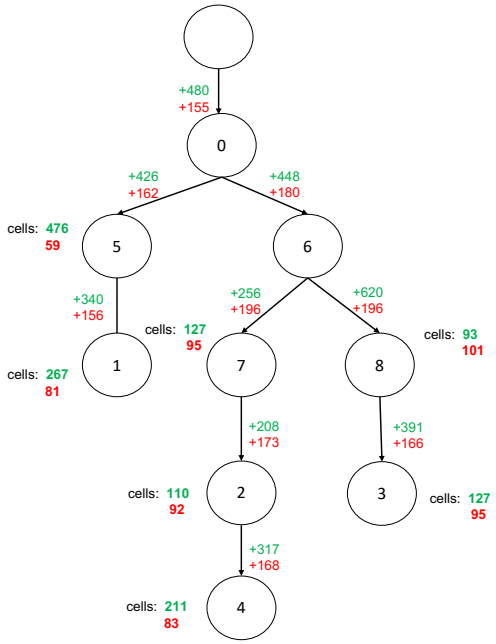
In words, $\mathrm{CMB}(i, M)$ is the fraction of mapped SNV loci M with mapped variant reads in cell i. For a specified *clade* j or subtree rooted at node v_j, SNVs M_j are the SNVs gained at node v_j. CMB is designed to assess the goodness of fit of a proposed clonal tree without a known ground truth 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 $\mathrm{CMB}(i, M_j)$ to be high, although the value will depend on copy number. By contrast, for cells placed outside of clade j, we expect $\mathrm{CMB}(i, M_j)$ to be low.

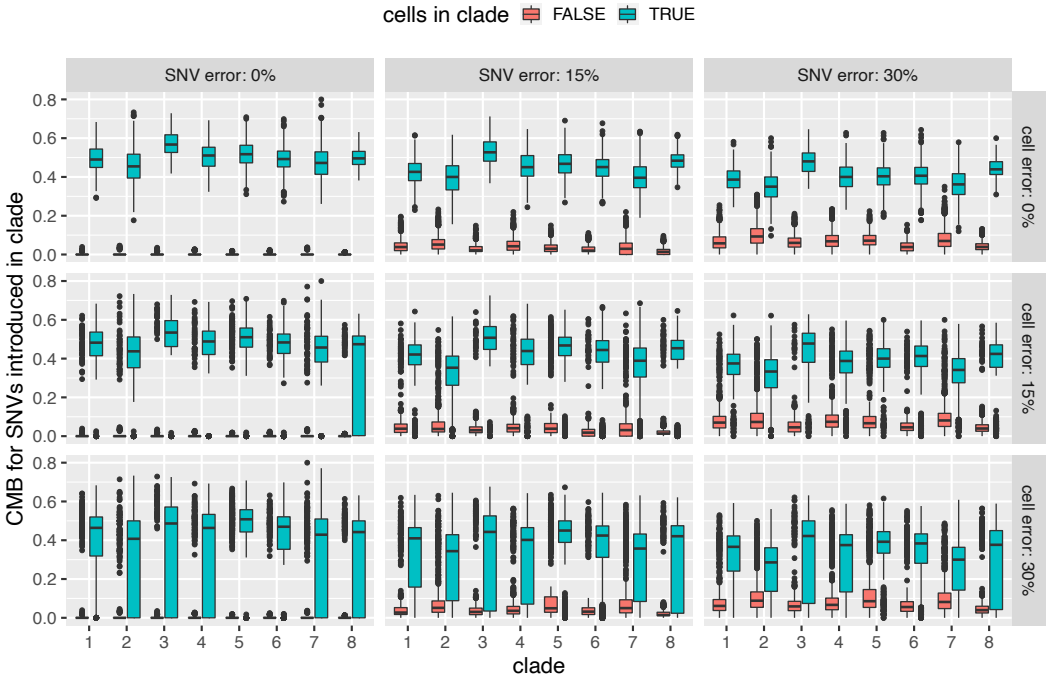
Because this is a newly proposed metric that has complex interactions between inference errors, we performed a sensitivity analysis on a simulated tree (Fig. S13a) with varying rates in $\{0\%, 15\%, 30\%\}$ of cell and SNV placement errors (Fig. S13). For reference, we also show cell and SNV placement on the simulated ground truth tree (Fig. S13b) 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 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.



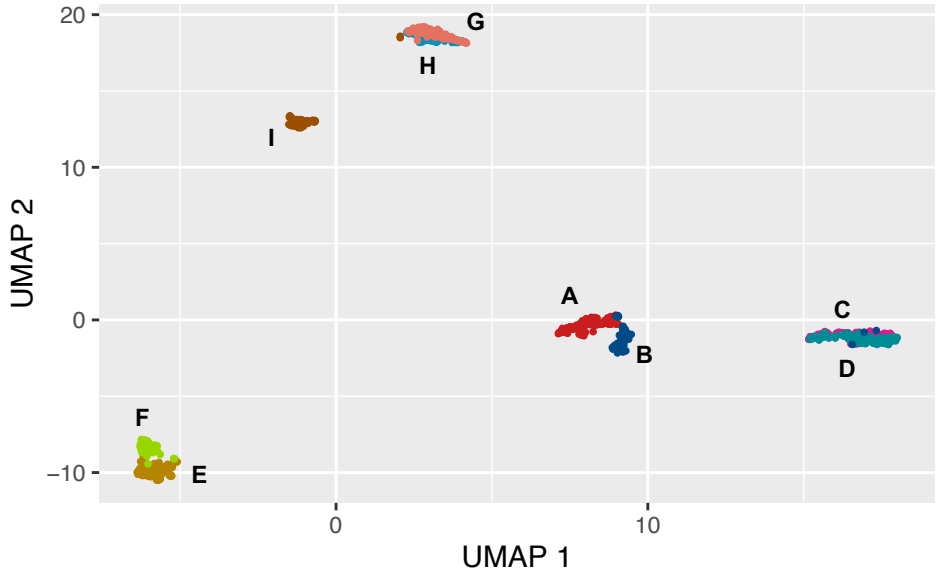
(b) Simulated tree with cell 30% and SNV 30% error rate.



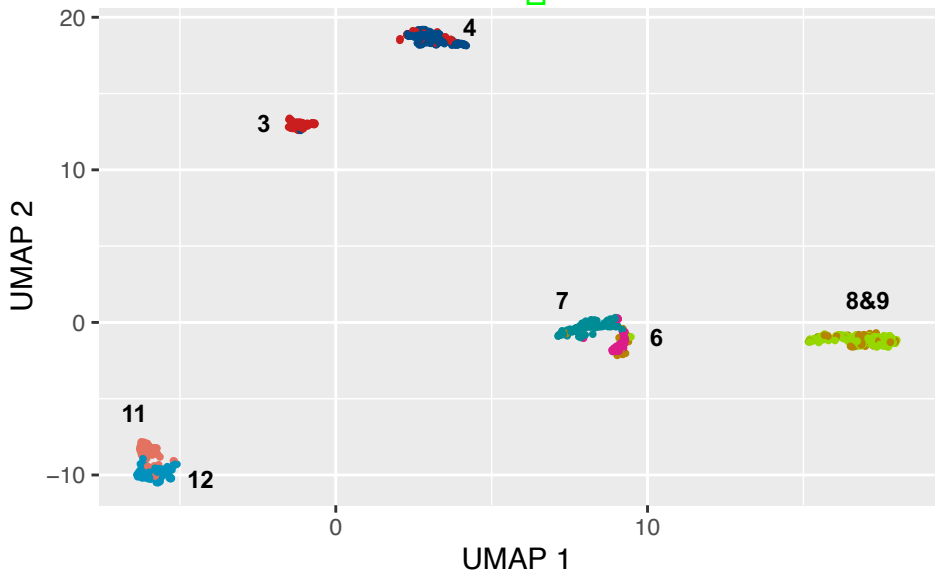
(c) Cell mutational burden (CMB) with varying cell and placement error rates.

Figure S13: 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.

B.2.4 Supplemental figures for high-grade serous ovarian cancer cells sequenced with DLP+



(a) Laks et al. [6] cell clusters



(b) Phertilizer cell clusters

Figure S14: UMAP for the high-grade serous ovarian cancer patient depicted with inferred cell clusterings.

B.2.5 Processing the triple negative breast cancer tumors sequencing data from ACT

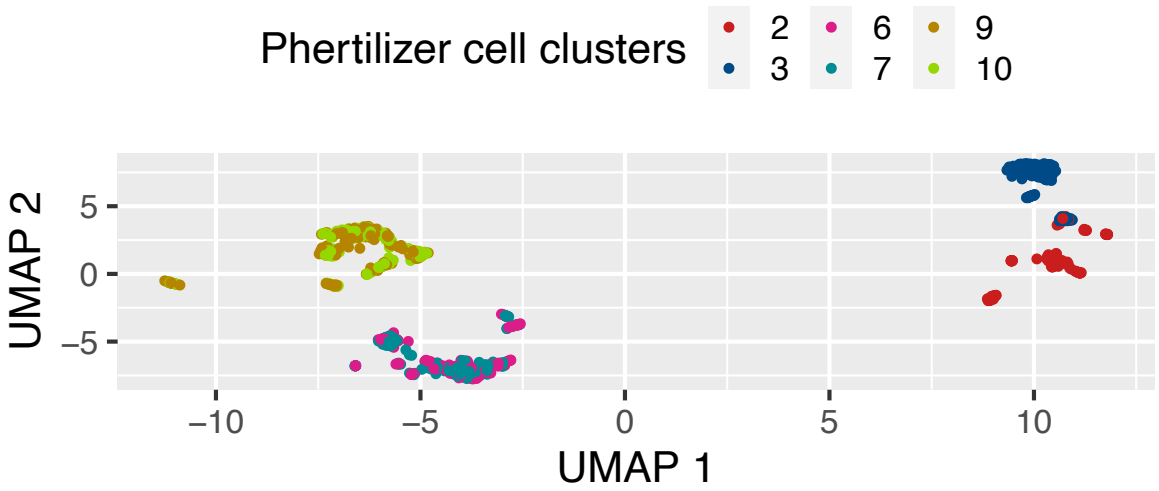
We merged reads from all cells into a single FASTQ file for each sample, aligned the merged reads to the Human reference genome hg19 (GRCh37) using bowtie2 (v.2.4.4), and sorted by samtools (v.1.15) forming pseudo-bulk samples. We then get a set of SNVs by running Mutect2 [38] in tumor-only mode on each 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 [43].

B.2.6 Supplemental tables and figures for triple negative breast cancer tumors sequenced with ACT

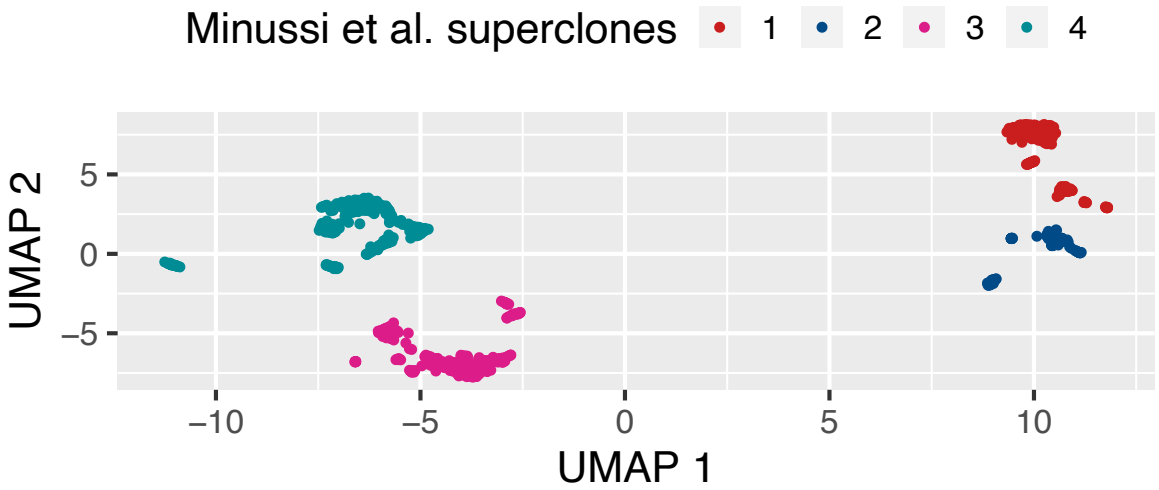
- Table S1 provides a summary of the ACT data of all eight triple negative breast cancer tumors
- Fig. S15 shows a comparison of PHERTILIZER and Minussi et. al. [7] inferred cell clustering in the embedding space
- Fig. S16 shows the cell mutational burden (CMB) results for BASELINE+SCITE inferred trees for breast tumors TN3 and TN5
- Fig. S17 shows PHERTILIZER supplemental results for breast tumor TN2
- Fig. S18 shows PHERTILIZER supplemental results for breast tumor TN4
- Fig. S19 shows PHERTILIZER supplemental results for breast tumor TN8

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 S1: 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.

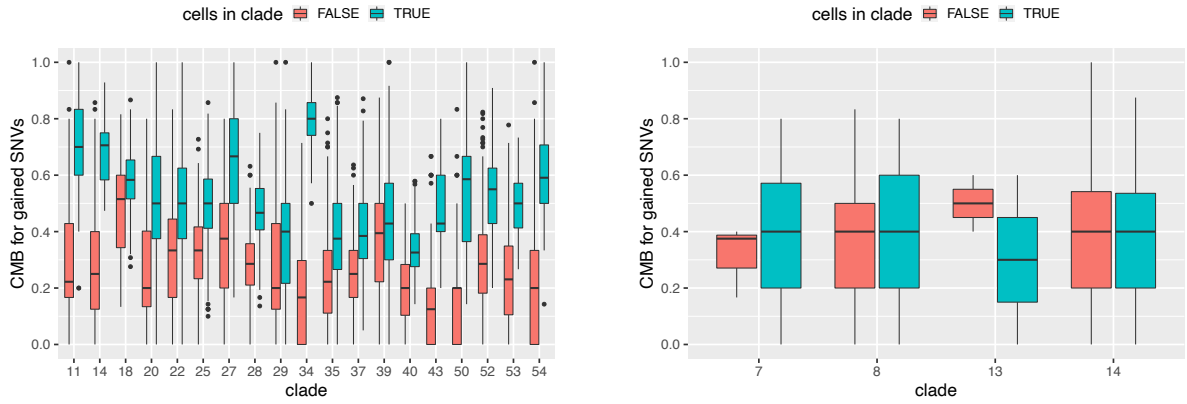


(a) PHERTILIZER UMAP for the triple negative breast tumor TN1.



(b) Minussi et al. [7] UMAP for the triple negative breast tumor TN1.

Figure S15: Cell clustering comparison for triple negative breast tumor TN1 in embedding space between PHERTILIZER and Minussi et al. [7].



(a) Triple negative breast tumor TN3

(b) Triple negative breast tumor TN5

Figure S16: Cell mutational burden per clade in trees inferred by BASELINE+SCITE for tumors TN3 and TN5.

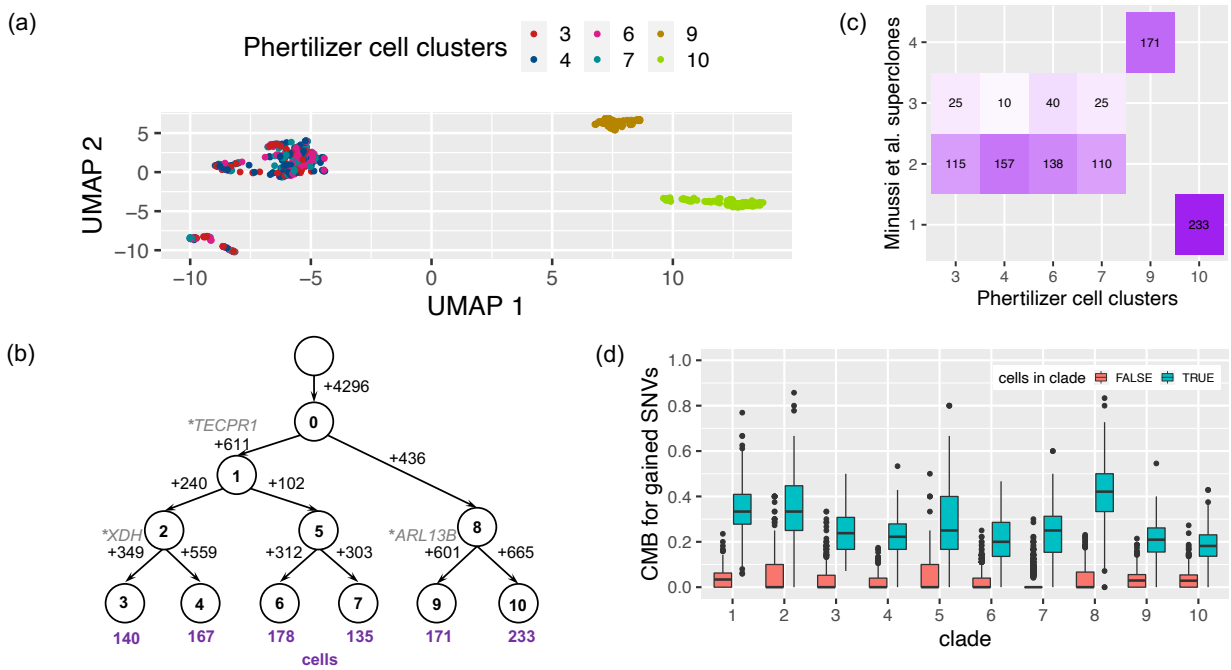


Figure S17: **PHERTILIZER infers clonal tree for breast cancer tumor TN2.** (a) UMAP for the triple negative breast tumor TN2. (b) The tree inferred by PHERTILIZER with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs ('*': stop-gain variant). (c) A mapping between PHERTILIZER's cell clusters and the Minussi et al. [7] superclones. (d) The CMB distribution for the inferred clades for TN2.

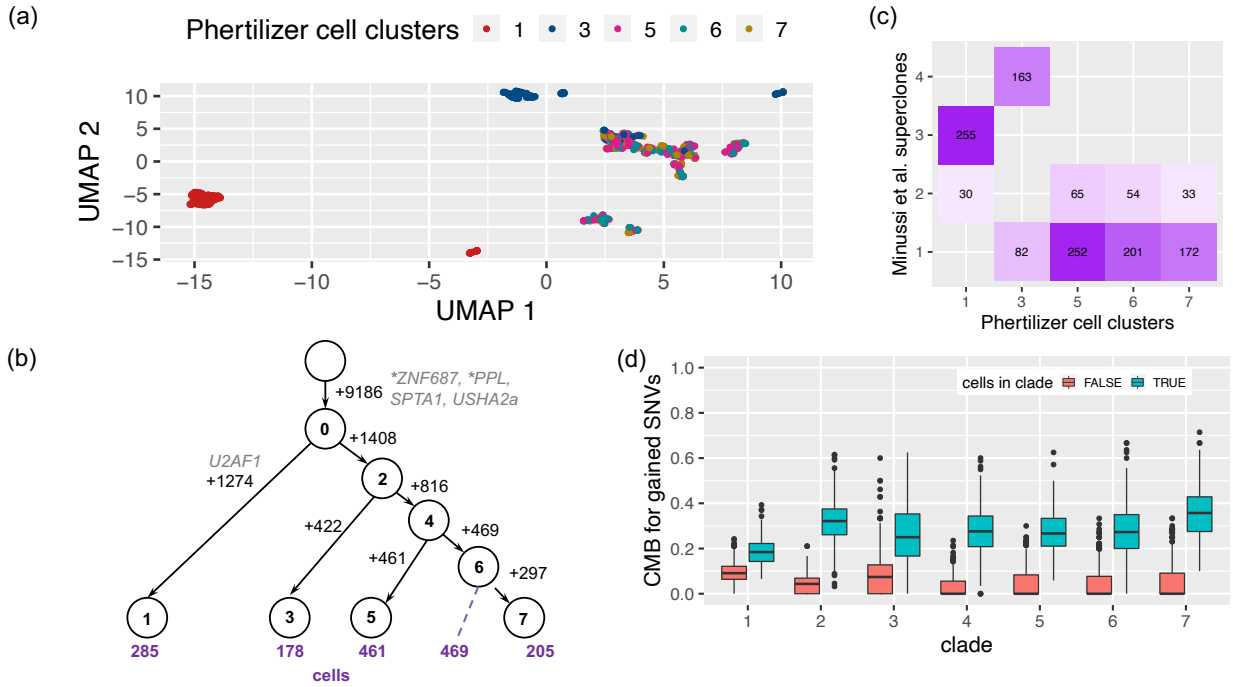


Figure S18: **PHERTILIZER infers clonal tree for breast cancer tumor TN4.** (a) UMAP for triple negative breast tumor TN4. (b) The tree inferred by PHERTILIZER with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs ('*': stop-gain variant). (c) A mapping between PHERTILIZER's cell clusters and the Minussi et al. [7] superclones. (d) The CMB distribution for the inferred clades for TN4.

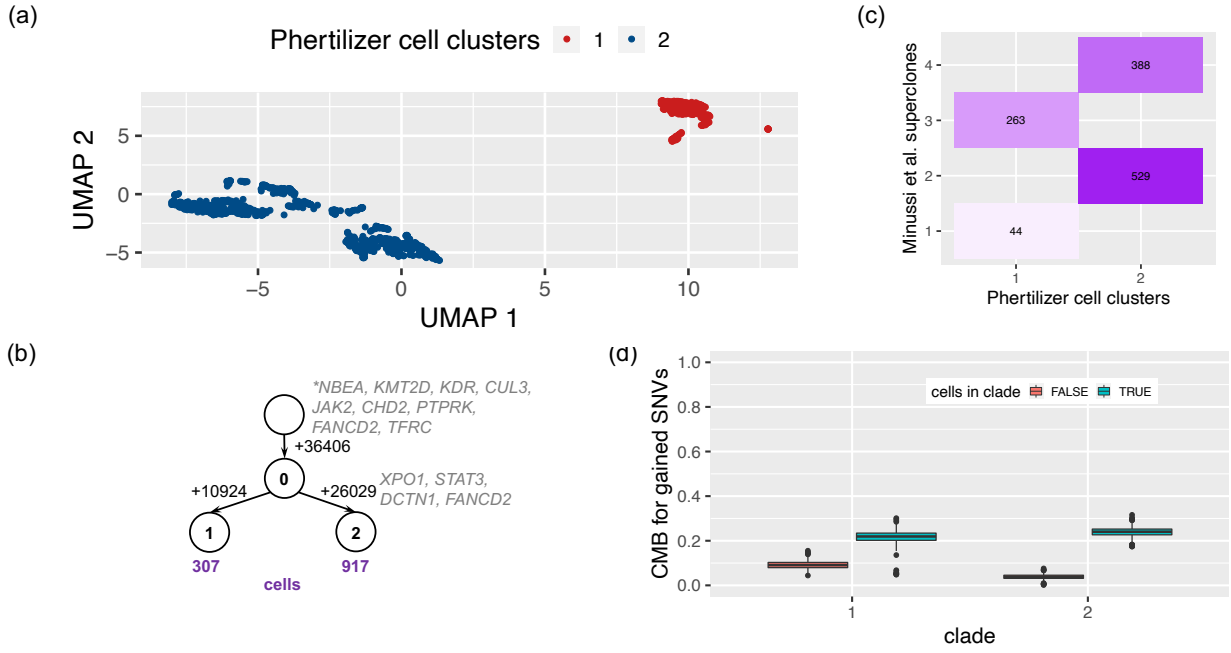


Figure S19: **PHERTILIZER infers clonal tree for breast cancer tumor TN8.** (a) UMAP for triple negative breast tumor TN8. (b) The tree inferred by PHERTILIZER with numbers of SNVs labeled beside the edges, and numbers of cells labeled beneath the leaves. Cancer-related genes are labeled next to the SNVs ('*': stop-gain variant). (c) A mapping between PHERTILIZER's cell clusters and the Minussi et al. [7] superclones. (d) The CMB distribution for the inferred clades for TN8.