Supplementary material for SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data

H. Zafar[1,2], N. Navin[3], K. Chen[2], and L. Nakhleh[1,*]

[1]Department of Computer Science, Rice University, Houston, Texas, USA

[2]Department of Bioinformatics and Computational Biology, the University of Texas M.D. Anderson Cancer Center, Houston, Texas 77030, USA

[3]Department of Genetics, the University of Texas M.D. Anderson Cancer Center, Houston, Texas 77030, USA

[*]Corresponding author: nakhleh@rice.edu

July 7, 2019

Contents

1 Supplemental Methods **2**

1.1 Singlet Model of SiCloneFit 3

1.1.1 Model Overview 3

1.1.2 Model Description 3

1.1.3 Model of Evolution 5

1.1.4 Single-cell Error Model 6

1.1.5 Posterior Distribution . . . 7
1.1.6 Likelihood Function . . . 8
1.1.7 Prior Distributions . . . 8
1.1.7.1 Prior on Clonal Genotypes . . . 8
1.1.7.2 Prior on Partition of Cells into Clonal Clusters . . . 9
1.1.7.3 Prior on Phylogeny . . . 9
1.1.7.4 Prior on Other Parameters . . . 9
1.1.8 Inference . . . 9
1.1.9 Partial Reversible-jump MCMC Partial Gibbs Sampling Algorithm . 10
1.1.9.1 Algorithm For Sampling Cluster Indicators . . . 13
1.1.9.1.1 Likelihood Ratio . . . 13
1.1.9.1.2 Prior Ratio . . . 14
1.1.9.1.3 Proposal Ratio and Jacobian . . . 15
1.1.9.2 Algorithm For Sampling Clonal Phylogeny and Evolution Model Parameters . . . 17
1.1.9.3 Algorithm For Sampling Clonal Genotypes . . . 18
1.1.9.4 Algorithm For Sampling Error Rates . . . 19
1.1.9.4.1 False Positive Rate . . . 20
1.1.9.4.2 False Negative Rate . . . 20
1.2 Doublet Model of SiCloneFit . . . 21
1.2.1 Model Overview . . . 21
1.2.2 Model Description . . . 21
1.2.3 Posterior Distribution . . . 22
1.2.4 Likelihood Function . . . 23
1.2.5 Prior Distributions . . . 24
1.2.6 Inference . . . 25
1.2.7 Partial Reversible-jump MCMC Partial Gibbs Sampling Algorithm . 25

2 Supplemental Results **29**
2.1 Benchmarking on Simulated Datasets . . . 29

2.1.1 Simulation of Synthetic Datasets 29
2.1.1.1 Simulation of Clonal Clusters 29
2.1.1.2 Simulation of Clonal Phylogeny 30
2.1.1.3 Simulation of Clonal Genotypes 31
2.1.1.4 Simulation of Noisy Single-cell Genotypes 32
2.1.1.4.1 Simulating Doublets 32
2.1.1.4.2 Simulating False Negative and False Positive Errors 32
2.1.1.4.3 Simulating Missing Data 33
2.1.2 Summarizing Posterior Samples from SiCloneFit 33
2.1.3 Competitor Methods 34
2.1.3.1 SCG 35
2.1.3.2 OncoNEM 35
2.1.3.3 SCITE 35
2.1.3.4 SiFit 36
2.1.4 Performance Metrics 37
2.1.4.1 Accuracy of Clustering 37
2.1.4.1.1 Adjusted Rand Index 37
2.1.4.1.2 B-Cubed F-score 38
2.1.4.2 Accuracy in Inferring Clonal Genotypes 38
2.1.4.3 Accuracy in Inferring Clonal Phylogeny 39
2.1.5 Results and Discussion 39
2.1.5.1 Testing the Finite-site Model 39
2.1.5.1.1 Performance on Datasets with Varying Deletion Probability 40
2.1.5.1.2 Performance on Datasets with Varying Probability of LOH 41
2.1.5.1.3 Performance on Datasets with Varying Probability of Recurrent Mutation 42

2.1.5.2 Performance on Datasets with Varying Number of Cells Without Doublets 43
2.1.5.2.1 Clustering Accuracy 44
2.1.5.2.2 Genotyping Accuracy 44
2.1.5.2.3 Clonal Phylogeny Inference Accuracy 44
2.1.5.3 Performance on Datasets With Varying Number of Clonal Populations 45
2.1.5.4 Performance on Datasets with Increasing Error Rates . . . 46
2.1.5.4.1 Robustness to Increasing False Negative Rate . . 46
2.1.5.4.2 Robustness to Increasing False Positive Rate . . 47
2.1.5.5 Performance on Datasets with Missing Data 48
2.1.5.5.1 Clustering Accuracy 48
2.1.5.5.2 Genotyping Accuracy 49
2.1.5.5.3 Clonal Phylogeny Inference Accuracy 49
2.1.5.6 Performance on Datasets Generated Under Neutral Evolution 49
2.1.5.7 Estimation of Error Rates by SiCloneFit 50
2.1.5.8 Estimation of Number of Clusters by SiCloneFit 51
2.1.5.9 Scalability of SiCloneFit for Large Datasets 52
2.1.5.10 Performance on Datasets with Varying Number of Cells with Doublets 53
2.1.5.10.1 Clustering Accuracy 53
2.1.5.10.2 Genotyping Accuracy 54
2.1.5.10.3 Clonal Phylogeny Inference Accuracy 54
2.1.5.11 Performance on Datasets Containing Doublets with Varying Number of Clonal Populations 54
2.1.5.12 Performance on Datasets Containing Doublets and Missing Data 55
2.1.5.12.1 Clustering Accuracy 55
2.1.5.12.2 Genotyping Accuracy 55

2.1.5.12.3 Clonal Phylogeny Inference Accuracy 56
2.2 Inference of Clonal Clusters, Genotypes and Phylogeny from Experimental SCS Data 56
2.2.1 Analysis of Patient CRC1 56
2.2.2 Analysis of Patient CRC2 58
2.3 Identification of Doublets from Experimental SCS Data 60

3 Supplemental Figures **61**

4 Supplemental Tables **96**

1 Supplemental Methods

Here, we describe the model and inference algorithm of SiCloneFit, a Bayesian nonparametric framework for simultaneous reconstruction of clonal populations of cells, clonal genotypes and clonal phylogeny from noisy somatic single nucleotide variant (SNV) profiles of single cells. This probabilistic framework jointly solves different aspects of intra-tumor phylogeny problem and automatically

1. estimates the number of clonal populations,
2. infers the clonal population of origin for each single cell,
3. estimates the clonal genotypes, and
4. places the clonal clusters at the leaves of a clonal phylogeny, a phylogenetic tree that reflects the evolutionary relationships between different clonal populations.

For an ease of exposition, we first describe the basic singlet (all cells are assumed to be singlets) model and later on extend that model to account for doublets.

1.1 Singlet Model of SiCloneFit

1.1.1 Model Overview

We derive the SiCloneFit model in the following section. The probabilistic graphical model is presented in Supplemental Fig. S1. A list of model variables is provided in Supplemental Table S2, hyper-parameters are described in Supplemental Table S3 and associated indices have been described in Supplemental Table S1.

1.1.2 Model Description

We assume that we have measurements from m single cells. For each cell, n somatic single nucleotide variant (SNV) sites have been measured. The data can be represented by a matrix $D_{n\times m} = (D_{ij})$ of observed genotypes, where D_{ij} is the observed genotype at the i^{th} site of cell j. Let g_t be the set of possible true genotype values for the SNVs, and g_o be the set of observable values for the SNVs. For binary measurements for SNVs, $g_t = \{0, 1\}$, whereas $g_o = \{0, 1, X\}$, where 0, 1 and X denote the absence of mutation, presence of mutation, and missing value respectively. If ternary measurements are available for SNVs, $g_t = \{0, 1, 2\}$ and $g_o = \{0, 1, 2, X\}$, where 0 denotes homozygous reference genotype, 1 and 2 denote heterozygous, and homozygous non-reference genotypes, respectively, and X denotes missing data.

We assume that there is a set of K clonal populations from which m single cells are sampled and the clonal populations can be placed at the leaves of a clonal phylogeny, $\mathcal{T}$. Each clonal population consists of a set of cells that have identical genotype (with respect to the set of mutations in consideration) and a common ancestor. The genotype vector associated with a clone c is called clonal genotype (denoted by G_c) and it records the genotype values for all n sites for the corresponding clone. The true genotype vector of each cell is identical to the clonal genotype of the clonal population where it belongs to. The clonal genotype matrix, $G_{K\times n}$, represents the clonal genotypes of K clones. It is important to note that, K, the number of clones is unknown. To automatically infer the number of clones and assign the cells to clones, we introduce a tree-structured infinite mixture model. [18] describes a nonparametric Bayesian prior over trees similar to mixture models using

a Chinese restaurant process (CRP) [21] prior. For this tree-structured CRP, each node of the tree represents a cluster. In our model, we extend this idea to define a nonparametric Bayesian prior over binary trees, leaves of which represent the mixture components (clonal clusters). A Chinese restaurant process defines a distribution for partitioning customers into different tables. In our problem, single cells are analogous to customers and clonal clusters are analogous to tables. Let c_j denote the cluster assignment for cell j and assume that cells $1:j-1$ have already been assigned to clonal clusters $\{1,\ldots,|\boldsymbol{c}_{1:j-1}|\}$, where $|\boldsymbol{c}_{1:j-1}|$ denotes the number of clusters induced by the cluster indicators of $j-1$ cells. The cluster assignment of cell j, c_j is based on the distribution defined by a Chinese restaurant process is given by

$$\begin{aligned} p(c_j = c|c_{1:(j-1)},\alpha_0) &= \frac{n_c}{j-1+\alpha_0} \\ p(c_j \neq c_k \forall k < j|c_{1:(j-1)},\alpha_0) &= \frac{\alpha_0}{j-1+\alpha_0} \end{aligned} \tag{1}$$

where n_c denotes the number of cells already assigned (excluding cell j) to cluster c. α_0 is the concentration parameter for the CRP model.

The clonal phylogeny, $\mathcal{T}$, is a rooted directed binary tree whose number of leaves is equal to the number of clonal clusters, $K = |\boldsymbol{c}|$ defined by the assignment of m cells to different clusters by the CRP. The root of $\mathcal{T}$ represents normal (unmutated) genotype and somatic mutations are accumulated along the branches of the phylogeny. Each leaf in the clonal phylogeny corresponds to a clonal cluster, $c \in \{1,\ldots,K\}$ and is associated with a clonal genotype G_c that records the set of mutations accumulated along the branches from the root. To model the evolution of the clonal genotypes, we employ a finite-site model of evolution, $\mathcal{M}_\lambda$, that accounts for the effects of point mutations, deletion and loss of heterozygosity on the clonal genotypes. The model of evolution assigns transition probabilities to different genotype transitions along the branches of the clonal phylogeny. The true genotype of each cell is identical to the clonal genotype of the clonal cluster where it is assigned. However, observed genotypes of single cells differ from their true genotype due to amplification errors introduced during the single-cell sequencing work flow. The effect of amplification errors is modeled using an error model distribution parameterized

by FP error rate, α and FN error rate, β. The generative process can be described as follows:

1. draw $\alpha_0 \sim Gamma(a, b)$, $\alpha \sim Beta(a_\alpha, b_\alpha)$, $\beta \sim Beta(a_\beta, b_\beta)$
2. For $j \in \{1, 2, \ldots, m\}$, draw $c_j \sim CRP(\alpha_0)$.
   From this, derive $K = |\boldsymbol{c}|$, the total number of clusters (or clones) implicitly defined by $\boldsymbol{c}$.
3. draw $\mathcal{T} \sim T_{prior}(K)$.
4. For $\lambda \in \mathcal{M}_\lambda$, draw $\lambda \sim Beta(a_{M_\lambda}, b_{M_\lambda})$
5. For $k \in \{1, 2, \ldots, K\}$, draw $G_k \sim F(G_k|\mathcal{T}, \mathcal{M}_\lambda)$.
6. For $j \in \{1, 2, \ldots, m\}$ and $i \in \{1, 2, \ldots, n\}$, draw $D_{ij} \sim E(D_{ij}|G_{c_j i}, \alpha, \beta)$.

$\boldsymbol{c}$ denotes the clonal assignments of all cells. T_{prior} is the prior distribution on phylogenetic trees for a fixed number of leaves. $\mathcal{M}_\lambda$ denotes the set of parameters in the finite-sites model of evolution. F denotes a distribution on the genotypes at the leaves of a phylogenetic tree and can be computed using Felsenstein's pruning algorithm [6] given the phylogeny and a finite-site model of evolution. E is the error model distribution that relates the observed genotype at locus i for cell j, D_{ij} to clonal genotype $G_{c_j i}$. $a, b, a_\alpha, b_\alpha, a_\beta, b_\beta, a_M, b_M$ denote different hyperparameters used in this model.

1.1.3 Model of Evolution

To capture the effect of point mutations, LOH and deletion on the clonal genotypes along the branches of clonal phylogeny, we employ a finite-site model of evolution similar to the one introduced in SiFit [29]. The finite-site model of evolution, $\mathcal{M}_\lambda$, is modeled using a continuous-time Markov chain that assigns a probability with each possible transition of genotypes. The branches of clonal phylogeny $\mathcal{T}$, have associated branch lengths that represent expected number of mutations per locus. We assume that the genomic loci evolve identically and independently. For ternary genotype, $g_t = \{0, 1, 2\}$, a 3×3 transition probability matrix describes the model of evolution. The transition probability matrix, P_t, along a branch of length t is given by $P_t = exp(Qt)$, where, Q denotes the transition

rate matrix of the Markov chain. The transition rate matrix consists of the infinitesimal rates (during infinitesimally small time, Δt) for switching between genotype states for the continuous-time Markov chain. As in SiFit, we assume that only one event can occur at a site during Δt, the smallest unit of time. The parameter λ_r accounts for the effect of recurrent mutation and the parameter λ_l captures mutation loss due to deletion and LOH. The product of the transition rate matrix and the branch length (t) is given by:

$$Qt = \begin{bmatrix} -t & t & 0 \\ \frac{(\lambda_r+\lambda_l)\times t}{2} & -(\lambda_r+\lambda_l)\times t & \frac{(\lambda_r+\lambda_l)\times t}{2} \\ 0 & \lambda_r \times t & -\lambda_r \times t \end{bmatrix} \tag{2}$$

In Eq. (2), $Qt(i,j)$ denotes the rate of genotype i changing to genotype j along a branch of length t, $i,j \in \{0,1,2\}$. We assume that the parameters λ_r and λ_l are Beta distributed as they represent relative rates with value between 0 and 1. $P_t(i,j)$ denotes the probabality of transition of genotype i to genotype j along a branch of length t. Each entry of P_t is a function of t, λ_r and λ_l.

For binary genotype states, the product of transition rate matrix and branch length is given by:

$$Qt = \begin{bmatrix} -t & t \\ \frac{(\lambda_r+\lambda_l)\times t}{2} & -\frac{(\lambda_r+\lambda_l)\times t}{2} \end{bmatrix} \tag{3}$$

1.1.4 Single-cell Error Model

The FP and FN errors in single-cell SNV profiles have been modeled using two parameters α and β respectively as in SiFit [29]. The error model distribution, $E(D_{ij}|G_{c_j i}, \alpha, \beta)$, gives the probability of observing genotype D_{ij} for locus i in cell j, given the true clonal genotype $G_{c_j i}$ and Supplemental Table S4 shows it for ternary genotype. Supplemental Table S5 shows the error model distribution for binary genotype. α and β are assumed to be Beta distributed variables as they represent probability of FP and FN errors respectively.

1.1.5 Posterior Distribution

The SiCloneFit model has several hidden variables as well as some observed variables. The posterior distribution, $\mathcal{P}$ over the latent variables is given by

$$\begin{aligned}
\mathcal{P}(\boldsymbol{c}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \alpha_0 | \boldsymbol{D}, a_\alpha, b_\alpha, a_\beta, b_\beta, a_M, b_M, a, b) \propto \\
P(\boldsymbol{D}|\boldsymbol{c}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \alpha_0, a_\alpha, b_\alpha, a_\beta, b_\beta, a_M, b_M, a, b) \times \\
P(\boldsymbol{c}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \alpha_0 | a_\alpha, b_\alpha, a_\beta, b_\beta, a_M, b_M, a, b) \\
= E(\boldsymbol{D}|\boldsymbol{c}, \boldsymbol{G}, \alpha, \beta) F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda) P(\boldsymbol{c}|\alpha_0) P(\mathcal{T}) \\
P(\alpha|a_\alpha, b_\alpha) P(\beta|a_\beta, b_\beta) P(\mathcal{M}_\lambda|a_M, b_M) P(\alpha_0|a, b) \quad (4)
\end{aligned}$$

The hidden variables that we want to estimate from this model are

1. $\boldsymbol{c}$, a vector containing the cluster assignment for all cells,
2. $\boldsymbol{G}$, a $K \times n$ clonal genotype matrix, where G_k denotes the genotype of clone k, $K = |\boldsymbol{c}|$, the number of clusters defined by $\boldsymbol{c}$,
3. $\mathcal{T}$, the clonal phylogeny, representing the genealogical relationships between the clones,
4. $\mathcal{M}_\lambda$, parameters of the model of evolution,
5. α, false positive rate, and
6. β, false negative rate.

The number of clones is implicitly defined by the vector $\boldsymbol{c}$. The posterior probability is a product of likelihood function and prior. These are described in the following.

1.1.6 Likelihood Function

The likelihood function employed by SiCloneFit is given by

$$P(\boldsymbol{D}|\boldsymbol{c},\boldsymbol{G},\mathcal{T},\mathcal{M}_\lambda,\alpha,\beta,\alpha_0,a_\alpha,b_\alpha,a_\beta,b_\beta,a_M,b_M,a,b) = E(\boldsymbol{D}|\boldsymbol{c},\boldsymbol{G},\alpha,\beta) = \prod_{i=1}^{n}\prod_{j=1}^{m} E(D_{ij}|G_{c_j i},\alpha,\beta) \quad (5)$$

In Eq. (5), $E(D_{ij}|G_{c_j i},\alpha,\beta)$ is obtained from the error model distribution for binary and ternary genotype as defined in Supplemental Table S5 and Supplemental Table S4 respectively.

1.1.7 Prior Distributions

The SiCloneFit model incorporates a compound prior given by

$$P(\boldsymbol{c},\boldsymbol{G},\mathcal{T},\mathcal{M}_\lambda,\alpha,\beta,\alpha_0|a_\alpha,b_\alpha,a_\beta,b_\beta,a_M,b_M,a,b) = F(\boldsymbol{G}|\mathcal{T},\mathcal{M}_\lambda)P(\boldsymbol{c}|\alpha_0)P(\mathcal{T})P(\alpha|a_\alpha,b_\alpha)P(\beta|a_\beta,b_\beta)P(\mathcal{M}_\lambda|a_M,b_M)P(\alpha_0|a,b) \quad (6)$$

Below we describe each prior distribution.

1.1.7.1 Prior on Clonal Genotypes

$F(\boldsymbol{G}|\mathcal{T},\mathcal{M}_\lambda)$ denotes the prior distribution on the clonal genotype matrix keeping the clonal phylogeny and parameters of model of evolution fixed. $F(\boldsymbol{G}|\mathcal{T},\mathcal{M}_\lambda)$ can be efficiently calculated using Felsenstein's pruning algorithm [6] as

$$F(\boldsymbol{G}|\mathcal{T},\mathcal{M}_\lambda) = \prod_{i=1}^{n} F(\boldsymbol{G}_{*i}|\mathcal{T},\mathcal{M}_\lambda) \quad (7)$$

Here, $\boldsymbol{G}_{*i}$ denotes the genotype of all clones at i^{th} site. The prior probability for site i, $F(\boldsymbol{G}_{*i}|\mathcal{T},\mathcal{M}_\lambda)$ is given by the partial likelihood of the root r of clonal phylogeny $\mathcal{T}$ for genotype 0 and is computed using Felsenstein's pruning algorithm, a dynamic programming on clonal phylogeny that marginalizes over all possible mutational histories along the branches of the phylogeny.

1.1.7.2 Prior on Partition of Cells into Clonal Clusters

$P(\boldsymbol{c}|\alpha_0)$ denotes the prior probability of partitioning m single cells into $|\boldsymbol{c}|$ clusters under a CRP with concentration parameter α_0 and is given by

$$P(\boldsymbol{c}|\alpha_0) = \frac{\Gamma(\alpha_0)\alpha_0^{|\boldsymbol{c}|}}{\Gamma(\alpha_0 + m)} \prod_{k \in \boldsymbol{c}} \Gamma(n_k) \tag{8}$$

In Eq. (8), Γ denotes Gamma function, which is defined as $\Gamma(N) = (N-1)!$ for a positive integer N. n_k denotes the number of cells assigned to a clonal cluster k in the current cluster assignment $\boldsymbol{c}$.

1.1.7.3 Prior on Phylogeny

$P(\mathcal{T})$ denotes the prior probability on the clonal phylogeny. This is a product of prior on topology and prior on branch length. We consider uniform distribution for the prior on topology and exponential distribution for the prior on branch lengths. The overall prior probability for the branches is given by a product over the branches in the phylogeny.

1.1.7.4 Prior on Other Parameters

The values of the parameters α, β, $\mathcal{M}_\lambda = \{\lambda_r, \lambda_l\}$ lie between 0 and 1. So, we use Beta prior for these parameters. The hyperparameters for α and β are computed from the mean and standard deviation of these prior distribution and are kept fixed. The mean is computed from a simple estimation of α and β from the observed genotype matrix assuming usual rate for these parameters and wide standard deviation is used to cover a wide range of values.

For the concentration parameter α_0, we assume a Gamma prior as suggested in [5]. We set the value of hyperparameters for the Gamma distribution to $a = 1, b = 1$ for all the analyses performed, but this is also a configurable parameter in the software.

1.1.8 Inference

As analytically computing the posterior distribution given by Eq. (4) is computationally intractable, we implemented a Markov chain Monte Carlo (MCMC) sampling procedure

based on the Gibbs sampling algorithm. Different classes of Gibbs sampling algorithm have been designed to infer from infinite mixture models based on conjugate as well as non-conjugate prior distributions [16, 20]. Our algorithm is inspired by a partial Metropolis-Hastings partial Gibbs Sampling algorithm described in [20]. In our case, while performing the partial Metropolis-Hastings steps, the dimensionality of the sample may change due to addition of a new cluster (resulting in addition of new edge in the clonal phylogeny) or removal of an existing singleton cluster (resulting in removal of edges from the clonal phylogeny). In case the dimensionality changes, the absolute value of the determinant of the Jacobian matrix is also taken into account, which results in partial reversible-jump MCMC [9] updates. The resulting algorithm is a partial reversible-jump MCMC partial Gibbs sampling algorithm.

Our sampling algorithm samples the hidden variables from their corresponding conditional posterior distributions. In each iteration, it first samples the cluster indices for each cell, then the parameters of the model of evolution and the clonal phylogeny (on a number of leaves equal to the number of clones defined by cluster indices vector) is sampled. After that the clonal genotypes are sampled followed by sampling of α and β. Finally, the concentration parameter α_0 is sampled. The sampling algorithm is outlined below.

1.1.9 Partial Reversible-jump MCMC Partial Gibbs Sampling Algorithm

Given $\alpha_0^{(t-1)}$, $\{c_j^{(t-1)}\}_{j=1}^{m}$, $\{G_k^{(t-1)}\}_{k=1}^{|\boldsymbol{c}|}$, $\mathcal{T}^{(t-1)}$, $\mathcal{M}_\lambda^{(t-1)}$, $\alpha^{(t-1)}$, and $\beta^{(t-1)}$ from the previous iteration, we need to sample a new set of these parameters. $t-1$ denotes the previous iteration.

Set

- $\boldsymbol{c} = \boldsymbol{c}^{(t-1)}$, $\alpha_0 = \alpha_0^{(t-1)}$
- $\boldsymbol{G} = \{G_k^{(t-1)}\}_{k=1}^{|\boldsymbol{c}|}$
- $\mathcal{T} = \mathcal{T}^{(t-1)}$, $\mathcal{M}_\lambda = \mathcal{M}_\lambda^{(t-1)}$
- $\alpha = \alpha^{(t-1)}$, $\beta = \beta^{(t-1)}$

Sample cluster indicators:

1. For $j = 1, \ldots, m$, update c_j as follows:

   - If c_j is not a singleton (i.e., $c_j = c_l$ for some $l \neq j$)

     (a) let c_j^* be a newly created clone.

     (b) propose a new clonal tree, $\mathcal{T}^* \sim q_T(\mathcal{T}^*|\mathcal{T})$, by adding the new clone c_j^* to $\mathcal{T}$. q_T is the proposal distribution that adds a new leaf to the clonal phylogeny.

     (c) Sample genotype vector for the new clone, $G_{c_j^*} \sim \mathcal{F}(G_{c_j^*}|\mathcal{T}^*, \boldsymbol{G}^*_{\setminus c_j^*}, \mathcal{M}_\lambda)$. $\boldsymbol{G}^*_{\setminus c_j^*}$ is the clonal genotype matrix excluding the genotype vector for clone c_j^*. New clonal genotype matrix after sampling $G_{c_j^*}$ is denoted by $\boldsymbol{G}^*$.

     (d) compute acceptance ratio $a(c_j^*, c_j)$ as follows:

$$a(c_j^*, c_j) = min\left[1, \frac{\alpha_0}{m-1}\frac{E(D[j]|G_{c_j^*}, \alpha, \beta)}{E(D[j]|G_{c_j}, \alpha, \beta)}\frac{F(\boldsymbol{G}^*|\mathcal{T}^*, \mathcal{M}_\lambda)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)}\frac{P(\boldsymbol{c}^*|\alpha_0)}{P(\boldsymbol{c}|\alpha_0)}\frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})}\frac{q_T(\mathcal{T}|\mathcal{T}^*)}{q_T(\mathcal{T}^*|\mathcal{T})}J_q\right] \tag{9}$$

     J_q is the jacobian. $D[j]$ is the j^{th} column of observed genotype matrix.

     (e) Set the new c_j to this c_j^* with probability $a(c_j^*, c_j)$

     (f) If new c_j is set to c_j^*,

     - Set $\boldsymbol{G} = \boldsymbol{G}^*$, $\mathcal{T} = \mathcal{T}^*$

   - Otherwise, when c_j is a singleton,

     (a) Sample c_j^* from $\boldsymbol{c}_{-j}$, choosing $c_j^* = c$ with probability $\frac{n_c}{m-1}$.

     (b) Propose a new clonal tree, $\mathcal{T}^* \sim q_T(\mathcal{T}^*|\mathcal{T})$, by removing the clone c_j from $\mathcal{T}$.

     (c) Propose new clonal genotype matrix $\boldsymbol{G}^*$, by removing G_{c_j} from $\boldsymbol{G}$.

     (d) compute acceptance ratio $a(c_j^*, c_j)$ as follows:

$$a(c_j^*, c_j) =$$
$$min\left[1, \frac{m-1}{\alpha_0}\frac{E(D[j]|G_{c_j^*}, \alpha, \beta)}{E(D[j]|G_{c_j}, \alpha, \beta)}\frac{F(\boldsymbol{G}^*|\mathcal{T}^*, M)}{F(\boldsymbol{G}|\mathcal{T}, M)}\frac{P(\boldsymbol{c}^*|\alpha_0)}{P(\boldsymbol{c}|\alpha_0)}\frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})}\frac{q_T(\mathcal{T}|\mathcal{T}^*)}{q_T(\mathcal{T}^*|\mathcal{T})}J_q\right] \tag{10}$$

(e) Set the new c_j to this c_j^* with probability $a(c_j^*, c_j)$.

(f) If new c_j is set to c_j^*,

– Set $\boldsymbol{G} = \boldsymbol{G}^*$, $\mathcal{T} = \mathcal{T}^*$

- If the new c_j is not set to c_j^*, it is the same as the old c_j. $\boldsymbol{G}$ and $\mathcal{T}$ remains same.

2. For $j = 1, \ldots, m$, update c_j as follows:

   - If c_j is a singleton, do nothing.
   - Otherwise, choose a new value for c_j from $\{c_1, \ldots, c_m\}$ using the following probabilities:

$$P(c_j = c|\boldsymbol{c}_{-j}, D[j], \boldsymbol{G}, \alpha, \beta) \propto \frac{n_c}{m-1}E(D[j]|G_c, \alpha, \beta)$$

Sample clonal phylogeny and evolution model parameters:

Sample new clonal phylogeny $\mathcal{T}^*$ and new set of values for parameters of model of evolution, $\mathcal{M}_\lambda^*$ from the joint conditional posterior distribution, $\mathcal{P}_{\mathcal{T},\mathcal{M}_\lambda}(\mathcal{T}^*, \mathcal{M}_\lambda^*|\mathcal{T}, \mathcal{M}_\lambda, \boldsymbol{G}, a_M, b_M)$

$$\mathcal{T}^*, \mathcal{M}_\lambda^* \sim \mathcal{P}_{\mathcal{T},\mathcal{M}_\lambda}(\mathcal{T}^*, \mathcal{M}_\lambda^*|\mathcal{T}, \mathcal{M}_\lambda, \boldsymbol{G}, a_M, b_M)$$

Sample clonal genotypes:

For $k = 1, ..., |\boldsymbol{c}|$

- Sample clonal genotype G_k for each clone as follows:
  For $i = 1, ..., n$, sample G_{ki} from the following distribution

$$G_{ki} \propto \mathcal{F}(G_{ki}|T, \boldsymbol{G}_{-ki}, M) \times \prod_{j|c_j=k} E(D_{ij}|G_{ki})$$

Sample error rates:

1. Sample $\alpha \sim \mathcal{P}_\alpha(\alpha|\boldsymbol{D}, \boldsymbol{c}, \boldsymbol{G}, \beta, a_\alpha, b_\alpha) \sim E(\boldsymbol{D}|\boldsymbol{c}, \boldsymbol{G}, \beta, \alpha)P(\alpha|a_\alpha, b_\alpha)$ using rejection sampling.

2. Sample $\beta \sim \mathcal{P}_\beta(\beta|\boldsymbol{D}, \boldsymbol{c}, \boldsymbol{G}, \alpha, a_\beta, b_\beta) \sim E(\boldsymbol{D}|\boldsymbol{c}, \boldsymbol{G}, \beta, \alpha)P(\beta|a_\beta, b_\beta)$ using rejection sampling.

Sample concentration parameter:

Sample $\alpha_0^t \sim p(\alpha_0|m, |\boldsymbol{c}|, a, b)$ based on the method described in [5] assuming the prior distribution for α_0 is $Gamma(a, b)$.

1.1.9.1 Algorithm For Sampling Cluster Indicators

Partial reversible-jump MCMC partial Gibbs updates are used for sampling the cluster indicators for cells as outlined above. In the partial reversible-jump MCMC steps, new clusters are assigned to cells based on an acceptance ratio. The calculation of acceptance ratio involves the calculation of likelihood ratio, prior ratio, proposal ratio and jacobian. Below, we describe how each of these terms are computed.

1.1.9.1.1 Likelihood Ratio

The likelihood ratio, L_r is defined by:

$$L_r = \frac{E(D[j]|G_{c_j^*}, \alpha, \beta)}{E(D[j]|G_{c_j}, \alpha, \beta)} \tag{11}$$

In Eq. (11), c_j^* and c_j are the new and old cluster indicators for cell j respectively. The values in the numerator and the denominator can be calculated by:

$$E(D[j]|G_{c_j=c}, \alpha, \beta) = \prod_{i=1}^{n} E(D_{ij}|G_{ci}, \alpha, \beta) \tag{12}$$

$E(D_{ij}|G_{ci}, \alpha, \beta)$ is given by the error model distribution as shown in Supplemental Table S5 or Supplemental Table S4.

1.1.9.1.2 Prior Ratio

The prior ratio, P_r is given by:

$$P_r = \frac{F(\boldsymbol{G}^*|\mathcal{T}^*, \mathcal{M}_\lambda)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)} \frac{P(\boldsymbol{c}^*|\alpha_0)}{P(\boldsymbol{c}|\alpha_0)} \frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})} \tag{13}$$

and is a product of three ratios from three prior distributions. The first ratio, $\frac{F(\boldsymbol{G}^*|\mathcal{T}^*, \mathcal{M}_\lambda)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)}$ can be computed using Eq. (7). The second ratio, $\frac{P(\boldsymbol{c}^*|\alpha_0)}{P(\boldsymbol{c}|\alpha_0)}$ can be computed using Eq. (8). The third ratio is the ratio of prior probabilities on clonal phylogeny. Let us assume, the number of clones based on the new set of cluster indicators is, $|\boldsymbol{c}^*| = K$. For non-singleton cells (i.e., $c_j = c_l$ for some $l \neq j$), when a new leaf is added to the clonal phylogeny, the third ratio is defined by

$$\frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})} = \frac{K-1}{(K-2)(2K-3)} \frac{f(\nu_1)f(\nu_2)f(\nu^*)}{f(\nu_1+\nu_2)} \tag{14}$$

In Eq. (14), ν_1 and ν_2 are the new branch lengths created by adding a new leaf to the branch of length $\nu = \nu_1 + \nu_2$ and ν^* is the branch length assigned to the branch connected to the new leaf. $f(\nu)$ is the edge length prior density evaluated at any branch of length ν. All other edge lengths maintain the same values before and after adding the new leaf, so all other terms in the prior ratio cancel each other.

For singleton cells, when an existing leaf is removed from the clonal phylogeny, the third ratio is defined by

$$\frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})} = \frac{(K-1)(2K-1)}{K} \frac{f(\nu_1+\nu_2)}{f(\nu_1)f(\nu_2)f(\nu^*)} \tag{15}$$

In Eq. (15), $\nu = \nu_1 + \nu_2$ is the branch length of the new branch after removing the leaf associated with branch of length ν^*. As a result of the removal of this leaf, two branches of length ν_1 and ν_2 are merged into one branch of length $\nu_1 + \nu_2$. All other edge lengths maintain the same values before and after removal of the leaf, so all other terms in the prior ratio cancel each other. For the distribution on branch lengths (f), we use exponential distribution.

1.1.9.1.3 Proposal Ratio and Jacobian

The proposal or Hastings ratio, Q_r, is given by:

$$Q_r = \frac{q_T(\mathcal{T}|\mathcal{T}^*)}{q_T(\mathcal{T}^*|\mathcal{T})} \tag{16}$$

where q_T is the proposal distribution. We have two moves corresponding to adding a new leaf and removing an existing leaf respectively. The moves and their corresponding proposal ratio are described below.

Add Clone: This move is performed when a new clonal cluster is created for a cell. This results in adding a new leaf to the existing clonal phylogeny and the new leaf corresponds to the new cluster. As a result, this move adds new parameters to the model. One branch of the existing phylogeny is chosen at random. Let us assume that the length of the chosen branch is ν. A new node is created on this branch which serves as the parent of the new clone/leaf to be added. As a result, the existing branch gets divided into two new branches of lengths ν_1 and ν_2. To choose the lengths of these new branches, we generate a uniformly random number, w_1 between 0 and 1, $w_1 \sim U(0,1)$ and the branch lengths are set as $\nu_1 = \nu * w_1$ and $\nu_2 = \nu * (1 - w_1)$. To propose the length of the branch that connects the new leaf to its parent, we generate another uniform random number, $w_2 \sim U(0,1)$ and it is transformed into a random deviate from the edge length prior distribution, $\nu^* = -\frac{1}{\theta}\ln(1 - w_2)$.

The Hastings ratio for adding a new clone to the clonal phylogeny is the probability of proposing a remove clone move that exactly reverses the proposed add clone move, divided by the probability of proposing the add clone move itself. Proposing an add clone move involves the following steps:

1. Choose to perform the add clone move
2. Choose an existing branch of the phylogeny
3. Divide the branch into two branches
4. Choose a length for the newly created edge

The probability of the first step is $\frac{\alpha_0}{m+\alpha_0-1}$, as the new clone is created with this probability. The probability of the second step is $\frac{1}{n_e}$, where n_e is the number of branches in the phylogeny before the move. If we assume that the number of clones based on the new set of cluster indicators is, $|\boldsymbol{c}^*| = K$, then $n_e = 2K - 3$. To divide the branch into two branches, we generate a uniform random variate w_1, so the third step has no effect on the probability of Add Clone move because the value w_1 has Uniform probability density 1.0, similarly the fourth move does not have any effect on the probability of Add Clone move as we generate another uniform random deviate w_2.

Proposing the corresponding Remove Clone move involves two steps:

1. Choose to perform Remove Clone move
2. Choose the leaf in the phylogeny to remove to restore the phylogeny that existed before the Add Clone move.

The probability of the first step is $\frac{1}{m+\alpha_0-1}$, as size of the new clone is 1. The probability of the second step is $\frac{1}{K}$, where K is the number of leaves in the phylogeny after Add Clone move. Therefore, the Hastings ratio is given by:

$$\begin{aligned} \text{Hastings ratio for Add Clone move} &= \frac{(\frac{1}{m+\alpha_0-1})(\frac{1}{K})}{(\frac{\alpha_0}{m+\alpha_0-1})(\frac{1}{2K-3})} \\ &= \frac{2K-3}{\alpha_0 * K} \end{aligned} \tag{17}$$

The Jacobian term for this move is given by:

$$\begin{aligned} J_q &= \begin{vmatrix} \frac{\partial \nu_1}{\partial \nu} & \frac{\partial \nu_1}{\partial w_1} & \frac{\partial \nu_1}{\partial w_2} \\ \frac{\partial \nu_2}{\partial \nu} & \frac{\partial \nu_2}{\partial w_1} & \frac{\partial \nu_2}{\partial w_2} \\ \frac{\partial \nu^*}{\partial \nu} & \frac{\partial \nu^*}{\partial w_1} & \frac{\partial \nu^*}{\partial w_2} \end{vmatrix} \\ &= \begin{vmatrix} w_1 & \nu & 0 \\ 1-w_1 & -\nu & 0 \\ 0 & 0 & \frac{1}{1-w_2} \end{vmatrix} \\ &= \frac{\nu}{1-w_2} \end{aligned} \tag{18}$$

Remove Clone: This move is performed when an existing clonal cluster is removed. This results in removing a leaf from the existing clonal phylogeny. As a result, this move removes some parameters from the model. The leaf to be removed is chosen and removed from the phylogeny, the associated branch is also removed. The parent node of the leaf is also removed, as a result two branches of lengths ν_1 and ν_2 get merged into a single branch of length $\nu = \nu_1 + \nu_2$.

Hastings ratio for the Remove Clone move is given by the probability of proposing an Add Clone move divided by the probability of the Remove Clone move and is calculated as follows:

$$\begin{aligned} \text{Hastings ratio for Remove Clone move} &= \frac{(\frac{\alpha_0}{m+\alpha_0-1})(\frac{1}{2K-1})}{(\frac{1}{m+\alpha_0-1})(\frac{1}{K+1})} \\ &= \frac{\alpha_0 * (K+1)}{2K-1} \end{aligned} \tag{19}$$

The Jacobian term for this move is given by:

$$\begin{aligned} J_q &= \begin{vmatrix} \frac{\partial \nu}{\partial \nu_1} & \frac{\partial \nu}{\partial \nu_2} & \frac{\partial \nu}{\partial \nu^*} \\ \frac{\partial w_1}{\partial \nu_1} & \frac{\partial w_1}{\partial \nu_2} & \frac{\partial w_1}{\partial \nu^*} \\ \frac{\partial w_2}{\partial \nu_1} & \frac{\partial w_2}{\partial \nu_2} & \frac{\partial w_2}{\partial \nu^*} \end{vmatrix} \\ &= \begin{vmatrix} 1 & 1 & 0 \\ \frac{1}{\nu} & -\frac{1}{\nu} & 0 \\ 0 & 0 & e^{-\nu^*} \end{vmatrix} \\ &= \frac{e^{-\nu^*}}{\nu} \end{aligned} \tag{20}$$

1.1.9.2 Algorithm For Sampling Clonal Phylogeny and Evolution Model Parameters

We designed a Metropolis-Hastings [10] sampler for sampling the clonal phylogeny and evolution model parameters from the joint conditional posterior given by:

$$\mathcal{P}_{\mathcal{T},\mathcal{M}_\lambda}(\mathcal{T}^*, \mathcal{M}_\lambda^* | \boldsymbol{G}, a_M, b_M) \propto F(\boldsymbol{G} | \mathcal{T}^*, \mathcal{M}_\lambda^*) p(\mathcal{T}^*) p(\mathcal{M}_\lambda^* | a_M, b_M) \tag{21}$$

We consider two different types of moves to explore the joint $\mathcal{T}, \mathcal{M}_\lambda$ space. In tree changing moves, a new clonal phylogenetic tree, $\mathcal{T}^*$ is proposed from current state $\mathcal{T}$. In parameter changing moves, a new value of the parameter, $\mathcal{M}_\lambda^*$ is proposed from the current parameter value $\mathcal{M}_\lambda$. The proposed configuration is accepted or rejected based on an acceptance ratio. The acceptance ratio for proposing a new clonal phylogenetic tree is given by:

$$\rho_T = min\left\{1, \frac{F(\boldsymbol{G}|\mathcal{T}^*, \mathcal{M}_\lambda)p(\mathcal{T}^*)q_T(\mathcal{T}|\mathcal{T}^*)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)p(\mathcal{T})q_T(\mathcal{T}^*|\mathcal{T})}\right\} \tag{22}$$

In Eq. (22), the likelihood ratio, $\frac{F(\boldsymbol{G}|\mathcal{T}^*, \mathcal{M}_\lambda)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)}$ is computed using Felsenstein's pruning algorithm [6]. q_T denotes the proposal distribution for proposing a new phylogeny from the current phylogeny. Here, we use a combination of branch change (alter branch lengths) and branch-rearrangement (alter the tree topology) proposals as used in [29]. The prior ratio is computed using uniform prior for topology and exponential prior for branch lengths.

The acceptance ratio for proposing a new parameter value is given by:

$$\rho_{\mathcal{M}_\lambda} = min\left\{1, \frac{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda^*)p(\mathcal{M}_\lambda^*|a_M, b_M)q_{\mathcal{M}_\lambda}(\mathcal{M}_\lambda|\mathcal{M}_\lambda^*)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)p(\mathcal{M}_\lambda^*|a_M, b_M)q_{\mathcal{M}_\lambda}(\mathcal{M}_\lambda^*|\mathcal{M}_\lambda)}\right\} \tag{23}$$

In Eq. (23), the likelihood is calculated in the same way as for Eq. (22). $q_{\mathcal{M}_\lambda}$ is the proposal distribution. The parameters, λ_r and λ_l are beta distributed variables. For each of these parameters, the next value is proposed from a normal distribution centered at the current value. The standard deviation is chosen so that a wide range of values are covered. The algorithm is shown in Algorithm 1.

1.1.9.3 Algorithm For Sampling Clonal Genotypes

The genotype of each clone is sampled by keeping the genotypes of other clones fixed. Genotype of each position can be sampled independently. The clonal genotype for clone k, G_k, where $k \in \{1, \ldots, |\boldsymbol{c}|\}$ is sampled from the conditional posterior distribution given by:

$$G_k \sim \mathcal{P}_G(G_k|D_{j|c_j=k}, \boldsymbol{G}_{\setminus k}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta) \tag{24}$$

Algorithm 1: Algorithm for sampling clonal phylogeny and evolution model parameters. $\mathcal{T}^s$ is the starting tree. $\mathcal{M}_\lambda^s$ is the starting value of model parameters. The algorithm runs for n_{iter} iterations. With probability p_λ, model parameters are updated.

Input: $\boldsymbol{G}, \mathcal{T}^s, \mathcal{M}_\lambda^s, n_{iter}, p_\lambda$
Output: $\mathcal{T}^*, \mathcal{M}_\lambda^*$
Initialization: $\mathcal{T}^0 \leftarrow \mathcal{T}^s, \mathcal{M}_\lambda^0 \leftarrow \mathcal{M}_\lambda^s$
for $i = 1...n_{iter}$ **do**
  $\mathcal{T} \leftarrow \mathcal{T}^{i-1}, \mathcal{M}_\lambda \leftarrow \mathcal{M}_\lambda^{i-1}$
  Sample $r \sim U(0,1)$
  if $r \leq p_\lambda$ **then**
    Sample $\mathcal{M}'_\lambda \sim q_{\mathcal{M}_\lambda}(\mathcal{M}'_\lambda|\mathcal{M}_\lambda)$
    Compute $\rho_{\mathcal{M}_\lambda} = min\left\{1, \frac{F(\boldsymbol{G}|\mathcal{T},\mathcal{M}'_\lambda)p(\mathcal{M}'_\lambda|a_M,b_M)q_{\mathcal{M}_\lambda}(\mathcal{M}_\lambda|\mathcal{M}'_\lambda)}{F(\boldsymbol{G}|\mathcal{T},\mathcal{M}_\lambda)p(\mathcal{M}'_\lambda|a_M,b_M)q_{\mathcal{M}_\lambda}(\mathcal{M}'_\lambda|\mathcal{M}_\lambda)}\right\}$
    Accept $\mathcal{M}'_\lambda$ with probability $\rho_{\mathcal{M}_\lambda}$
    $\mathcal{M}_\lambda^i \leftarrow \mathcal{M}'_\lambda, \mathcal{T}^i \leftarrow \mathcal{T}$
  else
    Sample $\mathcal{T}' \sim q_T(\mathcal{T}'|\mathcal{T})$
    Compute $\rho_T = min\left\{1, \frac{F(\boldsymbol{G}|\mathcal{T}',\mathcal{M}_\lambda)p(\mathcal{T}')q_T(\mathcal{T}|\mathcal{T}')}{F(\boldsymbol{G}|\mathcal{T},\mathcal{M}_\lambda)p(\mathcal{T})q_T(\mathcal{T}'|\mathcal{T})}\right\}$
    Accept $\mathcal{T}'$ with probability ρ_T
    $\mathcal{M}_\lambda^i \leftarrow \mathcal{M}_\lambda, \mathcal{T}^i \leftarrow \mathcal{T}'$
$\mathcal{T}^* \leftarrow \mathcal{T}^{n_{iter}}, \mathcal{M}_\lambda^* \leftarrow \mathcal{M}_\lambda^{n_{iter}}$
return $\mathcal{T}^*, \mathcal{M}_\lambda^*$

In Eq. (24), $\boldsymbol{G}_{\setminus k}$ denotes the genotypes of other clones and $D_{j|c_j=k}$ denotes the observed genotypes of the cells assigned to clone k. Clonal genotype G_k is a vector of length n and records the genotype state for n mutation loci. Genotype for locus i is sampled from a categorical distribution defined by

$$G_{ki} \propto \mathcal{F}(G_{ki}|\mathcal{T}, \boldsymbol{G}_{-ki}, \mathcal{M}_\lambda) \times \prod_{j|c_j=k} E(D_{ij}|G_{ki}, \alpha, \beta) \tag{25}$$

For $G_{ki} \in g_t$, $\mathcal{F}(G_{ki}|\mathcal{T}, \boldsymbol{G}_{-ki}, \mathcal{M}_\lambda)$ is calculated using Felsenstein's pruning algorithm and $E(D_{ij}|G_{ki}, \alpha, \beta)$ is given by the error model distribution as shown in Supplemental Table S5 or Supplemental Table S4.

1.1.9.4 Algorithm For Sampling Error Rates

Rejection sampling [3] is used for sampling the value of error rates α and β from their corresponding conditional posterior distributions.

1.1.9.4.1 False Positive Rate

The conditional posterior distribution from which α is sampled, is given by:

$$\alpha \sim \mathcal{P}_{\alpha}(\alpha|\boldsymbol{D}, \boldsymbol{c}, \boldsymbol{G}, \beta, a_{\alpha}, b_{\alpha}) \sim E(\boldsymbol{D}|\boldsymbol{c}, \boldsymbol{G}, \beta, \alpha)P(\alpha|a_{\alpha}, b_{\alpha}) \tag{26}$$

By varying α for a grid of values between 0.001 to 1, we first compute the maximum of the posterior distribution. Based on this maximum value, we create an envelope function for the range of values of α and this serves as the proposal distribution using which we sample a new value of α using rejection sampling.

1.1.9.4.2 False Negative Rate

The conditional posterior distribution from which β is sampled, is given by:

$$\beta \sim \mathcal{P}_{\beta}(\beta|\boldsymbol{D}, \boldsymbol{c}, \boldsymbol{G}, \alpha, a_{\beta}, b_{\beta}) \sim E(\boldsymbol{D}|\boldsymbol{c}, \boldsymbol{G}, \beta, \alpha)P(\beta|a_{\beta}, b_{\beta}) \tag{27}$$

By varying β for a grid of values between 0.001 to 1, we first compute the maximum of the posterior distribution. Based on this maximum value, we create an envelope function for the range of values of β and this serves as the proposal distribution using which we sample a new value of β using rejection sampling.

1.2 Doublet Model of SiCloneFit

The singlet model of SiCloneFit is extended to handle cases where some data points result from measuring two cells. We assume that the occurrence of doublets is a rare event, and simultaneous processing of more than two cells is extremely rare. Thus we only focus on the extension to two cells, or doublets. We also assume that simultaneous measurement of higher numbers of cells occurs sufficiently infrequently resulting in negligible impact.

To model multiple cell measurements, we need to define the expected genotype state when two cells are measured together. To do that for ternary data type, we use the binary operator $\oplus$ introduced in SiFit [29] and defined in Section 4. For presence/absence data such as a binary representation of SNVs, we can use a *logical or* to define $\oplus$.

1.2.1 Model Overview

The probabilistic graphical model for the extended SiCloneFit model for handling doublets is shown in Supplemental Fig. S2. The new variables introduced in this model are explained in Supplemental Table S7.

1.2.2 Model Description

To model doublets, we introduce a new variable Y_j corresponding to single cell j. Y_j is a Bernoulli variable that takes the value 0 if cell j is a singlet and the value 1 when cell j is a doublet. The probability of sampling a doublet is modeled by the variable δ, which is again another Beta distributed variable with hyper-parameters a_δ, b_δ. Instead of a single cluster indicator for each cell as defined in the SiCloneFit model (Supplemental Fig. S1), in the extended model, we introduce two cluster indicators for each cell. c_j^1 is the primary cluster indicator for cell j with a Chinese restaurant process prior based on hyper-parameter α_0, whereas c_j^2 is a secondary cluster indicator for cell j that can uniformly take values in the range $\{1, \ldots, |\boldsymbol{c^1}|\}$. If $Y_j = 1$, c_j^2 denotes the clone of origin of the cell that forms a doublet

by merging with cell j from clone c_j^1. The extended model is defined as

$$\begin{aligned}
\alpha|a_\alpha, b_\alpha &\sim Beta(a_\alpha, b_\alpha) \\
\beta|a_\beta, b_\beta &\sim Beta(a_\beta, b_\beta) \\
\delta|a_\delta, b_\delta &\sim Beta(a_\delta, b_\delta) \\
Y_j|\delta &\sim Bernoulli(Y_j|\delta) \\
\alpha_0 &\sim Gamma(a, b) \\
c_j^1|\alpha_0 &\sim CRP(\alpha_0) \\
c_j^2|c_j^1 &\sim \mathcal{U}\{1, |\boldsymbol{c}^1|\} \\
\mathcal{T} &\sim T_{prior}(|\boldsymbol{c}^1|) \\
\mathcal{M}_\lambda|a_{M_\lambda}, b_{M_\lambda} &\sim Beta(a_{M_\lambda}, b_{M_\lambda}) \\
G_{ki}|\mathcal{T}, \mathcal{M}_\lambda &\sim F(G_{ki}|\mathcal{T}, \mathcal{M}_\lambda) \\
g_{ji}|\boldsymbol{G}, c_j^1, c_j^2, Y_j &= \begin{cases} G_{c_j^1 i} & Y_j = 0 \\ G_{c_j^1 i} \oplus G_{c_j^2 i} & Y_j = 1 \end{cases} \\
D_{ij}|c_j^1, c_j^2, Y_j, G_{c_j^1 i}, G_{c_j^2 i}, \alpha, \beta &\sim E(D_{ij}|g_{ji}, \alpha, \beta)
\end{aligned}$$

1.2.3 Posterior Distribution

The posterior distribution for the doublet model of SiCloneFit, $\mathcal{P}$, is given by

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \delta, \alpha_0 | \boldsymbol{D}, a_\alpha, b_\alpha, a_\beta, b_\beta, a_\delta, b_\delta, a_M, b_M, a, b) \propto \\
&\quad P(\boldsymbol{D}|\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \delta, \alpha_0, a_\alpha, b_\alpha, a_\beta, b_\beta, a_\delta, b_\delta, a_M, b_M, a, b) \times \\
&\quad P(\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \delta, \alpha_0 | a_\alpha, b_\alpha, a_\beta, b_\beta, a_\delta, b_\delta, a_M, b_M, a, b) \\
&\quad = E(\boldsymbol{D}|\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \alpha, \beta) F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda) P(\boldsymbol{c^1}|\alpha_0) P(\boldsymbol{c^2}|\boldsymbol{c^1}) P(\mathcal{T}) \\
&\quad P(\boldsymbol{Y}|\delta) P(\alpha|a_\alpha, b_\alpha) P(\beta|a_\beta, b_\beta) P(\delta|a_\delta, b_\delta) P(\mathcal{M}_\lambda|a_M, b_M) P(\alpha_0|a, b) \quad (28)
\end{aligned}$$

The hidden variables that we want to estimate from this model are

1. $\boldsymbol{c^1}$, a vector containing the primary clone indicator for each cell

2. $\boldsymbol{c^2}$, a vector containing the secondary clone indicator for cells that are inferred as doublets

3. $\boldsymbol{Y}$, a vector containing the indicator for each cell that denotes if the cell is a doublet or singlet

4. $\boldsymbol{G}$, a $K \times n$ clonal genotype matrix, where G_k denotes the genotype of clone k, $K = |\boldsymbol{c^1}|$

5. $\mathcal{T}$, the clonal phylogeny, representing the genealogical relationships between the clones

6. $\mathcal{M}_\lambda$, parameters of the model of evolution

7. α, false positive rate

8. β, false negative rate

9. δ, doublet rate

The number of clones is implicitly defined by the vector $\boldsymbol{c^1}$. The posterior probability is a product of likelihood function and prior. The likelihood function is described in Section 1.2.4. The prior distributions for the same variables as in the singlet model are already explained in Section 1.1.7 and the prior distributions for the new variables are described in Section 1.2.5.

1.2.4 Likelihood Function

The likelihood function for the extended SiCloneFit model is given by

$$\begin{aligned} P(\boldsymbol{D}|\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \delta, \alpha_0, a_\alpha, b_\alpha, a_\beta, b_\beta, a_\delta, b_\delta, a_M, b_M, a, b) &= E(\boldsymbol{D}|\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \alpha, \beta) \\ &= \prod_{i=1}^{n} \prod_{j=1}^{m} E(D_{ij}|g_{ji}, \alpha, \beta) \end{aligned} \tag{29}$$

where g_{ji} is given by Eq. (30)

$$g_{ji} = \begin{cases} G_{c_j^1 i} & Y_j = 0 \\ G_{c_j^1 i} \oplus G_{c_j^2 i} & Y_j = 1 \end{cases} \tag{30}$$

$E(D_{ij}|g_{ji}, \alpha, \beta)$ is given by the error model distribution as shown in Supplemental Table S4 and Supplemental Table S5.

1.2.5 Prior Distributions

The complete prior of the extended SiCloneFit model is given by

$$\begin{aligned} &P(\boldsymbol{c^1}, \boldsymbol{c^2}, \boldsymbol{Y}, \boldsymbol{G}, \mathcal{T}, \mathcal{M}_\lambda, \alpha, \beta, \delta, \alpha_0 | a_\alpha, b_\alpha, a_\beta, b_\beta, a_\delta, b_\delta, a_M, b_M, a, b) \\ &\quad = F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda) P(\boldsymbol{c^1}|\alpha_0) P(\boldsymbol{c^2}|\boldsymbol{c^1}) P(\mathcal{T}) P(\boldsymbol{Y}|\delta) P(\alpha, \beta, \delta, \mathcal{M}_\lambda, \alpha_0|\mathcal{H}) \end{aligned} \tag{31}$$

where,

$$P(\alpha, \beta, \delta, \mathcal{M}_\lambda, \alpha_0|\mathcal{H}) = P(\alpha|a_\alpha, b_\alpha) P(\beta|a_\beta, b_\beta) P(\delta|a_\delta, b_\delta) P(\mathcal{M}_\lambda|a_M, b_M) P(\alpha_0|a, b)$$

$\mathcal{H}$ denotes the set of hyperparameters, $\mathcal{H} = \{a_\alpha, b_\alpha, a_\beta, b_\beta, a_\delta, b_\delta, a_M, b_M, a, b\}$. The prior distributions $F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)$, $P(\mathcal{T})$, $P(\alpha|a_\alpha, b_\alpha)$, $P(\beta|a_\beta, b_\beta)$, $P(\mathcal{M}_\lambda|a_M, b_M)$, $P(\alpha_0|a, b)$ have been described in Section 1.1.7. $P(\boldsymbol{c^1}|\alpha_0)$ denotes the prior probability of partitioning m single cells into $|\boldsymbol{c^1}|$ clusters under a CRP with concentration parameter α_0 as described in Section 1.1.7.

$P(\boldsymbol{c^2}|\boldsymbol{c^1})$ denotes the prior distribution on the secondary cluster indicators given the primary cluster indicators. We use a uniform distribution as the prior for c_j^2, the secondary cluster indicator for cell j. The value of c_j^2 is drawn uniformly from the range $\{1, \ldots, |\boldsymbol{c^1}|\}$, $|\boldsymbol{c^1}|$ is the number of clusters implicitly defined by $\boldsymbol{c^1}$.

Y_j is a Bernoulli variable that indicates whether cell j is a doublet or a singlet. The Bernoulli distribution is parameterized by δ, the doublet rate, which gives the success probability.

We assume δ to be a Beta distributed variable as it denotes the probability of sampling a doublet and takes value between 0 and 1.

1.2.6 Inference

We extended the Gibbs sampler designed for the basic SiCloneFit model (Supplemental Fig. S1) to obtain a Markov Chain Monte Carlo sampler for the extended SiCloneFit model (Supplemental Fig. S2). The sampler is outlined below.

1.2.7 Partial Reversible-jump MCMC Partial Gibbs Sampling Algorithm

Given $\alpha_0^{(t-1)}$, $\{c_j^{1(t-1)}\}_{j=1}^m$, $\{c_j^{2(t-1)}\}_{j=1}^m$, $\{Y_j\}_{j=1}^m$, $\{G_k^{(t-1)}\}_{k=1}^{|\boldsymbol{c^1}|}$, $\mathcal{T}^{(t-1)}$, $\mathcal{M}_\lambda^{(t-1)}$, $\alpha^{(t-1)}$, $\beta^{(t-1)}$, and $\delta^{(t-1)}$ from the previous iteration, we need to sample a new set of these parameters. $t-1$ denotes the previous iteration.
Set

- $\boldsymbol{c^1} = \boldsymbol{c}^{1(t-1)}$, $\alpha_0 = \alpha_0^{(t-1)}$
- $\boldsymbol{c^2} = \boldsymbol{c}^{2(t-1)}$
- $\boldsymbol{Y} = \boldsymbol{Y}^{(t-1)}$
- $\boldsymbol{G} = \{G_k^{(t-1)}\}_{k=1}^{|\boldsymbol{c^1}|}$
- $\mathcal{T} = \mathcal{T}^{(t-1)}$, $\mathcal{M}_\lambda = \mathcal{M}_\lambda^{(t-1)}$
- $\alpha = \alpha^{(t-1)}$, $\beta = \beta^{(t-1)}$, $\delta = \delta^{(t-1)}$

Sample primary cluster indicators:

1. For $j = 1, \ldots, m$, update c_j^1 as follows:
   - If c_j^1 is not a singleton (i.e., $c_j^1 = c_l^1$ for some $l \neq j$)
     (a) let c_j^{1*} be a newly created clone.
     (b) propose a new clonal tree, $\mathcal{T}^* \sim q_T(\mathcal{T}^*|\mathcal{T})$, by adding the new clone c_j^{1*} to $\mathcal{T}$. q_T is the proposal distribution that adds a new leaf to the clonal phylogeny.

(c) Sample genotype vector for the new clone, $G_{c_j^{1*}} \sim \mathcal{F}(G_{c_j^{1*}}|\mathcal{T}^*, \boldsymbol{G}^*_{\setminus c_j^{1*}}, \mathcal{M}_\lambda)$. $\boldsymbol{G}^*_{\setminus c_j^{1*}}$ is the clonal genotype matrix excluding the genotype vector for clone c_j^{1*}. New clonal genotype matrix after sampling $G_{c_j^{1*}}$ is denoted by $\boldsymbol{G}^*$.

(d) compute acceptance ratio $a(c_j^{1*}, c_j^1)$ as follows:

$$a(c_j^{1*}, c_j^1) = min[1, r]$$
$$r = \frac{\alpha_0}{m-1} \frac{E(D[j]|G_{c_j^{1*}}, G_{c_j^2}, Y_j, \alpha, \beta)}{E(D[j]|G_{c_j^1}, G_{c_j^2}, Y_j, \alpha, \beta)} \frac{F(\boldsymbol{G}^*|\mathcal{T}^*, \mathcal{M}_\lambda)}{F(\boldsymbol{G}|\mathcal{T}, \mathcal{M}_\lambda)} \frac{P(\boldsymbol{c}^{1*}|\alpha_0)}{P(\boldsymbol{c}^1|\alpha_0)} \frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})} \frac{q_T(\mathcal{T}|\mathcal{T}^*)}{q_T(\mathcal{T}^*|\mathcal{T})} J_q \tag{32}$$

J_q is the jacobian. $D[j]$ is the j^{th} column of observed genotype matrix.

(e) Set the new c_j^1 to this c_j^{1*} with probability $a(c_j^{1*}, c_j^1)$

(f) If new c_j^1 is set to c_j^{1*},

- Set $\boldsymbol{G} = \boldsymbol{G}^*$, $\mathcal{T} = \mathcal{T}^*$

- Otherwise, when c_j^1 is a singleton,

(a) Sample c_j^{1*} from $\boldsymbol{c}^1_{-j}$, choosing $c_j^{1*} = c$ with probability $\frac{n_c}{m-1}$.

(b) Propose a new clonal tree, $\mathcal{T}^* \sim q_T(\mathcal{T}^*|\mathcal{T})$, by removing the clone c_j^1 from $\mathcal{T}$.

(c) Propose new clonal genotype matrix $\boldsymbol{G}^*$, by removing $G_{c_j^1}$ from $\boldsymbol{G}$.

(d) Propose a new secondary cluster indicator vector $\boldsymbol{c}^{2*}$ in which for cells $l \mid c_l^2 = c_j^1$, set $c_l^2 = c_j^{1*}$. Secondary cluster indicators for other cells remain the same.

(e) compute acceptance ratio $a(c_j^{1*}, c_j^1)$ as follows:

$$a(c_j^{1*}, c_j^1) = min[1, r]$$
$$r = \frac{m-1}{\alpha_0} \frac{E(D[j]|G_{c_j^{1*}}, G_{c_j^2}, Y_j, \alpha, \beta)}{E(D[j]|G_{c_j^1}, G_{c_j^2}, Y_j, \alpha, \beta)} \frac{F(\boldsymbol{G}^*|\mathcal{T}^*, M)}{F(\boldsymbol{G}|\mathcal{T}, M)} \frac{P(\boldsymbol{c}^{1*}|\alpha_0)}{P(\boldsymbol{c}^1|\alpha_0)} \frac{T_{prior}(\mathcal{T}^*)}{T_{prior}(\mathcal{T})} \frac{q_T(\mathcal{T}|\mathcal{T}^*)}{q_T(\mathcal{T}^*|\mathcal{T})} J_q \tag{33}$$

(f) Set the new c_j^1 to this c_j^{1*} with probability $a(c_j^{1*}, c_j^1)$.

(g) If new c_j^1 is set to c_j^{1*},

- Set $\boldsymbol{G} = \boldsymbol{G}^*$, $\mathcal{T} = \mathcal{T}^*$, $\boldsymbol{c}^2 = \boldsymbol{c}^{2*}$

- If the new c_j^1 is not set to c_j^{1*}, it is the same as the old c_j^1. $\boldsymbol{G}$, $\boldsymbol{c}^2$ and $\mathcal{T}$ remain the same.

2. For $j = 1, \ldots, m$, update c_j^1 as follows:

   - If c_j^1 is a singleton, do nothing.
   - Otherwise, choose a new value for c_j^1 from $\{c_1^1, \ldots, c_m^1\}$ using the following probabilities:

$$P(c_j^1 = c | \boldsymbol{c}_{-j}^1, D[j], \boldsymbol{G}, Y_j, \alpha, \beta) \propto \frac{n_c}{m-1} E(D[j] | G_c, G_{c_j^2}, Y_j, \alpha, \beta)$$

Sample secondary cluster indicators:

For $j = 1, \ldots, m$, update c_j^2 as follows:

- If $Y_j = 0$, do nothing.
- Otherwise, choose a new value for c_j^2 from $\{1, \ldots, |\boldsymbol{c}^1|\}$ using the following probabilities:

$$P(c_j^2 = c | c_j^1, D[j], \boldsymbol{G}, Y_j, \alpha, \beta) \propto E(D[j] | G_{c_j^1}, G_c, Y_j, \alpha, \beta)$$

Sample clonal phylogeny and evolution model parameters:

Sample new clonal phylogeny $\mathcal{T}^*$ and new set of values for parameters of model of evolution, $\mathcal{M}_\lambda^*$ from the joint conditional posterior distribution, $\mathcal{P}_{\mathcal{T},\mathcal{M}_\lambda}(\mathcal{T}^*, \mathcal{M}_\lambda^* | \mathcal{T}, \mathcal{M}_\lambda, \boldsymbol{G}, a_M, b_M)$

$$\mathcal{T}^*, \mathcal{M}_\lambda^* \sim \mathcal{P}_{\mathcal{T},\mathcal{M}_\lambda}(\mathcal{T}^*, \mathcal{M}_\lambda^* | \mathcal{T}, \mathcal{M}_\lambda, \boldsymbol{G}, a_M, b_M)$$

Sample clonal genotypes:

For $k = 1, ..., |\boldsymbol{c}^1|$

- Sample clonal genotype G_k for each clone as follows:

  For $i = 1, ..., n$, sample G_{ki} from the following distribution

$$G_{ki} \propto \mathcal{F}(G_{ki}|T, \boldsymbol{G}_{-ki}, M) \times \prod_{j|c_j^1=k} E(D_{ij}|G_{ki}, G_{c_j^2 i}, Y_j, \alpha, \beta)$$

Sample error rates:

1. Sample $\alpha \sim \mathcal{P}_\alpha(\alpha|\boldsymbol{D}, \boldsymbol{c}^1, \boldsymbol{c}^2, \boldsymbol{Y}, \boldsymbol{G}, \beta, a_\alpha, b_\alpha) \sim E(\boldsymbol{D}|\boldsymbol{c}^1, \boldsymbol{c}^2, \boldsymbol{Y}, \boldsymbol{G}, \beta, \alpha)P(\alpha|a_\alpha, b_\alpha)$ using rejection sampling.

2. Sample $\beta \sim \mathcal{P}_\beta(\beta|\boldsymbol{D}, \boldsymbol{c}^1, \boldsymbol{c}^2, \boldsymbol{Y}, \boldsymbol{G}, \alpha, a_\beta, b_\beta) \sim E(\boldsymbol{D}|\boldsymbol{c}^1, \boldsymbol{c}^2, \boldsymbol{Y}, \boldsymbol{G}, \beta, \alpha)P(\beta|a_\beta, b_\beta)$ using rejection sampling.

3. Sample $\delta \sim \mathcal{P}_\delta(\delta|\boldsymbol{Y}, a_\delta, b_\delta)$

$$\delta \sim \mathcal{P}_\delta(\delta|\boldsymbol{Y}, a_\delta, b_\delta) \sim Beta(a_\delta + \sum_{j=1}^{m} Y_j, b_\delta + m - \sum_{j=1}^{m} Y_j)$$

Sample doublet indicators:

For $j = 1, \ldots, m$, sample Y_j based on the following distribution:

$$\begin{aligned} P(Y_j = 0|D[j], c_j^1, c_j^2, \boldsymbol{G}, \alpha, \beta, \delta) &\propto E(D[j]|G_{c_j^1}, \alpha, \beta)P(Y_j = 0|\delta) \\ P(Y_j = 1|D[j], c_j^1, c_j^2, \boldsymbol{G}, \alpha, \beta, \delta) &\propto E(D[j]|G_{c_j^1}, G_{c_j^2}, Y_j = 1, \alpha, \beta)P(Y_j = 1|\delta) \end{aligned} \quad (34)$$

In Eq. (34), $E(D[j]|G_{c_j^1}, \alpha, \beta)$ or $E(D[j]|G_{c_j^1}, G_{c_j^2}, Y_j = 1, \alpha, \beta)$ is calculated based on the likelihood function for cell j and $P(Y_j|\delta)$ is given by the prior distribution on Y_j, $Bernoulli(Y_j|\delta)$.

Sample concentration parameter:

Sample $\alpha_0^t \sim p(\alpha_0|m, |\boldsymbol{c}^1|, a, b)$ based on the method described in [5] assuming the prior distribution for α_0 is $Gamma(a, b)$.

2 Supplemental Results

2.1 Benchmarking on Simulated Datasets

Ground truth clonal structure and clonal phylogeny is not known for real tumor datasets. Consequently, simulation experiments become the approach of choice. To evaluate the performance of SiCloneFit, we performed comprehensive simulations. The simulation studies were aimed at analyzing the following:

1. How accurately SiCloneFit clusters the cells into different clones.
2. How accurately SiCloneFit infers the genotypes of the clones.
3. How accurately SiCloneFit infers the clonal phylogeny.

Here, we describe in detail the benchmarking of SiCloneFit along with other competitor methods on a wide variety of simulation experiments. The remainder of this section is organized as follows. Section 2.1.1 describes the simulation strategy for generating realistic ground truth data set for benchmarking purposes. In Section 2.1.2, the methods for summarizing the posterior samples of SiCloneFit are explained. In Section 2.1.3, we introduce the competitor methods against which we compared SiFit's performance. We describe the metrics used for comparing the different phylogeny inference methods in Section 2.1.4. Finally, we show and discuss the results of different experiments in Section 2.1.5.

2.1.1 Simulation of Synthetic Datasets

2.1.1.1 Simulation of Clonal Clusters

To simulate a number of clones and draw cells from the clones, we first fix the number of clones, K. For each clone k, we first sample observed prevalences $\mathbf{\Phi}^{obs} = \{\Phi_1^{obs}, \Phi_2^{obs}, \ldots, \Phi_K^{obs}\}$ from a Dirichlet distribution

$$\Phi_k^{obs} \sim Dir(\lambda, \mathbf{\Phi}^{true}) \qquad k = 1, 2, \ldots, K \tag{35}$$

where $\mathbf{\Phi}^{true} = \{\Phi_1^{true}, \Phi_2^{true}, \ldots, \Phi_K^{true}\}$ are the true prevalences for clones 1 to K sampled from a beta distribution. Let us assume, m is the number of cells that we want to simulate in

our experiment. The m cells are sampled from a multinomial distribution with parameters $\mathbf{\Phi}^{obs}$ as given by:

$$n_1, n_2, \ldots, n_K \sim Mult(\mathbf{\Phi}^{obs}) \tag{36}$$

where, n_k is the number of cells sampled from clone k and $\sum_{k=1}^{K} n_k = m$. The cells sampled from clone k have the true genotype which is same as the clonal genotype of clone k. This process of sampling cells is equivalent to sampling the cells from a Dirichlet-multinomial distribution, i.e., $n_1, n_2, \ldots, n_K \sim Dirichlet - multinomial(\lambda, \mathbf{\Phi}^{true})$. The simulation of clonal clusters follows the steps introduced in [25].

2.1.1.2 Simulation of Clonal Phylogeny

A clonal phylogeny is a binary leaf labeled phylogenetic tree where the leaves represent clones. [4] described different models of tumor evolution, linear and branching being the most notable one for point mutations. We construct linear and branching topologies for clonal phylogeny using the Beta-splitting model [24] parameterized by two parameters α_T and β_T. First, a generating sequence $(S_i)_{i\geq 1}$, a realization of a sequence of independent and identically distributed random variables is generated. To construct the generating sequence, a sequence of i.i.d random variables, $(b_1, b_2, \ldots)$ are sampled from the distribution $\mathcal{B}(\alpha_T + 1, \beta_T + 1)$, where $\mathcal{B}(\alpha_T, \beta_T)$ is a distribution on $[0, 1]$ with density $B(\alpha_T, \beta_T)^{-1} x^{\alpha_T - 1}(1 - x)^{\beta_T - 1}$. $B(\alpha_T, \beta_T)$ is defined by:

$$B(\alpha_T, \beta_T) = \int_0^1 x^{\alpha_T - 1}(1 - x)^{\beta_T - 1} dx \tag{37}$$

Another sequence of i.i.d random variables $(u_1, u_2, \ldots)$ are sampled from the uniform distribution on $[0, 1]$. The generating sequence is defined as $(S_i = (u_i, b_i))_{i\in\mathbb{N}}$. Once, the generating sequence is fixed, a nonrandom organizing process helps to create ranked planar binary tree with the desired number of leaves (clones). The organizing process incrementally creates a tree with K leaves (for a clonal phylogeny with K clones) starting from a single root node, labelled by the interval $[0, 1]$ as follows:

- Step 1: The root is split into a left leaf labelled by $[0, b_1]$ and a right leaf labelled by $[b_1, 1]$.

- Step 2: If $u_2 \in [0, b_1]$, the left child node of the root is further split into two nodes, the left one is labeled by $[0, b_1 b_2]$ and the right one is labeled by $[b_1 b_2, 1]$. If $u_2 \in [b_1, 1]$, the right child node of the root is split into left and right leaves with respective labels $[b_1, b_1 + (1 - b_1)b_2]$ and $[b_1 + (1 - b_1)b_2, 1]$.

- Step i: The leaf whose internal label $[a, b]$ contains u_i is chosen. It is split into a left leaf with label $[a, a + (b - a)b_i]$ and a right leaf with label $[a + (b - a)b_i, b]$.

- The process is stopped at the end of Step $K - 1$.

To generate a linear tree topology, values of α_T and β_T are chosen very close to -1. We choose $\alpha_T = -0.9999999999999999$ and $\beta_T = -0.9999999999999999$ for generating linear, comb like tree.

For generating a branching tree topology, we set $\alpha_T = 10000000$ and $\beta_T = 10000000$. After choosing a topology, the branch lengths are sampled from the prior distribution on branch length.

2.1.1.3 Simulation of Clonal Genotypes

To generate the genotype of each clone at the leaves of the clonal phylogeny, we first specify the number of mutation sites, n that we want to simulate. The root node of the phylogeny is populated with homozygous reference genotype ($g = 0$) at each site. In each branch of the tree, a Poisson distributed number of sites, p, are mutated. If t is the branch length, the parameter for the Poisson distribution is chosen as $t \times n$, so that on an average, a child node in the tree differs from its parent by a proportion of loci which is given by the branch length. When mutating a new site, the genotype changes from homozygous reference ($g = 0$) to heterozygous ($g = 1$). Recurrent mutations are introduced with probability r. If the locus in the node, for which a recurrent mutation happens, has a homozygous reference genotype ($g = 0$), then a parallel mutation happens in that branch, i.e, the genotype changes from homozygous reference ($g = 0$) to heterozygous ($g = 1$). If the locus in the node already contains a mutated genotype then a back mutation results in reverting the genotype to homozygous reference ($g = 0$). To simulate loss of heterozygosity (LOH) events, the loci with heterozygous ($g = 1$) genotypes are set to either homozygous refer-

ence ($g = 0$) or homozygous non-reference ($g = 2$) genotypes with probability ω. If LOH happens at a locus, either of the homozygous genotypes are chosen with equal probability. Deletion is simulated with probability d at a branch. Deletion can affect multiple loci at a time. For a heterozygous site, deletion can happen for any of the copies resulting in either of the homozygous genotypes ($g = 0$ or $g = 2$). Deletion does not affect the homozygous reference genotypes but can change the homozygous non-reference genotypes to heterozygous genotype. In this way, sites are evolved at each branch of the tree. At the corner case, when there is no new locus to mutate at a branch, recurrent mutations are introduced. After considering all the branches of the tree, we have the clonal genotypes at the leaves of the clonal phylogeny. The simulation of recurrent mutations, deletions and LOH are performed in the same way as introduced in SiFit [29].

2.1.1.4 Simulation of Noisy Single-cell Genotypes

The true genotype of a cell is same as the clonal genotype of the clone from which the cell was sampled. To obtain the noisy genotype for each cell, we introduce doublets, false positive and false negative errors and missing values.

2.1.1.4.1 Simulating Doublets

Doublets are events when two cells get trapped in the same well resulting in merging the genotypes of the two cells. The expected genotype of doublets can be constructed using the $\oplus$ operator defined in Section 4. In simulating doublets, we use similar strategy as used previously in [29]. δ denotes the fraction of cells that are doublets. With probability δ, a cell is chosen to be a doublet. The co-trapped cell with which the candidate cell merges to form a doublet can originate from any of the existing clones. We uniformly randomly choose the parent clone for the co-trapped cell and its genotype is combined with that of the candidate cell to form the new genotype of the doublet.

2.1.1.4.2 Simulating False Negative and False Positive Errors

False negative (FN) and false positive (FP) errors are introduced in the single-cell genotypes. For the datasets without doublets, FN and FP are introduced to true genotypes of

single cells. For the datasets with doublets, FN and FP are introduced to singlets as well as to doublets formed after simulation of doublet genotypes. FP and FN are introduced in the same way as described in [29].

2.1.1.4.3 Simulating Missing Data

To introduce missing values in the datasets, uniformly randomly genotype information of sites are removed with probability equal to the fraction of missing values that we want to introduce.

2.1.2 Summarizing Posterior Samples from SiCloneFit

To summarize the clustering samples from the Gibbs sampler of SiCloneFit, we utilized the maximum posterior expected adjusted rand (MPEAR) method introduced in [7]. In our case, the number of clusters can vary from one sample to another and the labels associated with the clusters can also change. As a result, we used a method based on posterior similarity matrix. The MPEAR method first computes a posterior similarity matrix, an $m \times m$ matrix (for m cells), in which each entry contains the posterior probability of two cells belonging to the same clonal clusters. Given the posterior similarity matrix, the posterior expected adjusted rand (PEAR) index can be utilized as a metric for assessing the performance of a proposed clustering configuration. We reported the clustering configuration that achieves the highest PEAR index as the summary cluster configuration. For the singlet model of SiCloneFit, the cluster samples, $\boldsymbol{c}$ were used for computing MPEAR clustering estimate. For the doublet model of SiCloneFit, we used the primary cluster indicator vector, $\boldsymbol{c}^1$ for computing MPEAR clustering estimate. The R package *mcclust* was used for computing MPEAR clustering summary.

To summarize the clonal phylogeny samples from the Gibbs sampler of SiCloneFit, we constructed a maximum clade credibility topology (MCCT) from the posterior samples. In this method, each sampled phylogeny is evaluated and each clade is given a score based on the posterior probability of appearing in the set of sampled phylogenies, and the product of the clade posterior probabilities is chosen as the score of a phylogeny. The phylogeny with the highest score is reported as the maximum clade credibility topology. In this process,

the branch lengths are also summarized over the posterior samples. We used the *SumTrees* program of the *DendroPy* [27] package to compute the MCCT.

From the posterior samples, we computed the posterior probability of the genotype of each cell at each site. The posterior probability of genotype g for cell j at position i is given by

$$P(I_{ij} = g|\mathcal{S}) = \frac{1}{N_S} \sum_{s=1}^{N_S} \mathbb{I}_{\{G^s_{c_j i}=g\}}, \tag{38}$$

where $\mathcal{S}$ denotes the set of N_S posterior samples. The genotype with the highest posterior probability is assigned as the inferred genotype, I_{ij} of that cell at that position.

The doublets are inferred when using the doublet-aware model of SiCloneFit based on the posterior probability computed from posterior samples as shown in Eq. (39)

$$P(Y_j = 1|\mathcal{S}) = \frac{1}{N_S} \sum_{s=1}^{N_S} \mathbb{I}_{\{Y_j=1\}} \tag{39}$$

Since, we consider a very low prior probability for a cell being a doublet, if the doublet posterior probability for a cell exceeds 0.05, we infer it as a doublet.

2.1.3 Competitor Methods

We compared SiCloneFit's performance to four other methods.

1. SCG (Single Cell Genotyper) [23]
2. OncoNEM [22]
3. SCITE [12]
4. SiFit [29]

OncoNEM, SCITE and SiFit were developed for the inference of tumor phylogeny from SCS data, whereas SCG was developed for the inference of clones from SCS data. From now on, we will use the term 'phylogeny-based methods' to refer to OncoNEM, SCITE and SiFit together.

2.1.3.1 SCG

Single Cell Genotyper (SCG) [23] is a statistical method that infers clonal genotypes and clonal structures from single cell somatic SNV profiles. However, it does not infer the clonal phylogeny and their inference procedure does not account for the phylogenetic structure underlying the clonal populations. We used SCG to infer the clones and clonal genotypes from the single-cell SNV profiles. The clonal phylogeny was obtained by running a maximum parsimony algorithm [26] on the clonal genotypes as suggested in [23].

2.1.3.2 OncoNEM

OncoNEM is a likelihood-based method that employs a heuristic search algorithm to find the maximum likelihood clonal tree. Nodes of the clonal tree represent the clonal clusters and the branches denote the evolutionary relationship between the clones. It is also possible to obtain the clonal genotypes by inferring the occurrence of the mutation on the branches of the clonal tree. OncoNEM's inference is also based on the "infinite sites assumption" and it does not account for the presence of doublets. We compared against OncoNEM only for the datasets without doublets. OncoNEM ran properly on small sized datasets ($m = 100$) but we were unable to get any result on larger datasets ($m = 500$). Comparison against OncoNEM are only shown for small sized datasets ($m = 100$).

2.1.3.3 SCITE

SCITE is an MCMC algorithm that allows one to infer the maximum likelihood mutation tree from imperfect somatic mutation profiles of single cells. The nodes of the mutation tree represent the mutations and the branches denote the order of the mutations in the evolutionary history. In the mutation tree, the sequenced cells can be attached to the nodes that correspond to their mutation states. Just like OncoNEM, SCITE also relies on the "infinite sites assumption" so that the mutation tree represents a perfect phylogeny and does not account for the presence of doublets. SCITE's results were compared only for the datasets without doublets. The genotypes for each cell can be inferred from the mutation tree and cell attachment inferred by SCITE. However, the cells were not clustered into clones. To obtain the clusters, first we computed an $m \times m$ distance matrix for the cells

based on their distances in the mutation tree. The distance between two cells was calculated by summing the number of mutation nodes on the shortest path that connects the two cells (essentially the hamming distance between the inferred genotypes of two cells). The resulting distance matrix was subject to K-medoids clustering (using 'clustering' library of R, `http://www.r-project.org`) for a varying number of clusters (2 to 20). The number of clusters and the clustering assignment that maximized the average silhouette score was inferred as the optimal clustering. To obtain the clonal tree, the cells that belong to a single cluster were attached to a node that was formed by collapsing the mutation nodes representing the parents of the corresponding cells in the mutation tree. The collapsed node's position in the tree was chosen so that its distance from the root (normal) node is minimized. The mutation nodes that did not have any cell attachment and had only one mutation node as the children were removed.

2.1.3.4 SiFit

SiFit is a likelihood-based algorithm that infers a tumor lineage tree under a finite-site model of evolution. It infers a tumor phylogeny, leaves of which represent the single cells and in doing so it also accounts for possible mutation recurrence and losses along the branches of the phylogeny. After reconstructing a maximum likelihood phylogeny, it also infers the mutations on the branches of the phylogeny using a maximum likelihood approach. Just like the other phylogeny-based methods (OncoNEM and SCITE), it does not account for the presence of doublets. We compared SiFit's results only for the datasets without doublets. SiFit's mutation placement algorithm infers the genotype of each cell for constructing the inferred genotype matrix. SiFit infers a full binary tree on a leafset of size equal to the number of cells. To infer the clonal clusters from this tree, the branch lengths were set to the number of mutations inferred on the branch. Then, an $m \times m$ distance matrix was computed for the cells, where each entry represents the distance between two cells in the tree. The distance between two cells was computed by summing the branch lengths on the shortest path that connects the two cells. K-medoids clustering was performed on the distance matrix using 'clustering' library of R (`http://www.r-project.org`), the number of clusters was varied from 2 to 20. The number of clusters and the cluster-

ing assignment that maximized the average silhouette score was inferred as the optimal clustering. To obtain the clonal tree, the branches in the subtree that contained the cells of a cluster were collapsed by setting the branch length to 0 and the branches connecting subtrees representing different clusters were set to 1.

2.1.4 Performance Metrics

When comparing the various methods, we wanted to quantify three different aspects of their performance

1. How accurately the method clusters the cells into different clones.
2. How accurately the method infers the genotypes of each clone.
3. How accurately the method reconstructs the clonal phylogeny.

To measure each of these aspects, we introduced three different performance metrics as described below.

2.1.4.1 Accuracy of Clustering

2.1.4.1.1 Adjusted Rand Index

For the datasets without doublets, we used the adjusted rand index [11] to assess clustering accuracy. The rand index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. The raw rand index score is then "adjusted for chance" into the adjusted rand index score. The adjusted rand index is thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical. For SiCloneFit, the MPEAR clustering estimate based on the posterior samples was used as the predicted clustering.

2.1.4.1.2 B-Cubed F-score

When considering doublets, the problem becomes more difficult as cells may belong to multiple clusters. This changes the problem from a strict clustering problem, to a restricted feature allocation problem [2]. We used the B-Cubed F-score, extended to handle feature allocations, for comparing the performance of the algorithms in the presence of doublets [1]. Both SCG and SiCloneFit can detect doublets. The cells detected as doublets were removed and the clustering of the rest was considered for measuring the B-cubed metric. Again, for SiCloneFit, the MPEAR clustering estimate was used as the predicted clustering.

2.1.4.2 Accuracy in Inferring Clonal Genotypes

In the absence of doublets, we measured the hamming distance between the predicted genotype of the clone where a cell is assigned and the true genotype of the cell. We computed the sum of hamming distances for all the cells and normalized it to summarize a method's genotyping performance. The genotyping error (g_e) is defined by,

$$g_e = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{I}(G_{T_{ij}} \neq G_{I_{ij}})}{n \times m} \tag{40}$$

where G_T is the true genotype matrix, G_I is the inferred genotype matrix and $\mathbb{I}$ is the indicator function. The genotyping error represents the number of incorrectly predicted genotypes per each cell per each genomic site. We distinguished the methods which predict only the binary genotype, that is the presence or absence of the B allele, from those, which attempt to predict the three state genotype A, AB, B. The predictions for any method which predicts the three state genotype can be converted to a binary representation by mapping the AB, B states to the B allele present state.

When considering datasets with doublets, we removed the cells that were inferred as doublets by the method and considered the rest of the cells for measuring the genotyping error as given by,

$$g_e = \frac{\sum_{i=1}^{n} \sum_{j \in \mathcal{S}_s} \mathbb{I}(G_{T_{ij}} \neq G_{I_{ij}})}{n \times m} \tag{41}$$

where $\mathcal{S}_s$ is the set of singlets inferred by the method. For SiCloneFit, we used the inferred genotypes based on posterior probability to compute the genotyping error.

2.1.4.3 Accuracy in Inferring Clonal Phylogeny

To measure the accuracy of the inferred clonal phylogeny, we used pairwise cell shortest-path distance introduced in [22] as the tree reconstruction error. The pairwise cell shortest-path distance is computed between the true and inferred clonal phylogenies. In our case, both the true tree $\mathcal{T}_T$ and the inferred tree $\mathcal{T}_I$ are built on the same set of m cells but potentially can differ in the number of internal nodes. The internal nodes that are direct parents of the leaves (cells) represent the clonal clusters, each leaf is connected to its parent by a branch of length 0. For every pair of cells i and j, we computed the shortest-path $d_{ij}(.)$ between the two cells in each tree. If the two cells belong to the same clone, their shortest-path distance is 0, otherwise the shortest-path distance equals the number of edges (regardless of direction) that separate the clones of the two cells. Finally, we summed up the absolute differences between the shortest-path distances of all unordered pairs of cells in the two trees to obtain the overall pairwise cell shortest-path distance:

$$d(\mathcal{T}_T, \mathcal{T}_I) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} |d_{ij}(\mathcal{T}_T) - d_{ij}(\mathcal{T}_I)| \tag{42}$$

For datasets without doublets, all cells were considered for measuring the above tree reconstruction error. For datasets that had doublets, we only considered the cells that were inferred as singlets by the method. For SiCloneFit, we used the MCCT phylogeny as the inferred clonal tree and computed its pairwise cell shortest-path distance.

2.1.5 Results and Discussion

2.1.5.1 Testing the Finite-site Model

SiCloneFit assumes a finite-site model of evolution that accounts for the effects of mutation loss and recurrence along the branches of the clonal phylogeny. To analyze how well this model captures the effects of such losses and recurrences, we simulated single-cell datasets with varying rates of mutation loss and recurrence. In our simulation, we used three different parameters for introducing loss of heterozygosity (LOH), deletion and recurrent mutations respectively. Corresponding to these three parameters, we performed three different sets of experiments

- The first set of experiment analyzes SiCloneFit's performance in different regimes of deletion probability (d). Deletion can result in mutation losses and in our simulations, deletion events can affect multiple loci at a time essentially violating the site independence assumption used for inference. These datasets only featured mutation losses, no parallel mutations were introduced.

- The second set of experiment analyzes SiCloneFit's performance in different regimes of the probability of LOH (w). LOH can also result in mutation losses affecting each locus independently as used in [23]. These datasets only featured mutation losses, no parallel mutations were introduced.

- The third set of experiment analyzes SiCloneFit's performance in different regimes of the probability of recurrent mutation (r). This parameter introduces parallel mutations in the datasets. These datasets did not contain any mutation loss due to deletion or LOH. An extreme setting of this parameter ($r = 0$) generated datasets under infinite-sites model as no mutation loss or recurrence were introduced in the datasets.

For these experiments, SiCloneFit's performance was compared against that of SiFit that also employs a finite-site model to account for mutation losses and recurrence. We wanted to test whether SiCloneFit's ability to cluster the cells into clones gives it an edge over SiFit in recovering the clonal genotypes for varying amount of mutation losses and recurrences.

2.1.5.1.1 Performance on Datasets with Varying Deletion Probability

We first simulated a clonal phylogeny with number of clones (leaves), $K = 10$. The number of cells, m was set to $m = 100$ and the number of sites was set to $n = 100$. At the root of the clonal tree, each site has homozygous reference genotype. The sequences were evolved along the branches of the tree starting from the root. In each branch of the tree, new mutations and mutation losses were simulated. No recurrent point mutations were introduced. For introducing mutation losses, the probability of deletion (d) was varied from 0.05 to 0.2 in steps of 0.05 i.e, $d \in \{0.05, 0.1, 0.15, 0.2\}$. Such deletion events can potentially alter the genotypes of multiple sites at a time. The range of d is chosen such that the expected number of deletion events during the evolutionary history of the tumor

remain reasonable. The probability of LOH was fixed at $w = 0.1$ to introduce mutation losses that independently affected some sites. This process gave us the clonal genotypes at the leaves of this clonal phylogeny. True genotype matrix corresponding to m single cells was constructed by sampling the clonal genotype of each cell. Errors were introduced into the true genotype matrix to simulate single-cell errors. The false negative rate for cell c, β_c, was sampled from a normal distribution with mean $\beta_{mean} = 0.2$ and standard deviation $\beta_{sd} = \frac{\beta_{mean}}{10}$. False negatives were introduced in the genotype matrix with probability β_c for cell c. We introduced false positives to the genotype matrix with error rate, $\alpha = 0.05$, by converting homozygous reference genotypes to heterozygous genotypes with probability α.

SiCloneFit's results were compared against that of SiFit. The clustering accuracy of each method is shown in Supplemental Fig. S3a. For each experimental setting, SiCloneFit achieved better clustering accuracy than SiFit. SiCloneFit maintained a high clustering accuracy (> 0.95 for $d \in \{0.05, 0.1, 0.15\}$ and > 0.9 for $d = 0.2$) for each value of the deletion probability. For $d = 0.2$, SiCloneFit's clustering accuracy degraded slightly as the introduction of more deletion events also incorporates more violations of the site independence assumption. SiCloneFit performed much better than SiFit by achieving lower tree reconstruction error (Supplemental Fig. S3b) and genotype error (Supplemental Fig. S3c) for all values of d. SiCloneFit achieved $2-5$ times reduction in genotyping error compared to SiFit. It shows that SiCloneFit's ability to cluster the cells into clones combined with the finite-site model makes it more powerful than SiFit in recovering the clonal genotypes of the single cells.

2.1.5.1.2 Performance on Datasets with Varying Probability of LOH

In the second set of experiments, we first simulated a clonal phylogeny on $K = 10$ clones. The number of cells, m was set to $m = 100$ and the number of sites was set to $n = 100$. In each branch of the tree, new mutations and mutation losses were simulated. For introducing mutation losses, this time we varied the probability of LOH (w) from 0.05 to 0.2 in steps of 0.05 i.e, $w \in \{0.05, 0.1, 0.15, 0.2\}$. LOH events independently affect the genomic sites. The range of w was chosen such that only a small fraction of mutated sites

suffer from mutation loss. The deletion probability (d) was fixed at $d = 0.1$ so that a small number of sites simultaneously get affected by deletion to allow for a deviation of site independence assumption. This process gave us the clonal genotypes at the leaves of the clonal phylogeny. True genotype matrix corresponding to m single cells was constructed by sampling the clonal genotype of each cell. Errors were introduced into the true genotype matrix to simulate single-cell errors in the same way as done for the first set of experiments.

SiCloneFit's results were compared against that of SiFit. The clustering accuracy of each method is shown in Supplemental Fig. S4a. For each value of LOH probability, w, SiCloneFit achieved better or similar clustering accuracy compared to SiFit. SiCloneFit also achieved lower tree construction error (Supplemental Fig. S4b) and lower genotype error (Supplemental Fig. S4c) compared to SiFit for each experimental setting. SiFit's median tree reconstruction error was more than 4 times higher than that of SiCloneFit for $w = 0.1$ and $w = 0.2$. SiCloneFit's median genotype error was $3 - 17$ times lower than that of SiFit for different values of w.

2.1.5.1.3 Performance on Datasets with Varying Probability of Recurrent Mutation

For the third set of experiments, we varied the probability of recurrent mutation, r, while generating the datasets. The number of clones, K was set to $K = 10$, the number of cells, m was set to $m = 100$ and the number of sites was set to $n = 100$. In each branch of the tree, new mutations and recurrent point mutations were simulated. Deletion probability, d and probability of LOH, w were set to 0, so that no mutation loss occurs. The probability of recurrent mutation was varied from 0.0 to 0.2 in steps of 0.05 i.e, $r \in \{0.0, 0.05, 0.1, 0.15, 0.2\}$. The setting corresponding to $r = 0.0$ generated datasets under the infinite-sites model as no mutation loss or parallel mutation occurred during the evolutionary history of the tumor. After simulating the clonal genotypes, the erroneous genotype matrix for m single cells was constructed following the same procedure as described in previous experiments.

SiCloneFit's results were compared against that of SiFit. For the datasets that correspond to infinite-sites model ($r = 0$), both SiCloneFit and SiFit achieved high clustering

accuracy (Supplemental Fig. S5a). However, for such datasets, SiCloneFit achieved much lower tree reconstruction error (Supplemental Fig. S5b) and genotyping error (Supplemental Fig. S5c) compared to that of SiFit. SiCloneFit's clustering accuracy was comparable to that of SiFit for all values of r except for $r = 0.05$, when SiFit's clustering accuracy was better. For all values of r, SiCloneFit achieved lower tree reconstruction error compared to SiFit. Similarly, SiCloneFit's genotyping error was lower than that of SiFit for all values of r.

2.1.5.2 Performance on Datasets with Varying Number of Cells Without Doublets

To compare SiCloneFit's performance against other methods, we first simulated single-cell datasets with varying number of cells. These datasets did not have any doublet. For these experiments, we first simulated a clonal phylogeny with number of clones (leaves), $K = 10$. The number of cells, m, sampled from the K clones, was varied as $m = 100$ and $m = 500$. The number of sites, n, was varied as $n = 50$ and $n = 100$ respectively. These datasets well represent the experimental targeted sequencing datasets. At the root of the clonal tree, each site has homozygous reference genotype. The sequences were evolved along the branches of the tree starting from the root. In each branch of the tree, we simulated four types of events that can alter the genotype of a site: new mutation, deletion, loss of heterozygosity (LOH) and recurrent point mutation. This process gave us the clonal genotypes at the leaves of this clonal phylogeny. The true genotype for the cells sampled from clone k is identical to the clonal genotype of clone k. m genotype sequences corresponding to m single cells constituted the true genotype matrix. Errors were introduced into the true genotype matrix to simulate single-cell errors. The false negative rate for cell c, β_c, was sampled from a normal distribution with mean $\beta_{mean} = 0.2$ and standard deviation $\beta_{sd} = \frac{\beta_{mean}}{10}$. False negatives were introduced in the genotype matrix with probability β_c for cell c. We introduced false positives to the genotype matrix with error rate, $\alpha = 0.05$, by converting homozygous reference genotypes to heterozygous genotypes with probability α.

SiCloneFit's results were compared against SCG, OncoNEM, SiFit and SCITE. Same imperfect genotype matrix was used as input to SiCloneFit, SCG, SiFit and SCITE. For

OncoNEM, the genotype matrix was binarized by converting the heterozygous and homozygous non-reference genotypes to 1, i.e., presence of mutation.

2.1.5.2.1 Clustering Accuracy

The clustering accuracy of each method is shown in Supplemental Fig. S6. For all datasets, SiCloneFit's results were compared against that of SCG, SiFit and SCITE. OncoNEM's results could only be compared for smaller sized datasets ($m = 100$), as OncoNEM failed to run for larger sized datasets ($m = 500$). Performance for each algorithm improved as the value of n increased. In each experimental setting, SiCloneFit outperformed all other algorithms. Specifically for $m = 500$ and $n = 100$, it achieved perfect clustering for almost all the datasets. SCG performed better than the phylogeny based methods (OncoNEM, SiFit and SCITE) for most experimental settings. SiFit's performance was the best among the phylogeny-based methods. For $m = 100$ and $n = 100$, it's median clustering accuracy was even higher than that of SCG. OncoNEM had the lowest clustering accuracy for smaller sized datasets, SCITE had the lowest for larger number of cells.

2.1.5.2.2 Genotyping Accuracy

For these datasets, we further wanted to evaluate the genotyping accuracy of each of these methods. The genotyping performance was measured in terms of hamming distance per cell per site, lower the hamming distance, better the genotyping. The genotyping performance is shown in Supplemental Fig. S7. For all experimental settings, SiCloneFit achieved the lowest genotyping error, SCITE had the highest genotyping error. Performance of each algorithm improved as the number of mutation sites (n) increased. Again SCG performed better than the phylogeny-based methods but worse than SiCloneFit. SiFit's performance was the best among the phylogeny-based methods.

2.1.5.2.3 Clonal Phylogeny Inference Accuracy

Finally, we compared each of the methods for their accuracy in reconstructing the genealogy of the clones. SiCloneFit directly reports the clonal phylogeny, but SCG does not infer any phylogeny. For SCG, we inferred the phylogeny using maximum parsimony method

on the inferred clonal genotypes (method suggested in the original study [23]). OncoNEM also reports a clonal phylogeny. SiFit infers a cell lineage tree, which was converted to an equivalent clonal phylogeny after inference of clonal clusters via K-medoids clustering as described in Section 2.1.3. SCITE infers a mutation tree that was converted to an equivalent clonal phylogeny after inference of clonal clusters via K-medoids clustering as described in Section 2.1.3. The comparison is shown in Supplemental Fig. S8. SiCloneFit achieved major improvement in reducing the tree reconstruction error for each experimental setting, it outperformed all other methods. With an increase in the number of sites, performance of each of the algorithms improved. For smaller sized datasets, OncoNEM performed the worst. SiCloneFit was followed by SCG owing to its better clustering accuracy. For smaller sized datasets, SiFit and SCITE performed comparably (SCITE performing slightly better) but worse than SCG. For larger sized datasets, SCITE performed better than SiFit for $n = 50$, for $n = 100$, SiFit achieved lower tree reconstruction error than that of SCITE.

2.1.5.3 Performance on Datasets With Varying Number of Clonal Populations

Next, we evaluated the performance of the methods in the presence of higher numbers of clones. As the number of clones increases, the problem becomes more difficult. First, we simulated clonal phylogenies with varying number of clones, $K = 10$ and $K = 15$. For each of these settings, $m = 100$ cells were sampled from K clones. The number of sites was set to $n = 100$. After obtaining the true genotypes of each cell by evolving clonal genotypes along the phylogeny, we introduced FP and FN errors using the same error rates as in the previous experiment. For each setting of K, m and n, we generated 10 datasets.

SiCloneFit's results were compared against that of SCG, OncoNEM, SiFit and SCITE and are shown in Supplemental Fig. S9. For different values of clones, SiCloneFit outperformed all algorithms based on all three metrics. SCG performed better than the phylogeny-based methods when smaller number of clones ($K = 10$) were present in the datasets. However, for larger number of clones, SiFit performed better than SCG by achieving higher clustering accuracy and lower genotype error. Among the phylogeny-based methods, SiFit's performance was the best. OncoNEM performed the worst based on clustering accuracy and tree reconstruction error, whereas SCITE had the highest genotyping error in

each experimental setting. For each of the methods, performance degraded as the number of clones increased, which is expected as mere increase of the number of clones without an increase in number of cells makes the problem more difficult. However, compared to SCG, SiCloneFit's performance was more robust against the increase in number of clones. SiCloneFit's mean clustering accuracy reduced by 3.9% when number of clones increased from 10 (mean ARI = 0.978) to 15 (mean ARI = 0.939), whereas there was a 10% reduction in SCG's mean clustering accuracy when number of clones increased from 10 (mean ARI = 0.933) to 15 (mean ARI = 0.84).

2.1.5.4 Performance on Datasets with Increasing Error Rates

The single-cell sequencing datasets show a range of variation in the error rates. As a consequence, we tested SiCloneFit's performance on datasets for which the error rates were higher.

2.1.5.4.1 Robustness to Increasing False Negative Rate

Allelic dropout is the major source of error in single-cell sequencing data resulting in false negatives [19]. To test the robustness of SiCloneFit to increase in false negative rate, β, we simulated datasets with increased false negative rate. The number of clones, K was set to 10, the number of cells, m was set to 100 and the number of sites, n, was set to 100. Mean false negative rate, β_{mean}, was varied from 0.2 to 0.4 in steps of 0.1 i.e, $\beta_{mean} \in \{0.2, 0.3, 0.4\}$. The false negative rate of cell c, β_c was sampled from a normal distribution as described in the previous experiments. The false positive rate was set to $\alpha = 0.05$. With these settings, for each value of $\beta_{mean} \in \{0.2, 0.3, 0.4\}$, 10 datasets were simulated.

Performance of SiCloneFit was compared against that of SCG, OncoNEM, SiFit and SCITE as shown in Supplemental Fig. S10. SiCloneFit achieved the best clustering accuracy for all values of false negative rate. SCG and SiFit achieved similar mean clustering accuracy (lower than that of SiCloneFit) for each experimental setting. SiFit had the best clustering accuracy among the phylogeny-based methods. For $\beta_{mean} \in \{0.2, 0.3\}$, OncoNEM performed the worst in terms of clustering accuracy, whereas for $\beta_{mean} = 0.4$, SCITE had the lowest clustering accuarcy. With an increase in false negative rate, the clus-

tering accuracy of each method degraded. However, SiCloneFit's clustering accuracy was robust to increase in FN rate.

In reconstructing the clonal phylogeny, SiCloneFit performed the best in all settings. OncoNEM had the highest tree reconstruction error for all settings. SiFit performed the best among the phylogeny-based methods, but it performed worse than SCG. SCG was the second best method after SiCloneFit. SiCloneFit's tree reconstruction performance was robust to increase in false negative rate, its tree reconstruction error only increased slightly when FN rate increased to $\beta = 0.4$.

SiCloneFit's genotyping performance was the best in all experimental settings, whereas SCITE had the highest genotyping error. SiFit performed better than OncoNEM and SCITE but worse than SCG. Genotyping error of each method increased with an ncrease in the false negative rate. However, SiCloneFit's performance was robust, there was only a slight increase in genotyping error of SiCloneFit when FN rate increased from $\beta = 0.2$ to $\beta = 0.3$ and then $\beta = 0.4$. SiCloneFit's superior performance based on all metrics over the other methods for all values of false negative rate shows that SiCloneFit is more robust against an increase in false negative rate.

2.1.5.4.2 Robustness to Increasing False Positive Rate

We performed another set of experiments to test how robust are the methods against increasing FP rate. The number of clones, K was set to 10, the number of cells, m was set to 100 and the number of sites, n, was set to 100. False negative rate β_{mean}, was set to 0.2 and the false negative rate of cell c, β_c was sampled from a normal distribution as described in the previous experiments. We varied false positive rate from 0.05 to 0.1 in steps of 0.05, i.e, $\alpha \in \{0.05, 0.1\}$. With these settings, for each value of $\alpha \in \{0.05, 0.1\}$, 10 datasets were simulated. Performance of SiCloneFit was compared against that of SCG, OncoNEM, SCITE and SiFit as shown in Supplemental Fig. S11.

For different values of false positive rate, SiCloneFit performed the best by achieving the highest clustering accuracy, OncoNEM had the lowest clustering accuracy. SCG and SiFit achieved similar mean clustering accuracy for $\alpha = 0.05$, but for higher false positive rate, $\alpha = 0.1$, SiFit's performance was better than that of SCG. SCITE performed better

than OncoNEM but worse than the other methods.

SiCloneFit also performed the best by achieving the lowest tree reconstruction error for all values of FP rate. Based on this metric too, OncoNEM performed the worst. SCG's performance was better than the phylogeny-based methods but worse than SiCloneFit.

Based on genotyping error, SiCloneFit outperformed all other methods for all experimental settings. Among the phylogeny-based methods, SiFit's performance was the best. SCITE had the highest mean genotyping error. SCG's genotyping performance was affected by the increase in FP rate. For two datasets generated with $\alpha = 0.1$, SCG reported a large number of incorrect genotypes ($\sim 40 \times$ median value), which shows that SCG's genotypes failed to converge.

SiCloneFit's clustering accuracy did not get affected by the increase in FP rate. Same trend was observed for SiCloneFit's tree reconstruction error. SiCloneFit's genotyping performance did not suffer much by the increase in FP rate. This shows its robustness against an increase in FP rate.

Based on all our previous experiments, SCG was the best competitor method. In our subsequent experiments that required comparison, we only compared SiCloneFit's performance to that of SCG.

2.1.5.5 Performance on Datasets with Missing Data

Due to uneven coverage and amplification bias, current single-cell sequencing datasets are challenged by missing data points where genotype states are unobserved. To investigate how missing data affect the performance of each method, we performed additional simulation experiments. For $K = 10$, $m = 500$ and $n = \{50, 100\}$, we generated datasets using the same error rates as before. For each combination of K, n and m, we generated 10 datasets, for each of which, two other datasets with missing data = $\{15\%, 30\%\}$ were generated. SiCloneFit's results were compared against SCG.

2.1.5.5.1 Clustering Accuracy

The clustering accuracy of each method under different levels of missing data is shown in Supplemental Fig. S12. For each setting, each method performed better when more sites

($n = 100$) were present. This is expected given the fact that sequencing more sites result in more data and incorporates more information. As the amount of missing data increased, performance of each method degraded. SiCloneFit performed either better than SCG (no missing data, 15% missing data) or similar to SCG (30% missing data). Even for datasets with 30% missing data, where overall performance of SiCloneFit was similar to that of SCG, the mean accuracy of SiCloneFit was higher for all values of number of sites n. For datasets with $n = 100$ sites, each method performed well even after removal of significant amount of data. This shows that the clustering performance of both SiCloneFit and SCG are robust against increasing missing data when sufficient number of genomic sites are sequenced.

2.1.5.5.2 Genotyping Accuracy

The genotyping performance of each method under different levels of missing data is shown in Supplemental Fig. S13. Genotyping error increased with an increase in the amount of missing data. SiCloneFit outperformed SCG in all cases except for the setting $n = 100$ and 30% missing data. For both the methods, the genotyping error was higher for datasets with number of sites $n = 100$. This is expected because of the increase in the number of sites.

2.1.5.5.3 Clonal Phylogeny Inference Accuracy

The performance of each method in inferring clonal phylogeny under different levels of missing data is shown in Supplemental Fig. S14. Under each setting, SiCloneFit outperformed SCG. With an increase in the number of sites, the phylogeny inference improved for SiCloneFit. But for SCG, phylogeny inference did not improve much with the increase in number of sites. It degraded for datasets with 15% missing data. SiCloneFit's phylogeny inference was not affected much by the increase in the amount of missing data. This shows that SiCloneFit's phylogeny inference is robust to the presence of missing values.

2.1.5.6 Performance on Datasets Generated Under Neutral Evolution

Neutral evolution (NE) represents an extreme case of branching evolution and postulates

that intratumor heterogeneity (ITH) is caused by accumulation of random mutations that lack any functional significance or selection advantage in the progression of a tumor [4]. Tumors evolved under this model should consist of many subpopulations without any evidence of a single clone being selected and expanded. Even though in most human cancers there is evidence of at least weak selection, which leads to the prevalence of clonal subpopulations that harbor driver mutations [8], some tumors might undergo neutral evolution as shown in [15, 28]. To analyze SiCloneFit's performance under neutral evolution that can lead to an absence of clonal structure, we conducted simulation experiments under the neutral evolution model proposed in Williams *et al.* [28].

According to the NE model proposed in [28], under neutral evolution, the number of subclonal mutations should follow $\frac{1}{f}$ power law distribution (f being the allelic frequency of a mutation). The cumulative distribution, $M(f)$ of subclonal mutations should have a linear relationship with $\frac{1}{f}$ and the R^2 goodness-of-fit measure should be $R^2 \geq 0.98$ for neutral evolution. In our simulation, to ensure that the cumulative distribution of the subclonal mutations follow the $\frac{1}{f}$ power law, we sampled clonal prevalences from a normal distribution with narrow standard deviation (to obtain very similar clonal prevalence for each clone) and the branch lengths of the tumor phylogeny were chosen to be of the same order. The cumulative distribution of subclonal mutations for two representative datasets are shown in Supplemental Fig. S15.

We generated datasets consisting of 100 and 200 cells. Following the study of Ling *et al.* [15], which reported on potential evidence of neutral evolution in a hepatocellular carcinoma by identifying 20 clones, we fixed the number of clones, K to 20. $n = 100$ mutation sites were simulated for each dataset. For each combination of K, n and m, we generated 5 datasets. Same error rate values as discussed in the previous experiment were used. We compared SiCloneFit's results to that of SCG, SiFit and SCITE (Supplemental Fig. S16). As we see, for these datasets, SiCloneFit performed either similarly or better than the other methods based on the different metrics. For the smaller datasets (100 cells), SiFit had slightly better clustering and genotyping accuracy than SiCloneFit, but SiCloneFit's tree reconstruction error was lower. For the larger datasets (200 cells), SiCloneFit outperformed all other methods based on all metrics. These results show that SiCloneFit

performs well even under neutral model of evolution.

2.1.5.7 Estimation of Error Rates by SiCloneFit

The posterior samples obtained from SiCloneFit's Gibbs sampler can be used for inferring the false positive and false negative rate of the SCS dataset.

To assess SiCloneFit's estimation of false positive rate, we simulated 30 datasets from different 30 clonal phylogenies. For these datasets, the number of clones, K was set to 10, $m = 100$ cells were sampled from these clones and the number of sites, n was set to 100. The false negative rate, β was set to 0.2. The false positive rate, α was varied from 0.01 to 0.15 in steps of 0.005. SiCloneFit's inference algorithm was used to obtain posterior samples from the resulting noisy matrices. False positive rate was inferred by averaging the posterior samples. SiCloneFit performed very well for estimating false positive rate as shown in Supplemental Fig. S17a. The estimated values of α were highly correlated (0.998) to the original FP rates used for generating the datasets.

We performed another set of experiment to analyze SiCloneFit's performance in estimating the false negative rate. Just like the previous experiment, we simulated 30 datasets from different 30 clonal phylogenies, the number of clones, K was set to 10, $m = 100$ cells were sampled from these clones and the number of sites, n was set to 100. The false positive rate, α was set to 0.05. The false negative rate, β was varied from 0.1 to 0.4 in steps of 0.01. The resulting noisy matrices were given to SiCloneFit for inference. False negative rate was inferred by averaging the posterior samples. Again, SiCloneFit's estimated values of $beta$ were highly correlated (0.992) to the original FN rates used (Supplemental Fig. S17b).

These experiments show that SiCloneFit is able to precisely infer FP rate (α) and FN rate (β) from the SCS datasets.

2.1.5.8 Estimation of Number of Clusters by SiCloneFit

To analyze whether SiCloneFit accurately infers the number of clonal clusters, we simulated three sets of datasets with different levels of sampling distortion. For these datasets, the number of clones, K was set to 10, $m = 100$ cells were sampled from these clones and

the number of sites, n was set to 100. The FN rate, β was set to 0.2 and the FP rate was set to 0.05. For the three sets, λ, the concentration coefficient for the Dirichlet-multinomial distribution used for sampling the cells from the clones, was set to $\lambda = 10$, $\lambda = 100$ and $\lambda = 1000$ respectively. Single-cell datasets may show sampling bias due to random sampling of cells from the tissue. Larger the value of λ, the closer the Dirichlet-multinomial distribution approximates the true prevalences of the clones. At higher values of λ, the sampled cells better represent the true proportions of the clones, whereas, for smaller values of λ, the sampled cells deviate from the true prevalences of the clones. As a result, inference of the number of clusters becomes difficult when the value of λ is small. In all our experiments, we used a smaller value of $\lambda = 10$ to introduce sampling distortion that is likely in real SCS datasets.

The number of clusters estimated by SiCloneFit for different values of λ is shown in Supplemental Fig. S18. As the sampled single cells more closely followed the true prevalences (increasing value of λ) of the clones, SiCloneFit's estimate of the number of clusters got better. Even for datasets with fair amount of sampling bias ($\lambda = 10$), SiCloneFit was able to infer the actual number of clusters for some datasets. The clusters that were missed by SiCloneFit mostly consisted of 1 cell with a genotype very similar to another more populated clone. For larger λ, SiCloneFit was able to infer the actual number of clusters for most of the datasets.

2.1.5.9 Scalability of SiCloneFit for Large Datasets

To analyze SiCloneFit's applicability on datasets containing large number of cells, we simulated datasets with $m = 2000$ cells. For these datasets, the number of clones, K was set to 10, and the number of sites, n was set to 100. The FN rate, β was set to 0.2 and the FP rate was set to 0.05. We compared SiCloneFit's result on these datasets to that on the datasets containing $m = 500$ cells. The results are shown in Supplemental Fig. S19. SiCloneFit performed well for these large datasets. There was only a small drop in performance when number of cells increased from 500 to 2000.

We also simulated datasets with higher number of genomic sites to evaluate SiCloneFit's scalability with the number of sequenced mutation sites. We generated datasets with

$n = 400$ sites, the number of clones, K was set to 10 and the number of cells was set to $m = 100$. The FN rate, β was set to 0.2 and the FP rate was set to 0.05. SiCloneFit's result for these datasets are shown in Supplemental Fig. S20. SiCloneFit's performance improved for higher number of sites based on all metrics compared to datasets with $n = 100$ genomic sites.

These experiments show that SiCloneFit scales well with both the number of cells and number of genomic sites making it suitable for potentially larger future SCS datasets.

2.1.5.10 Performance on Datasets with Varying Number of Cells with Doublets

To assess the performance of SiCloneFit in the presence of doublets, we generated datasets with doublets. The doublet rate, δ, was set to 0.1 to introduce 10% doublets. We first simulated a clonal phylogeny with number of clones (leaves), $K = 10$. The number of cells, m, sampled from the K clones, was varied as $m = 100$ and $m = 500$. The number of sites, n, was varied as $n = 50$ and $n = 100$ respectively. The clonal genotypes were simulated by introducing point mutations, LOH, deletion and recurrent mutations along the branches of the phylogeny as discussed previously. The true genotype matrix consisted of m genotype sequences corresponding to m single cells, where the true genotype of cell j is identical to the clonal genotype of the clone where cell j belongs to. After that, doublets were formed by merging the genotypes of two single cells with probability δ. The false negative rate for cell c, β_c, was sampled from a normal distribution with mean $\beta_{mean} = 0.2$ and standard deviation $\beta_{sd} = \frac{\beta_{mean}}{10}$. False negatives were introduced in the genotype matrix with probability β_c for cell c. We introduced false positives to the genotype matrix with error rate, $\alpha = 0.05$, by converting homozygous reference genotypes to heterozygous genotypes with probability α.

SiCloneFit's performance was compared against that of SCG. For these datasets, the extended model of SiCloneFit that can handle doublets was used for inference. Similarly, for SCG, its doublet aware model was used for inference. Comparisons were done with respect to the different metrics as explained in Section 2.1.4.

2.1.5.10.1 Clustering Accuracy

The clustering accuracy of each method is compared in Supplemental Fig. S21. For smaller sized datasets ($m = 100$), SiCloneFit's clustering accuracy was much higher than that of SCG. Clustering accuracy of both methods improved with the increase in number of sites or the increase in number of cells. For larger sized datasets, SCG's clustering accuracy significantly improved but for all experimental settings, SiCloneFit performed better than SCG. For some datasets, SCG failed to converge resulting in very low clustering accuracy. In such cases, SCG mostly reported just a single cluster.

2.1.5.10.2 Genotyping Accuracy

The genotyping performance was measured by hamming distance excluding the inferred doublets, lower the hamming distance, better the genotyping. The genotyping performance is shown in Supplemental Fig. S22. For genotyping, SiCloneFit either outperformed SCG or performed similarly. The total genotyping error was lower for datasets with smaller number of sites. Again, SCG's failure to converge for some datasets was also visible in its genotyping performance. For such datasets, SCG's genotyping error was very high.

2.1.5.10.3 Clonal Phylogeny Inference Accuracy

Finally, we also compared the clonal phylogeny inference accuracy of each of these methods. SiCloneFit directly reports clonal phylogeny, whereas SCG does not infer any phylogeny. For SCG, we inferred phylogeny by running maximum parsimony method on inferred clonal genotypes (method suggested in the original study [23]). The comparison is shown in Supplemental Fig. S23. SiCloneFit performed better than SCG in all experimental settings except for $m = 500$, $n = 100$. Specifically, SiCloneFit's performance was substantially better for datasets with 100 cells. With an increase in number of cells, SCG's performance also improved. For some datasets, SCG's tree reconstruction error was very high because it did not converge and assigned all the cells in a single cluster resulting in a clonal phylogeny with a single node.

2.1.5.11 Performance on Datasets Containing Doublets with Varying Number of Clonal Populations

For the datasets with doublets, we next varied the number of clones. First, we simulated clonal phylogenies with varying number of clones $K = 10$ and $K = 15$. For each of these settings, $m = 100$ cells were sampled from K clones. The number of sites was set to $n = 100$. After obtaining the true genotypes of each cell by evolving clonal genotypes along the phylogeny, we introduced doublets with rate $\delta = 0.1$. Then we introduced FP and FN errors using the same error rates as described previously. For each setting of δ, K, m and n, we generated 10 datasets.

SiCloneFit's results were compared against SCG and shown in Supplemental Fig. S24. For different values of clones, SiCloneFit outperformed SCG in terms of all three metrics. For each of the methods, performance degraded as the number of clones increased, which is expected as mere increase of number of clones without increasing the number of cells makes the problem more difficult. However, SiCloneFit's performance was more robust against the increase in number of clones. For clustering accuracy, SiCloneFit's performance did not degrade much with the increase in number of clones, but for SCG, the clustering accuracy significantly reduced when the number of clones increased. The tree reconstruction error was much higher for the clonal phylognies inferred from SCG's clonal genotypes when number of clones increased. Similarly, genotyping error of SCG increased at a higher rate than that of SiCloneFit. This shows that SiCloneFit performed much better for more difficult inference problems.

2.1.5.12 Performance on Datasets Containing Doublets and Missing Data

To assess the performance of SiCloneFit in the presence of both doublets as well as missing values, we generated datasets for $K = 10$, $m = 500$, $n = \{50, 100\}$ and $\delta = 0.1$. FP and FN error rates were the same as used previously. For each combination of K, n, m, and δ we generated 10 datasets, for each of which, two other datasets with missing data = $\{15\%, 30\%\}$ were generated. SiCloneFit's results were compared against that of SCG.

2.1.5.12.1 Clustering Accuracy

The clustering accuracy of each method under different levels of missing data is shown in Supplemental Fig. S25. For each experimental setting, SiCloneFit's clustering accuracy was either better or similar compared to SCG. SiCloneFit's clustering accuracy did not suffer much by the presence of missing values. For some of such datasets, SCG failed to cluster them into separate groups and sometimes incorrectly inferred every cell as a doublet. For these datasets, SCG's clustering accuracy was very low because mostly all the cells were grouped in a single cluster.

2.1.5.12.2 Genotyping Accuracy

The genotyping performance of each method under different levels of missing data is shown in Supplemental Fig. S26. Genotyping error increased with the increase in amount of missing data. SiCloneFit outperformed SCG in all cases. In each experimental setting with missing data, for some datasets, SCG completely failed to converge and wrongly inferred every cell as a doublet. For these datasets, SCG's genotyping error was very high.

2.1.5.12.3 Clonal Phylogeny Inference Accuracy

The performance of each method in inferring clonal phylogeny under different levels of missing data is shown in Supplemental Fig. S27. SiCloneFit mostly performed better than SCG. However, SCG's performance was better for datasets with $m = 500$, $n = 100$ and with missing data = $\{0\%, 15\%\}$. Again for some datasets, SCG clustered every cell in a single group, as a result the inferred clonal phylogeny had just one node and the tree reconstruction error was very high. For most of the datasets, SiCloneFit correctly identified the doublets and removed them.

2.2 Inference of Clonal Clusters, Genotypes and Phylogeny from Experimental SCS Data

We applied SiCloneFit to two experimental single-cell DNA sequencing datasets from two metastatic colon cancer patients, obtained from the study of Leung *et al.* [13]. These datasets were generated using a highly-multiplexed single-cell DNA sequencing process

[14] and a 1000 cancer gene panel was used as the target region for sequencing. These are two of the most recent SCS datasets and contain large numbers of cells and small numbers of mutation sites making the inference difficult. The application of SiCloneFit on these datasets shows its broad applicability to modern SCS datasets.

2.2.1 Analysis of Patient CRC1

This dataset consisted of 178 cells [13] obtained from both primary colon tumor and liver metastasis. The original study reported 16 somatic SNVs after variant calling. The reported genotypes were binary values, representing the presence or absence of a mutation at the SNV sites.

After running SiCloneFit on this dataset, we collected the samples from the posterior and computed a maximum clade credibility tree based on the posterior samples.

Five different clusters were identified from the SiCloneFit posterior samples. The largest cluster (N) consisted of normal cells without any somatic mutation. The primary tumor cells were clustered into two subclones (P1 and P2). Metastatic aneuploid tumor cells were clustered into one subclone (M). There was another cluster (D) consisting of diploid cells (mostly metastatic). The clonal genotype of each cluster was inferred based on the posterior samples. The inferred genotypes are shown in Supplemental Fig. S28. Based on the clonal genotypes, we inferred the ancestral sequences at the internal nodes and this enabled us to find the maximum likelihood solution for placing the mutations on the branches of the clonal phylogeny. The inferred clonal phylogeny suggested that the mutation in *GATA1* occurred twice (in the diploid and metastatic subclones) indicating it as a potential convergent evolution. To evaluate the accuracy of this, we performed a mixture-model Bayesian binomial test as used in the original study [13]. This test utilized the reference and variant read counts of all the cells for this mutation to determine if it was present in the diploid (D) subclone as indicated by the clonal phylogeny. 4 (out of 5) cells in the diploid subclone (D) displayed high posterior probability (0.9661, 0.8181, 0.914, 0.9587 respectively) of harboring this mutation indicating a strong evidence for its recurrence.

For comparison, we ran SCG on this dataset. SCG reported 4 clonal clusters: a cluster

(SN) consisting of unmutated normal cells, a cluster (SD) consisting of 3 metastatic diploid cells, a cluster (SP) consisting of primary diploid and aneuploid cells and another cluster (SM) consisting of metastatic cells. The clonal genotypes of the cells inferred by SCG is shown in Supplemental Fig. S29. SCG could not distinguish the primary tumor cells on the basis of the presence/absence of the *TPM4* mutation and genotyped all of them to contain *TPM4*. Thus it did not report two primary tumor subclones that were detected by SiCloneFit and instead only one primary tumor subclone (all primary tumor cells were assigned to this cluster) is inferred. The distinction of primary tumor cells based on the presence and absence of *TPM4* mutation was also inferred by SCITE in the original study [13]. In the original study, SCITE tree reported that the *TPM4* mutation was gained in the primary tumor cells after the metastatic divergence (Fig. 6A in [13]). As a result, a number of primary tumor cells placed before the metastatic divergence did not harbor the *TPM4* mutation, it was only present in the primary tumor cells that were placed after the metastatic divergence. This further supports that SiCloneFit's inference of two primary tumor subclones is more plausible compared to SCG's inference of single primary tumor subclone. In addition, SCG being a clonal clustering method did not infer the phylogeny of the subclones.

2.2.2 Analysis of Patient CRC2

This dataset consisted of 182 cells [13] obtained from both primary colon tumor and liver metastasis. The original study reported 36 somatic SNVs after variant calling. The reported genotypes were binary values, representing the presence or absence of a mutation at the SNV sites.

After running SiCloneFit on this dataset, we collected the samples from the posterior and computed a maximum clade credibility tree based on the posterior samples. Six different clusters were identified in the MPEAR solution based on the posterior samples. The largest cluster (N) consisted of normal cells that did not harbor any somatic mutation. There were two clusters consisting of primary aneuploid tumor cells (P1 and P2) and two clusters consisting of metastatic aneuploid tumor cells (M1 and M2). There was one more cluster (I) comprised of diploid cells that had somatic mutations completely different from

the primary or metastatic clusters, representing an independant clonal lineage consistent with the findings reported by Leung *et al.* [13]. The clonal genotype of each cluster was inferred based on the posterior samples. The inferred genotypes are shown in Supplemental Fig. S30. Based on the clonal genotypes, we inferred the ancestral sequences at the internal nodes and this enabled us to find the maximum likelihood solution for placing the mutations on the branches of the clonal phylogeny. In the original study [13], SCITE tree identified two metastatic divergence events for this patient and also identified 4 mutations that occurred between the two metastatic divergence points. These were termed as 'bridge mutations' (*FHIT, APC, CHN1 and ATP7B*). These bridge mutations were identified to be present in primary tumor cells as well as cells in the second metastatic subclone but absent in the cells in the first metastatic subclone. SiCloneFit also identified two metastatic subclones but reported on two 'bridge mutations' (*FHIT and ATP7B*) that differed between the two metastatic subclones. These two mutations were reported to be present in the primary tumor subclone P2 and the metastatic tumor subclone M2, but were absent in metastatic tumor subclone M1. On the other hand, the other two mutations (*APC* and *CHN1*) were reported to occur before any metastatic divergence and subsequently were present in all three subclones (P2, M2 and M1). To verify this, we performed the mixture-model Bayesian binomial test proposed in [13] based on the read counts for these 4 mutations. The results are shown in Supplemental Fig. S31 and Supplemental Fig. S32. Supplemental Fig. S31 supports SiCloneFit's inference of *FHIT* and *ATP7B* to be the two 'bridge mutations'. Supplemental Fig. S32 shows that the mutations *CHN1* and *APC* had high posterior probability in a number of cells in all three subclones (P2, M2 and M1) indicating they potentially occurred before the first metastatic divergence and were present in all three subclones. This indicates that SiCloneFit's placement of these mutations in the tumor phylogeny is more plausible than that of SCITE.

Other than the precursor mutations shared with the primary tumor clones, the metastatic tumor clones had three more mutations in common (*PTPRD*, *FUS* and *LINGO2*). This is an evidence for a potential convergent evolution. To evaluate the accuracy of this, we performed the mixture-model Bayesian binomial test [13] with the reference and variant read counts for these three recurrent mutations to determine if they were present in both the

metastatic subclones. The resulting posterior probabilities and heat map (Supplemental Fig. S33) provided strong evidence that *LINGO2* and *FUS* were present in both the subclones. *PTPRD* had strong evidence of being present in the second metastatic subclone (M2) but weak evidence of occurring in the first metastatic subclone (M1). The posterior probability pattern of *PTPRD* also suggested that this mutation might have been affected by allelic dropout.

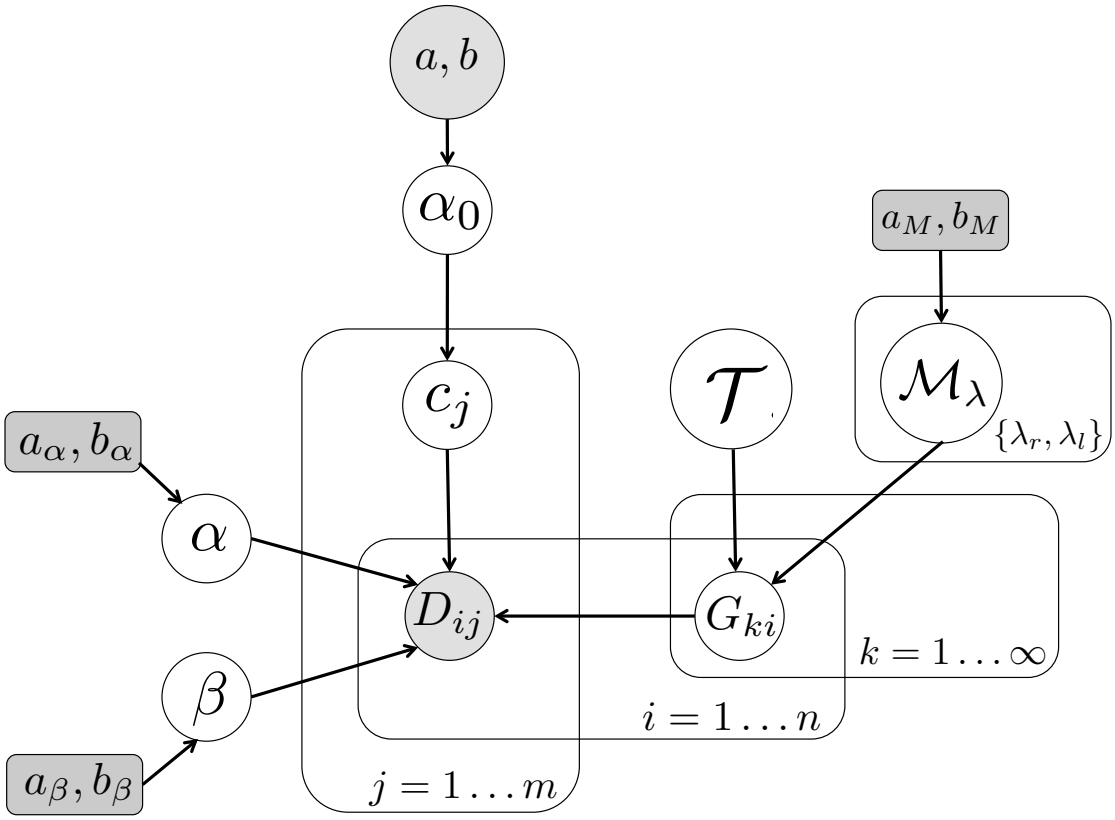
For comparison, we ran SCG on this dataset. SCG reported 5 clonal clusters: a cluster (SN) consisting of unmutated normal cells, a cluster (SP) consisting of primary aneuploid cells, two metastatic clusters (SM1 and SM2), and another cluster (SI) consisting of primary diploid cells. The clonal genotypes of the cells inferred by SCG is shown in Supplemental Fig. S34. Clustering and genotyping of SCG mostly agreed with that of SiCloneFit. However, SCG failed to detect two primary tumor subclones and instead clustered them together into one subclone (SP). As a result, the genotyping of the corresponding primary tumor cells were also incorrect and this can also affect the reconstruction of the mutational order.

2.3 Identification of Doublets from Experimental SCS Data

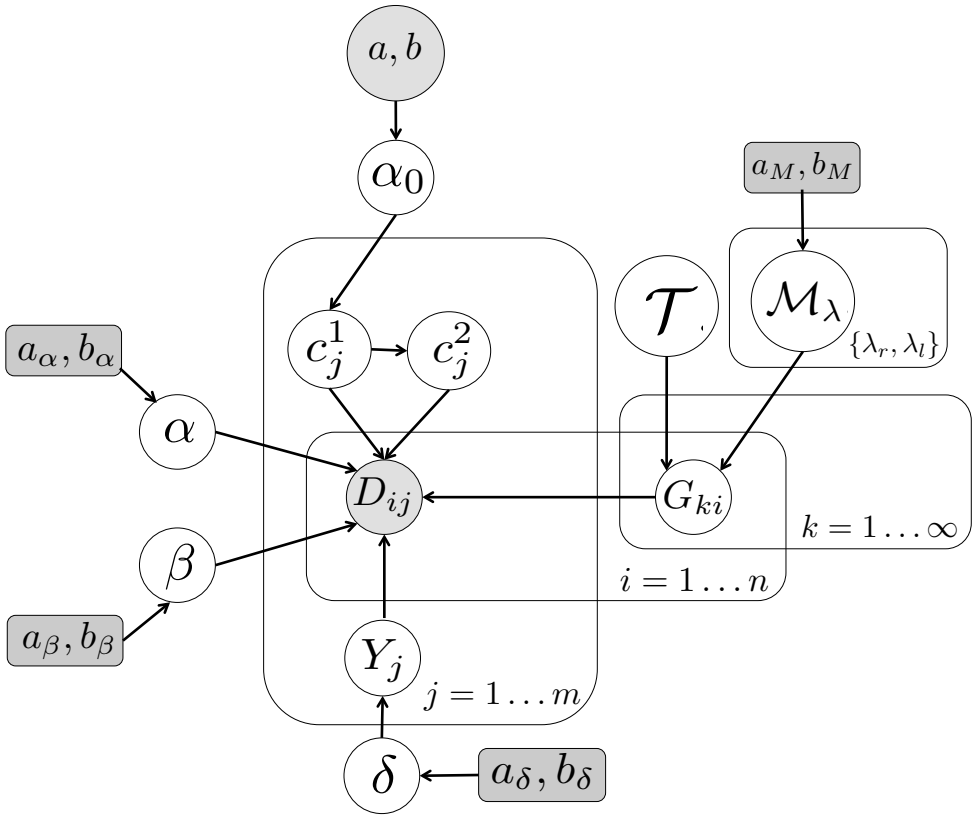
Neither SiCloneFit nor SCG detected any doublet from the above two colorectal cancer SCS datasets evidencing the absence of doublets in those datasets. In order to validate SiCloneFit's doublet detection from experimental SCS data, we applied SiCloneFit on a high grade serous ovarian cancer dataset introduced in McPherson *et al.* [17]. This dataset consisted of 370 cells and 43 somatic mutations were reported from these cells. SiCloneFit's doublet-aware model reported 20 doublets from this dataset. Since, ground truth doublets were not known for this dataset, we also ran SCG's doublet-aware model on this dataset. SCG reported 28 doublets for this dataset. Out of the 20 doublets identified by SiCloneFit, 17 were also reported by SCG. We further tested the 11 cells that were reported as doublets by SCG but not by SiCloneFit. 10 of them had similar posterior probabilities (computed by SCG) of being a doublet or a singlet. In other words, SiCloneFit inferred the most confident doublets inferred by SCG. The posterior probabilities of the inferred

doublets are shown in Supplemental Fig. S35. This experiment shows SiCloneFit's ability in detecting potential doublets from empirical single-cell datasets.

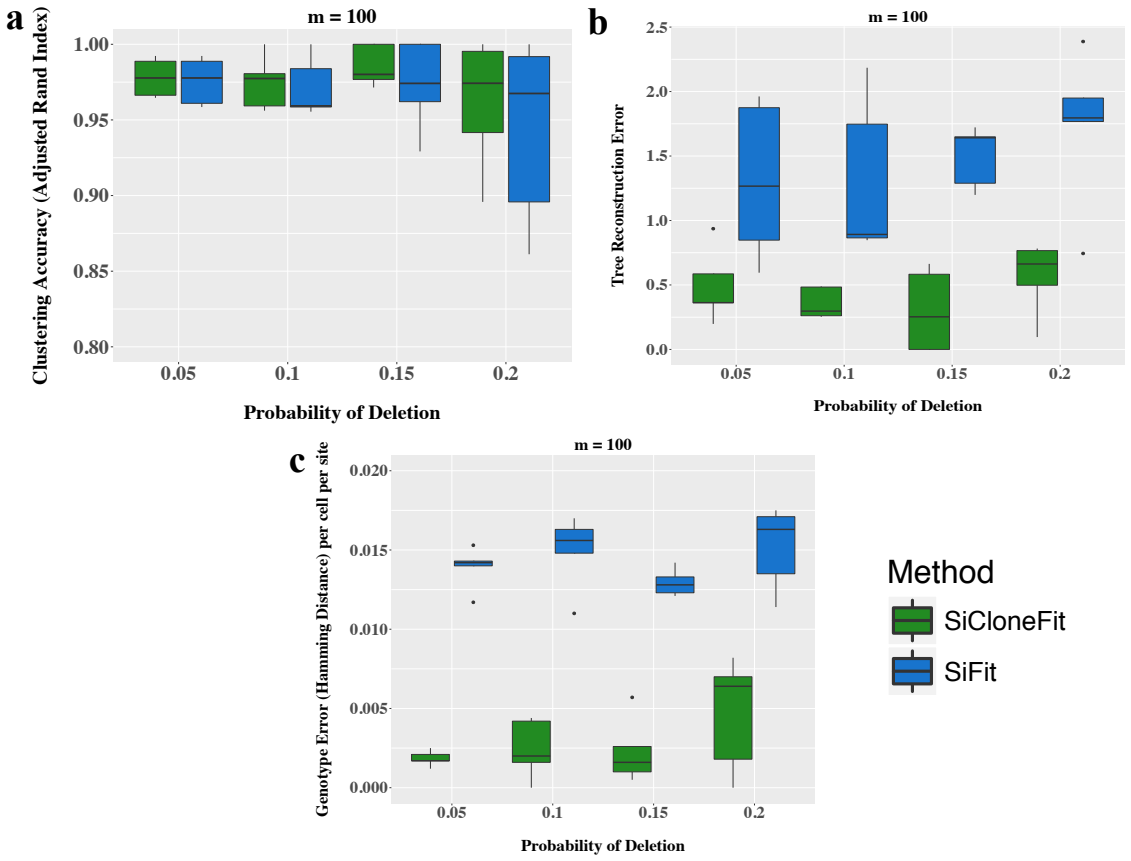
3 Supplemental Figures



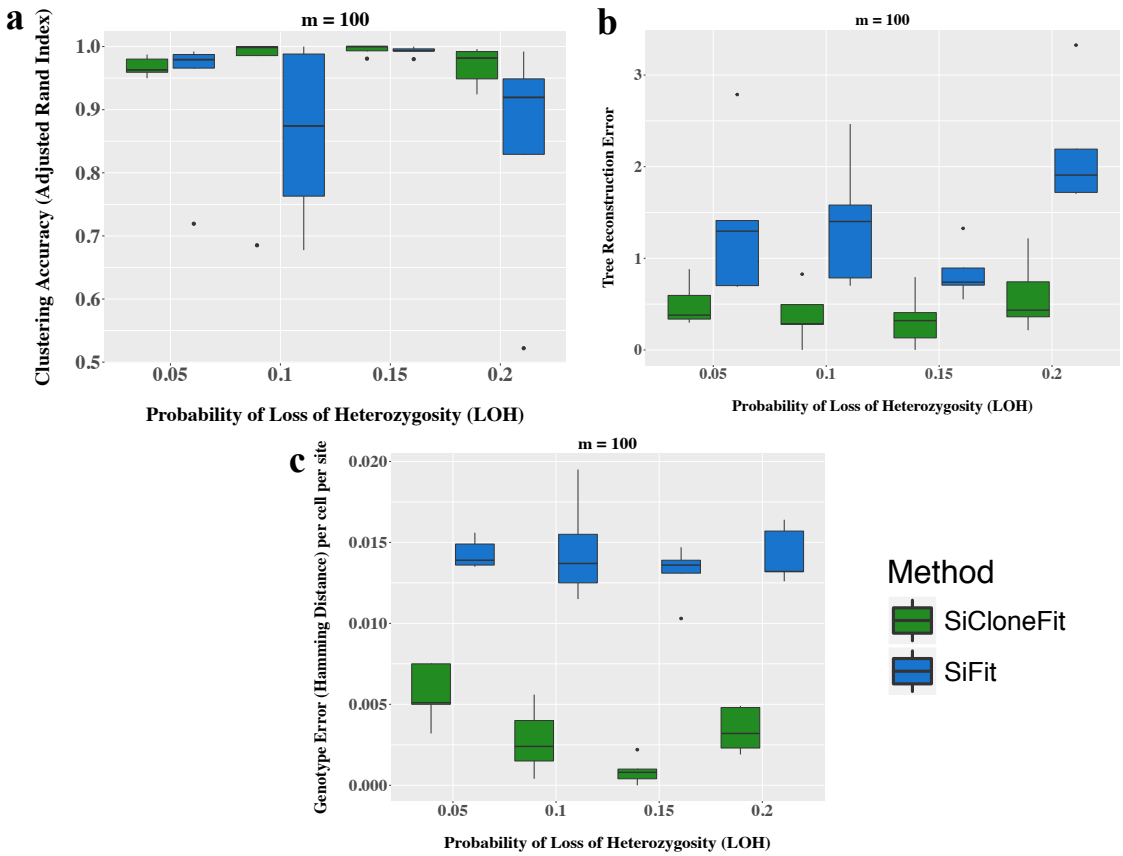
Supplemental Figure S1: **Probabilistic graphical model representing the SiCloneFit model.** The indices and variables of the model are described in Supplemental Table S1 and Supplemental Table S2 respectively. Shaded nodes represent observed values or fixed values, while the un-shaded nodes represent hidden variables and their values are estimated.



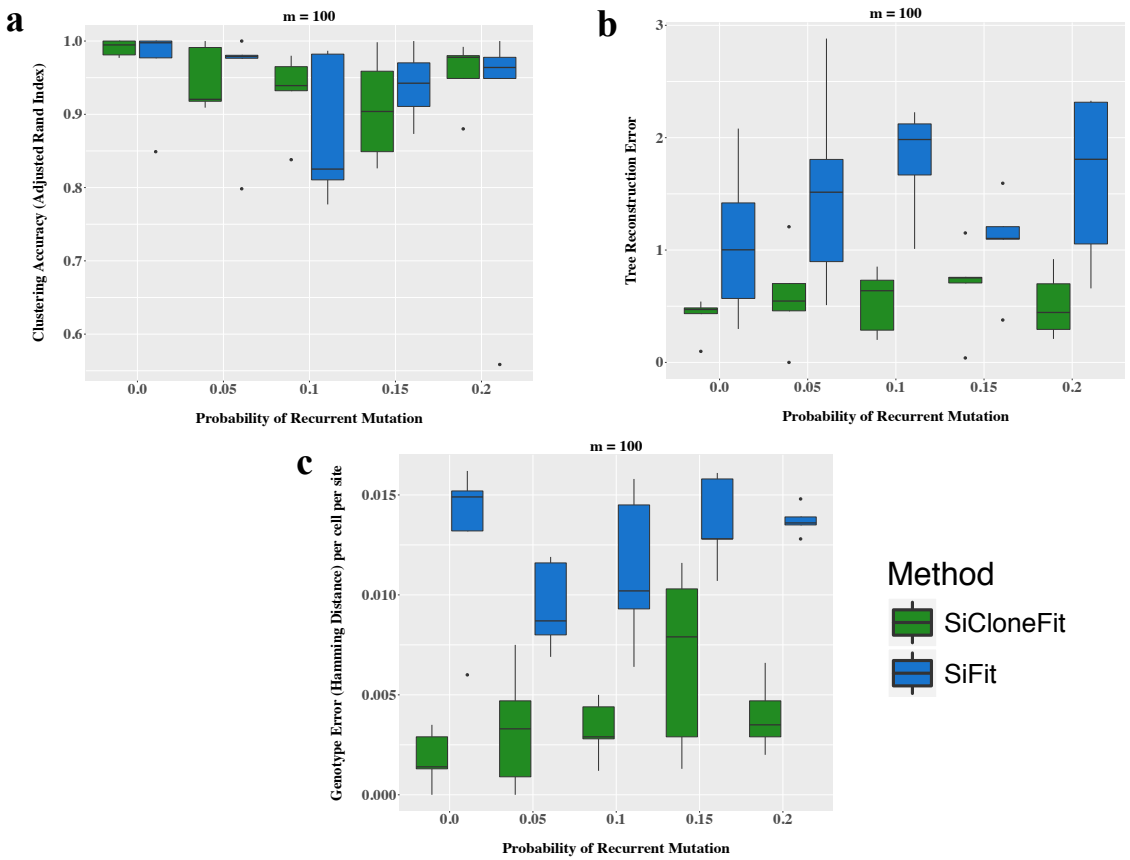
Supplemental Figure S2: **Probabilistic graphical model representing the extended SiCloneFit model for handling doublets.** The new variables introduced in this model are described in Supplemental Table S7. Shaded nodes represent observed values or fixed values, while the un-shaded nodes represent hidden variables and their values are estimated.



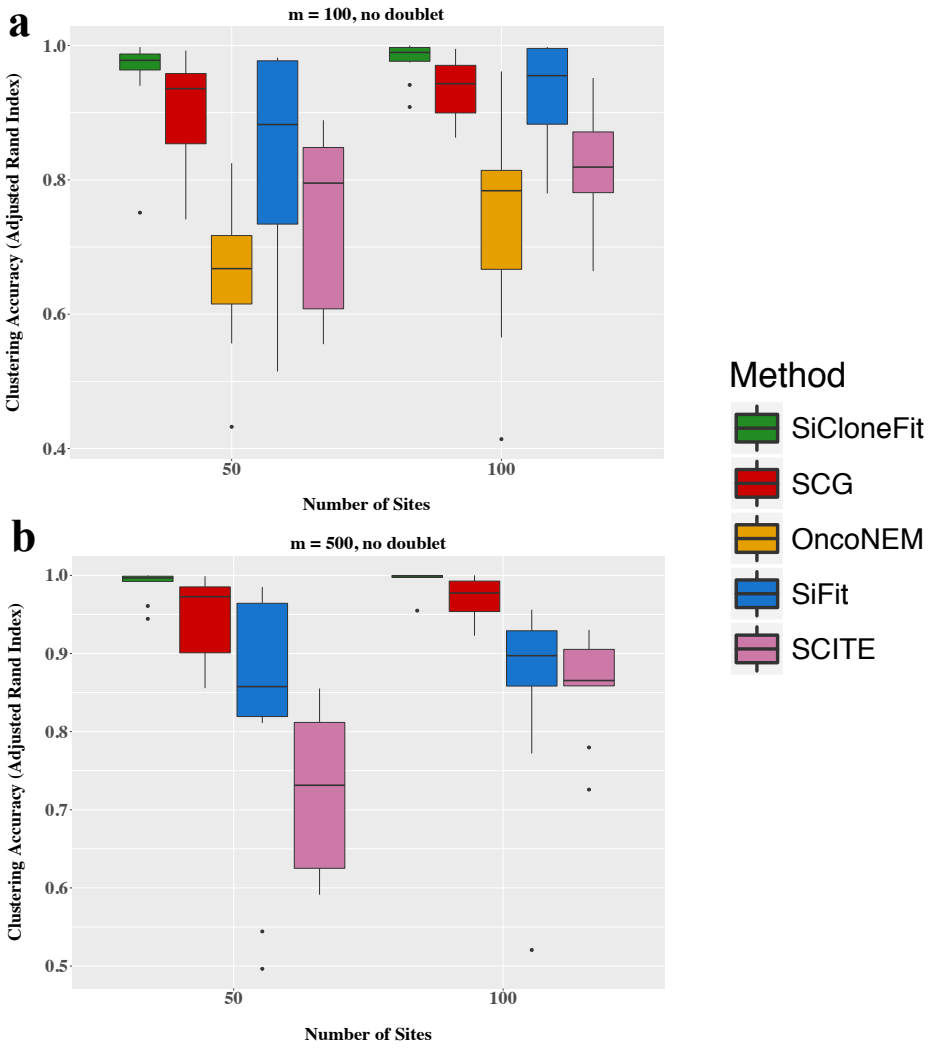
Supplemental Figure S3: **Performance comparison on datasets generated under varying values of probability of deletion.** SiCloneFit's performance is compared against that of SiFit for varying values of the probability of deletion, d. On the x-axis, we have results corresponding to $d \in \{0.05, 0.1, 0.15, 0.2\}$. The number of clones was set to $K = 10$, the number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 5 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering against the ground truth. (b) Comparison of tree reconstruction error in inferring the clonal phylogeny. (c) Comparison of genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



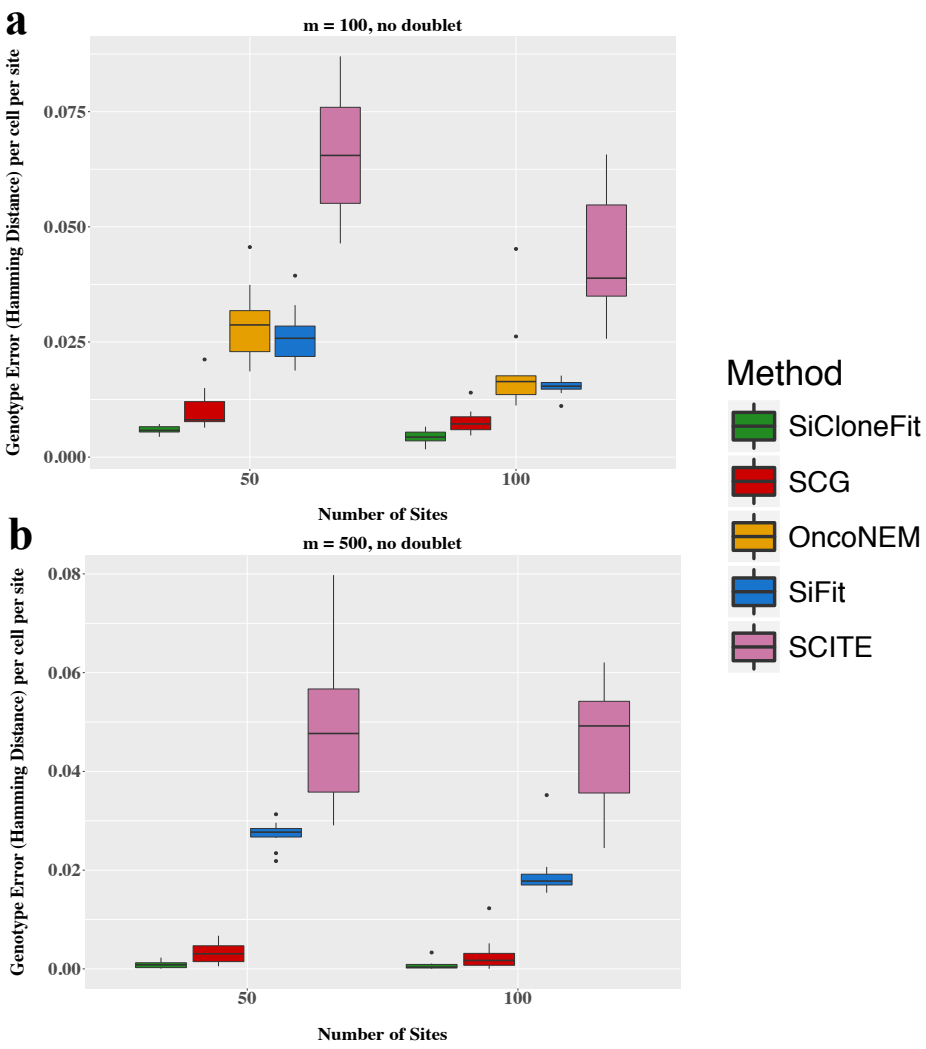
Supplemental Figure S4: **Performance comparison on datasets generated under varying values of probability of LOH.** SiCloneFit's performance is compared against that of SiFit for varying values of the probability of LOH, w. On the x-axis, we have results corresponding to $w \in \{0.05, 0.1, 0.15, 0.2\}$. The number of clones was set to $K = 10$, the number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 5 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering against the ground truth. (b) Comparison of tree reconstruction error in inferring the clonal phylogeny. (c) Comparison of genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



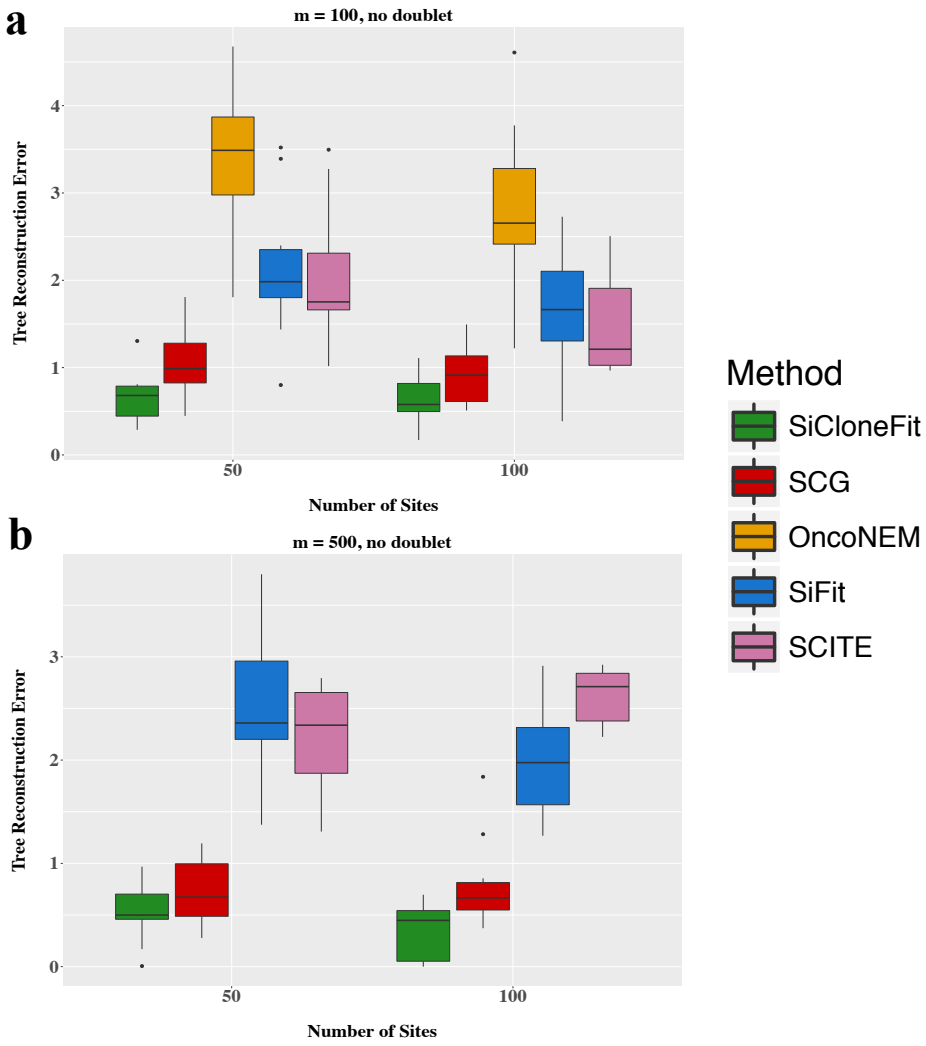
Supplemental Figure S5: **Performance comparison on datasets generated under varying values of the probability of recurrent mutation.** SiCloneFit's performance is compared against that of SiFit for varying values of the probability of recurrent mutation, r. On the x-axis, we have results corresponding to $r \in \{0.0, 0.05, 0.1, 0.15, 0.2\}$. The number of clones was set to $K = 10$, the number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 5 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering against the ground truth. (b) Comparison of tree reconstruction error in inferring the clonal phylogeny. (c) Comparison of genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and the inferred genotype matrix.



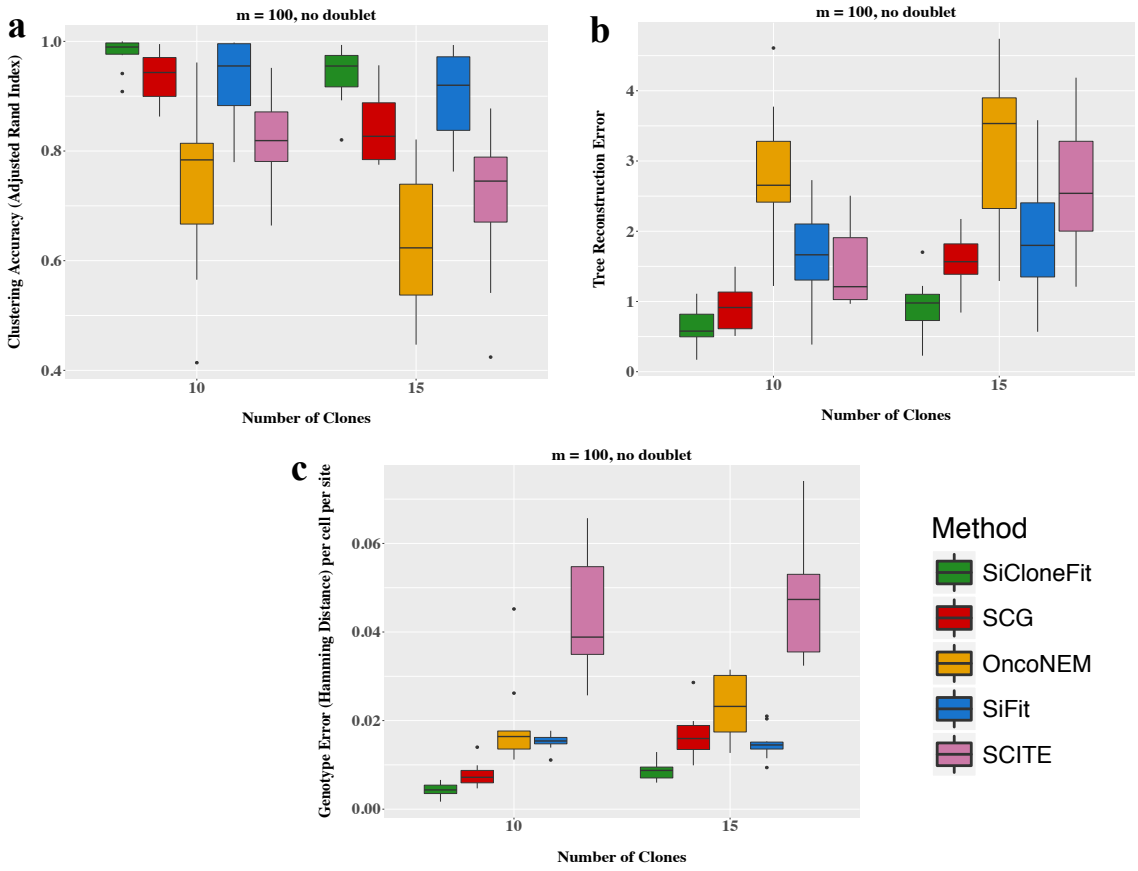
Supplemental Figure S6: **Clustering accuracy on datasets with varying number of cells.** SiCloneFit's clustering accuracy is compared against that of SCG, OncoNEM, SiFit and SCITE. The y-axis denotes the clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering against the ground truth. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for $m = 100$ (comparison against SCG, OncoNEM, SiFit and SCITE). (b) Results for $m = 500$ (comparison against SCG, SiFit and SCITE).



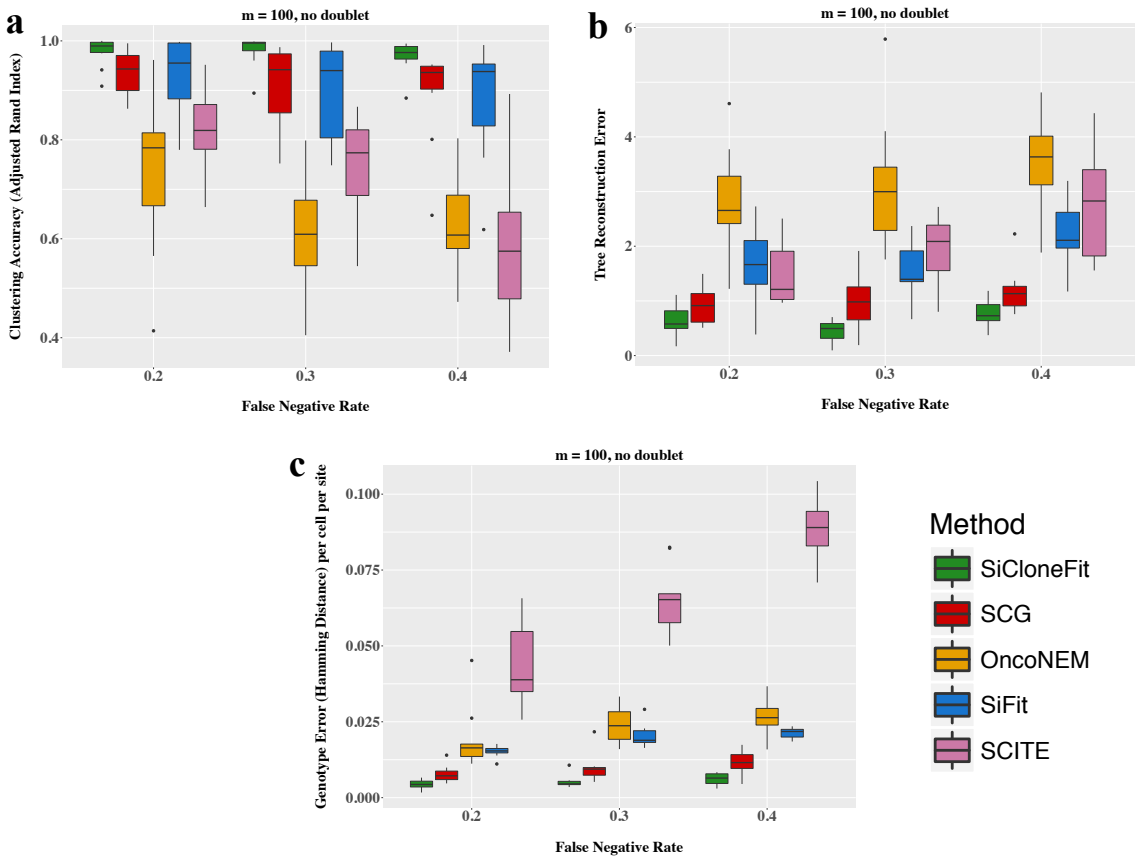
Supplemental Figure S7: **Genotyping performance on datasets with varying number of cells.** SiCloneFit's genotyping performance is compared against that of SCG, OncoNEM, SiFit and SCITE. The y-axis denotes the genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for $m = 100$ (comparison against SCG, OncoNEM, SiFit and SCITE). (b) Results for $m = 500$ (comparison against SCG, SiFit and SCITE).



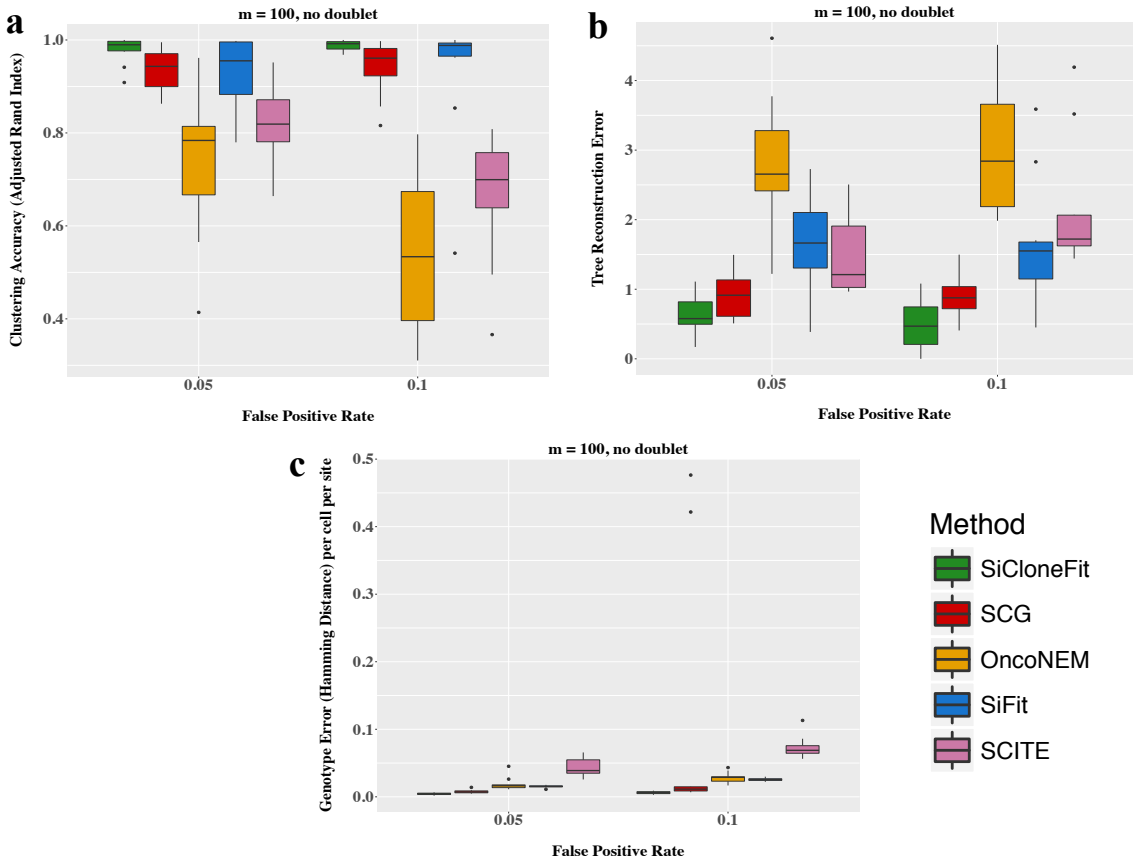
Supplemental Figure S8: **Performance in inferring clonal phylogeny on datasets with varying number of cells.** SiCloneFit's performance in inferring clonal phylogeny is compared against that of SCG, OncoNEM, SiFit and SCITE. The y-axis denotes the tree reconstruction error measured in terms of pairwise cell shortest-path distance between the true clonal phylogeny and inferred clonal phylogeny. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for $m = 100$ (comparison against SCG, OncoNEM, SiFit and SCITE). (b) Results for $m = 500$ (comparison against SCG, SiFit and SCITE).



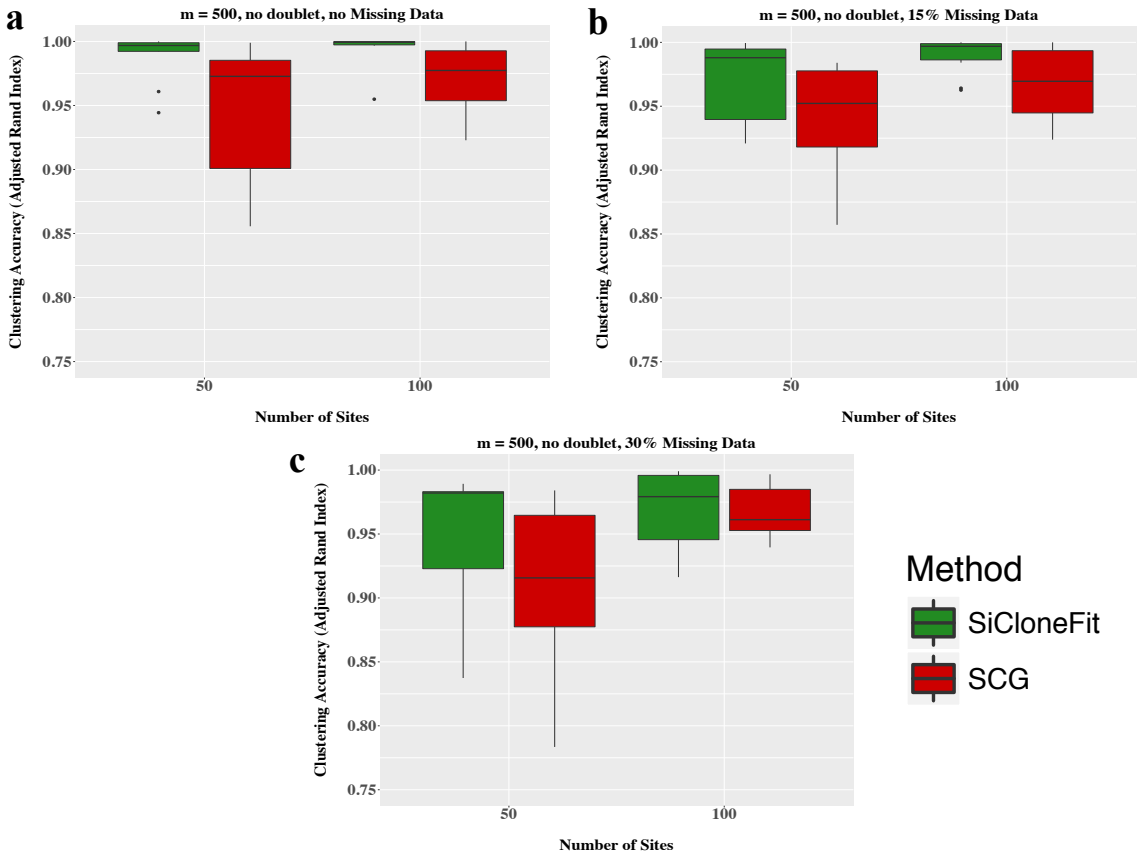
Supplemental Figure S9: **Performance comparison on datasets with varying number of clones.** SiCloneFit's performance is compared against that of SCG, OncoNEM, SiFit and SCITE for varying number of clones. On the x-axis, we have results corresponding to $K = 10$ and $K = 15$. The number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering from the ground truth. (b) Comparison of tree reconstruction error in inferring the clonal phylogeny. (c) Comparison of genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



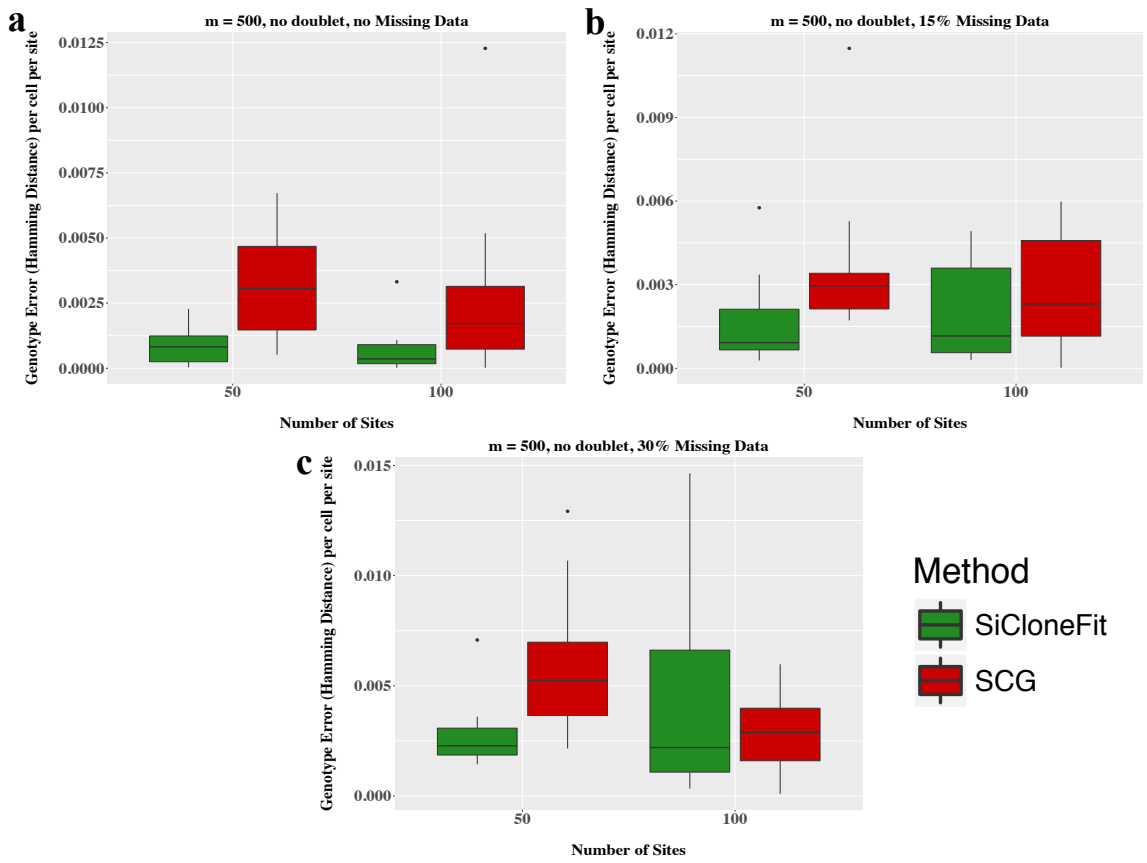
Supplemental Figure S10: **Performance comparison on datasets with varying false negative rate.** SiCloneFit's performance is compared against that of SCG, OncoNEM, SiFit and SCITE for varying false negative rates. On the x-axis, we have results corresponding to $\beta = 0.2$, $\beta = 0.3$ and $\beta = 0.4$. The number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering from the ground truth. (b) Comparison of tree reconstruction error in inferring the clonal phylogeny. (c) Comparison of genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



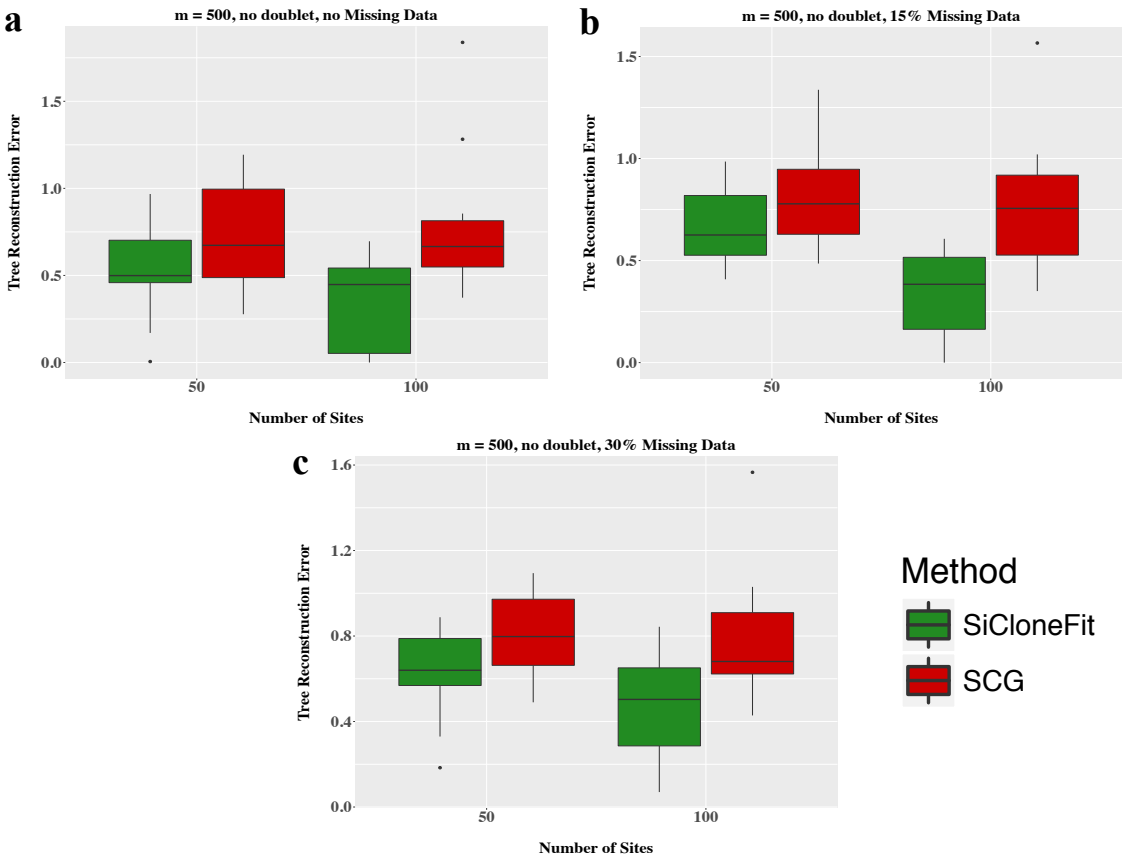
Supplemental Figure S11: **Performance comparison on datasets with varying false positive rate.** SiCloneFit's performance is compared against that of SCG, OncoNEM, SiFit and SCITE for varying false positive rates. On the x-axis, we have results corresponding to $\alpha = 0.05$ and $\alpha = 0.1$. The number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering from the ground truth. (b) Comparison of tree reconstruction error in inferring the clonal phylogeny. (c) Comparison of genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



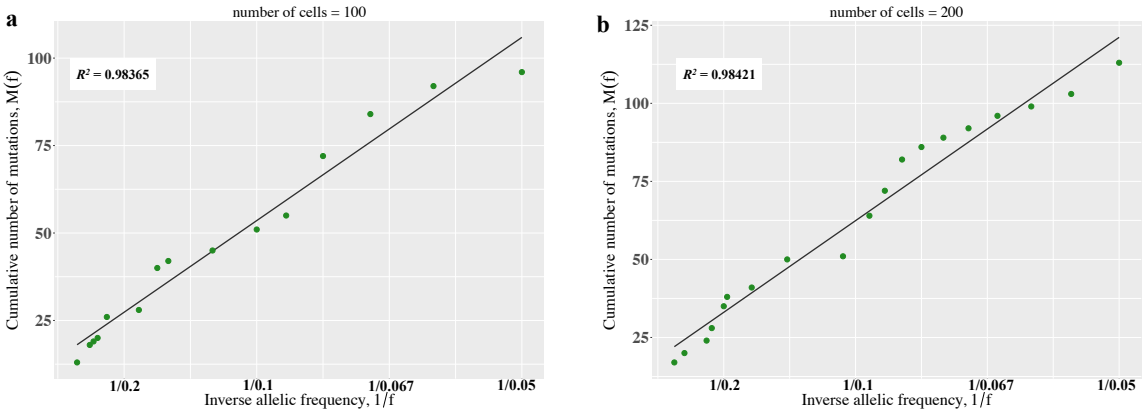
Supplemental Figure S12: **Clustering accuracy on datasets with missing data.** SiCloneFit's clustering accuracy is compared against that of SCG. The y-axis denotes the clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering from the ground truth. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for the datasets without any missing data. (b) Results for datasets with 15% missing data. (c) Results for datasets with 30% missing data.



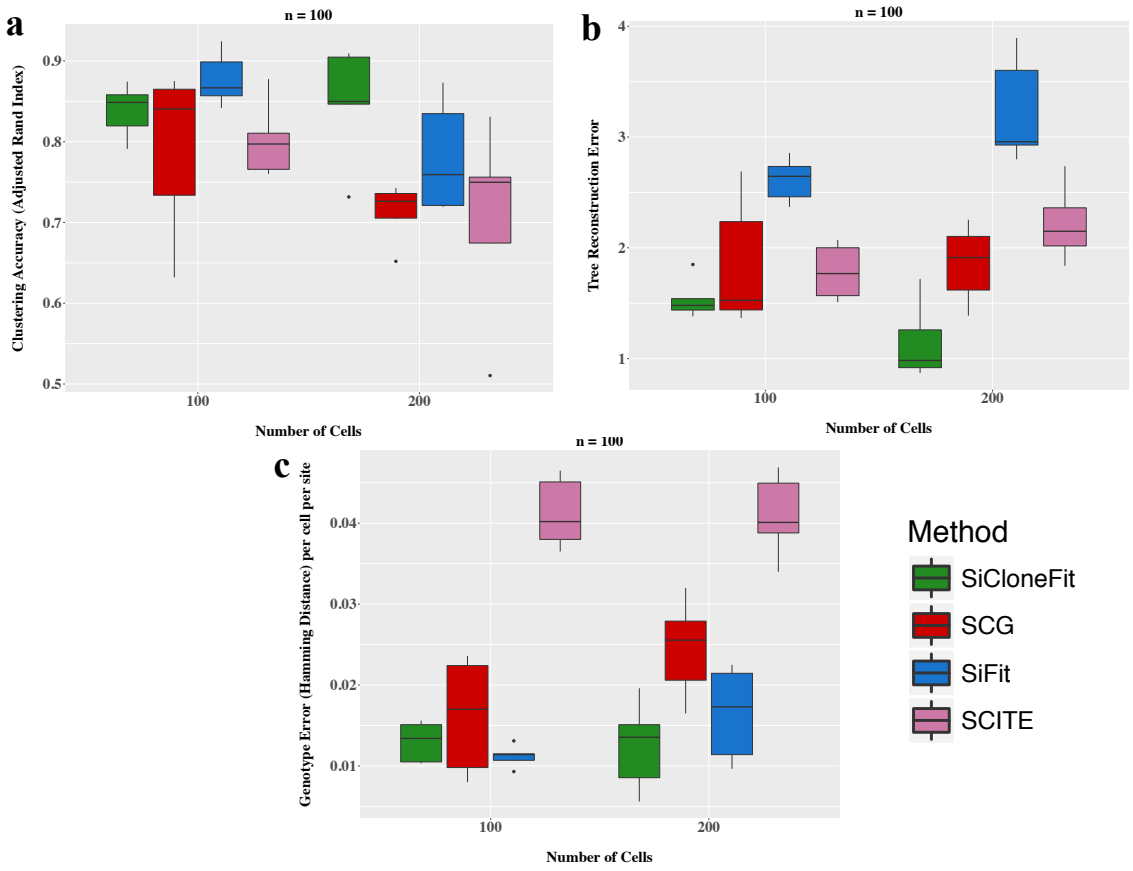
Supplemental Figure S13: **Genotyping performance on datasets with missing data.** SiCloneFit's genotyping performance is compared against that of SCG. The y-axis denotes the genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for the datasets without any missing data. (b) Results for datasets with 15% missing data. (c) Results for datasets with 30% missing data.



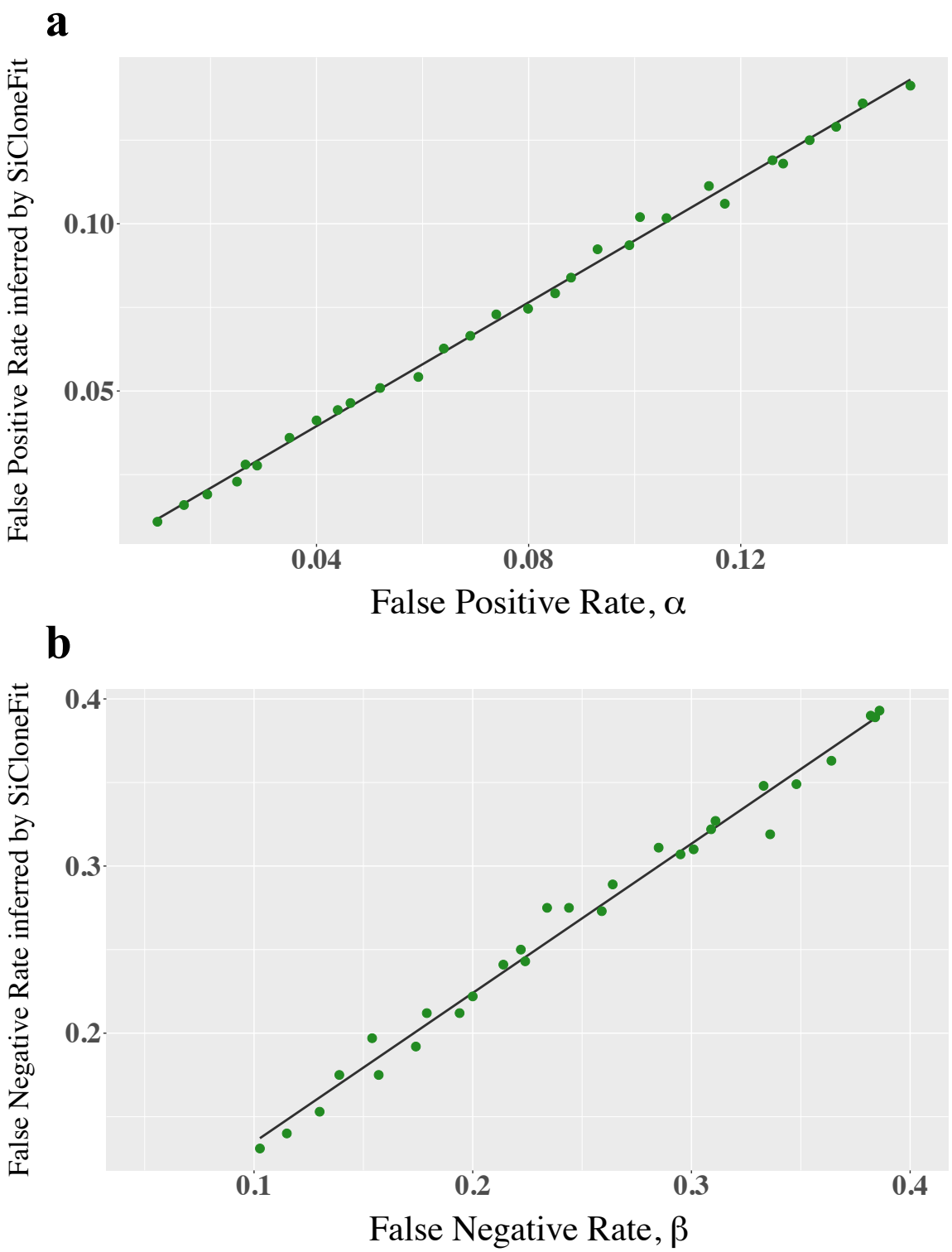
Supplemental Figure S14: **Performance in inferring clonal phylogeny on datasets with missing data.** SiCloneFit's performance in inferring clonal phylogeny is compared against that of SCG. The y-axis denotes the tree reconstruction error measured in terms of pairwise cell shortest-path distance between the true clonal phylogeny and inferred clonal phylogeny. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for the datasets without any missing data. (b) Results for datasets with 15% missing data. (c) Results for datasets with 30% missing data.



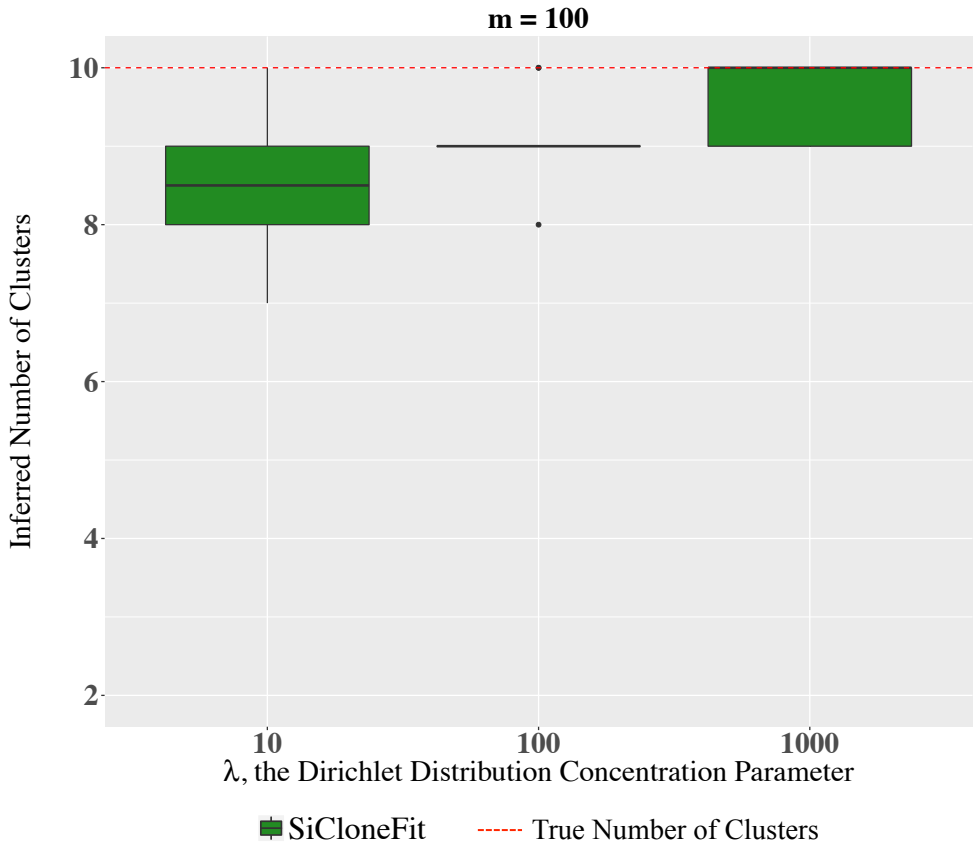
Supplemental Figure S15: **Cumulative distributions $M(f)$ of subclonal mutations under neutral evolution.** The cumulative distributions are linear with $\frac{1}{f}$. R^2 goodness-of-fit measure above the threshold value (0.98) suggested in [28]. Representative simulated dataset consisting of (a) 100 cells and (b) 200 cells.



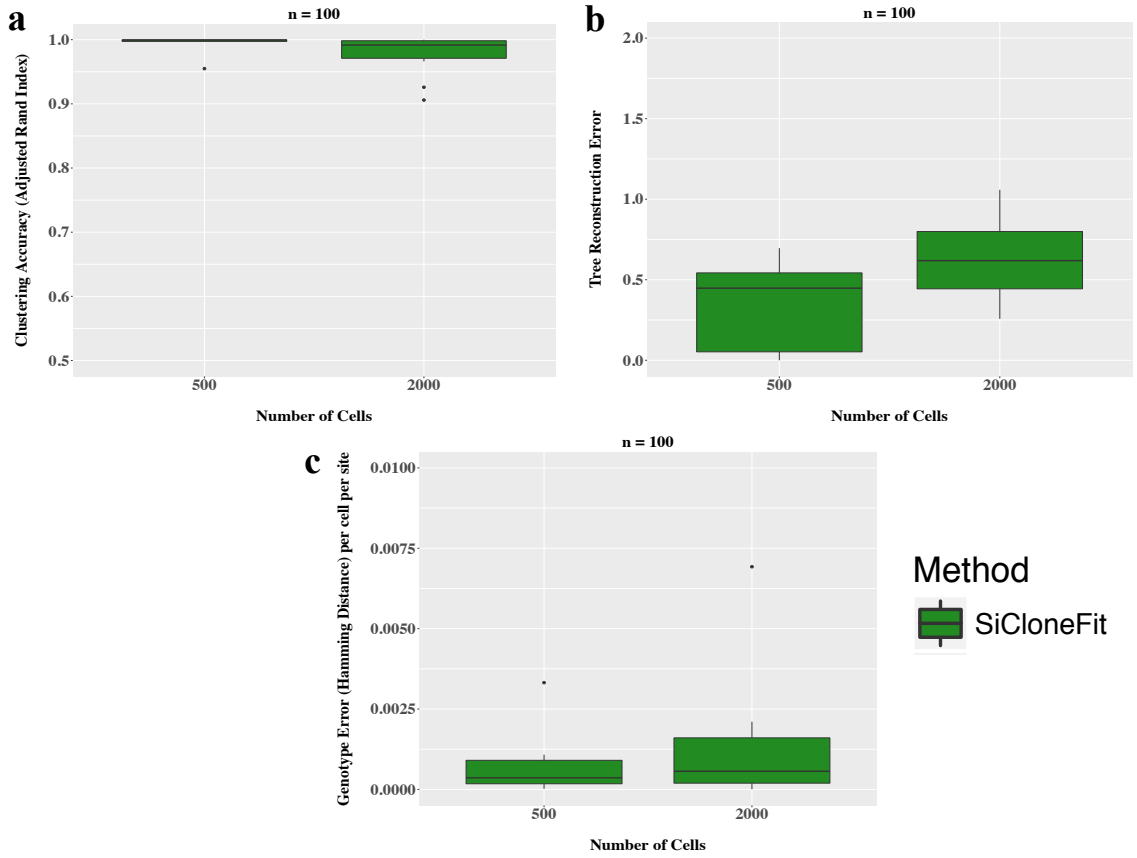
Supplemental Figure S16: **Performance comparison on datasets generated under neutral evolution.** SiCloneFit's performance is compared against that of SCG, SiFit and SCITE on simulated datasets under neutral evolution. On the x-axis, we have results corresponding to $m = 100$ and $m = 200$. The number of clones was set to $K = 20$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 5 simulated datasets. (a) Comparison of clustering accuracy measured in terms of Adjusted Rand Index that compares the inferred clustering against the ground truth. (b) Comparison based on the genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix. (c) Comparison based on the tree reconstruction error measured in terms of pairwise cell shortest-path distance between the true clonal phylogeny and inferred clonal phylogeny.



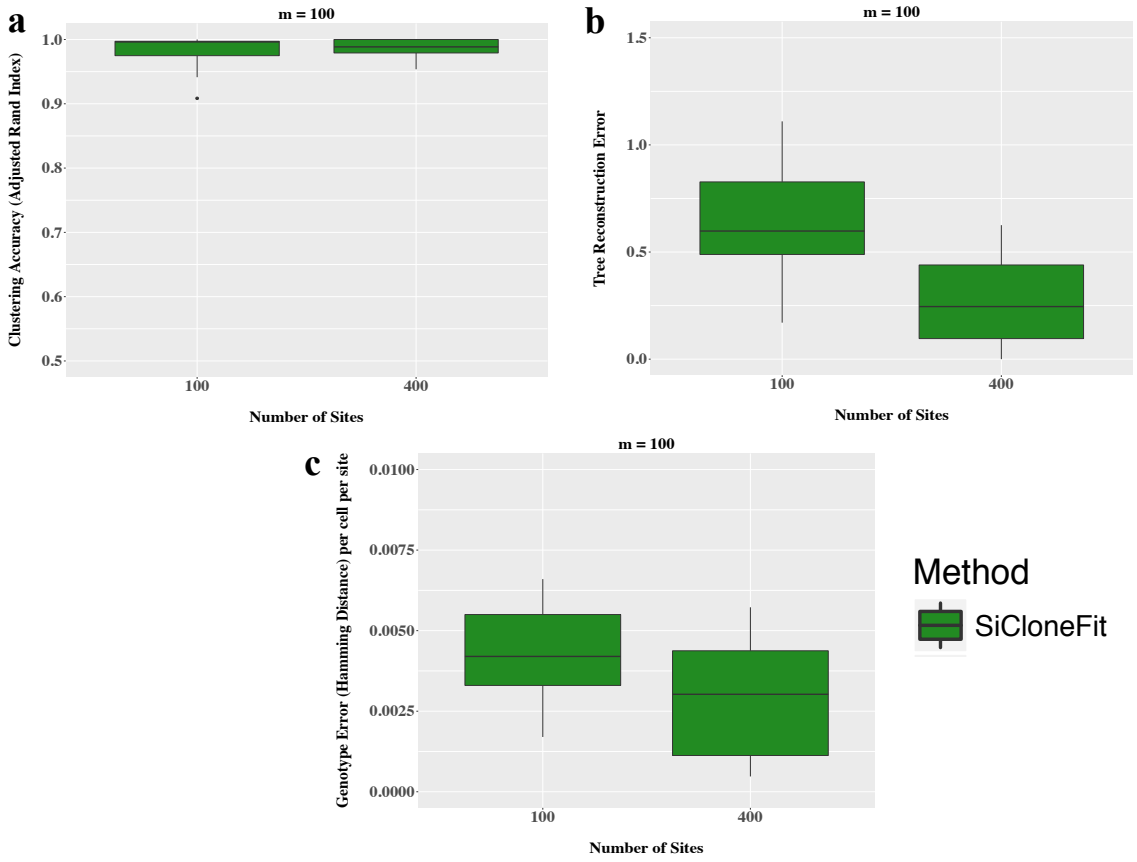
Supplemental Figure S17: **Estimation of error rates by SiCloneFit.** Error rates inferred by SiCloneFit are compared against the true error rates used for generating the data. The green dots correspond to the results of SiCloneFit. The black line represents a fitted regression line. (a) SiCloneFit's estimate of false positive rates is compared against the true false positive rates used during the simulation. (b) False negative rates inferred by SiCloneFit are compared to the true false negative rate used for generating the dataset.



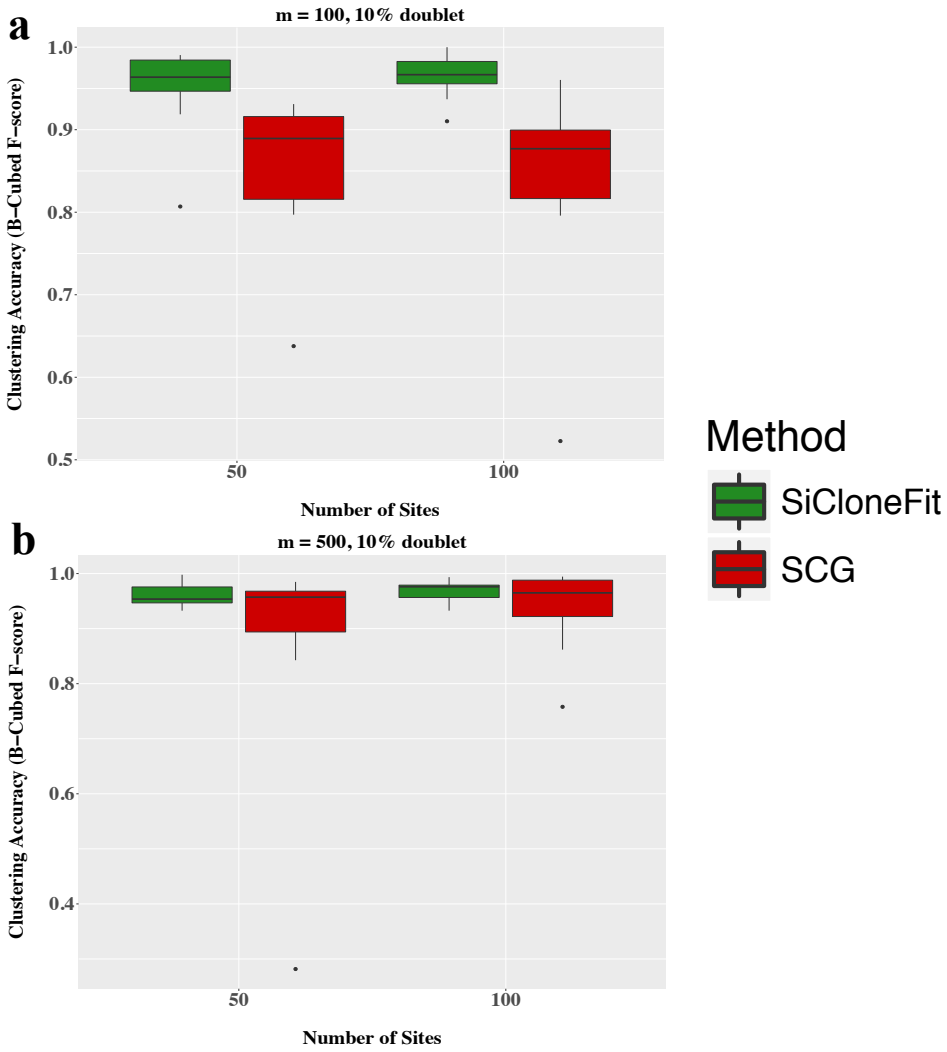
Supplemental Figure S18: **Estimation of number of clusters by SiCloneFit.** Number of clusters inferred by SiCloneFit is compared against the true number of clusters in the simulated datasets. On the x-axis, we have results corresponding to $\lambda = 10$, $\lambda = 100$ and $\lambda = 1000$. λ denotes the concentration parameter of the Dirichlet distribution used for sampling the observed prevalences of the clones. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters.



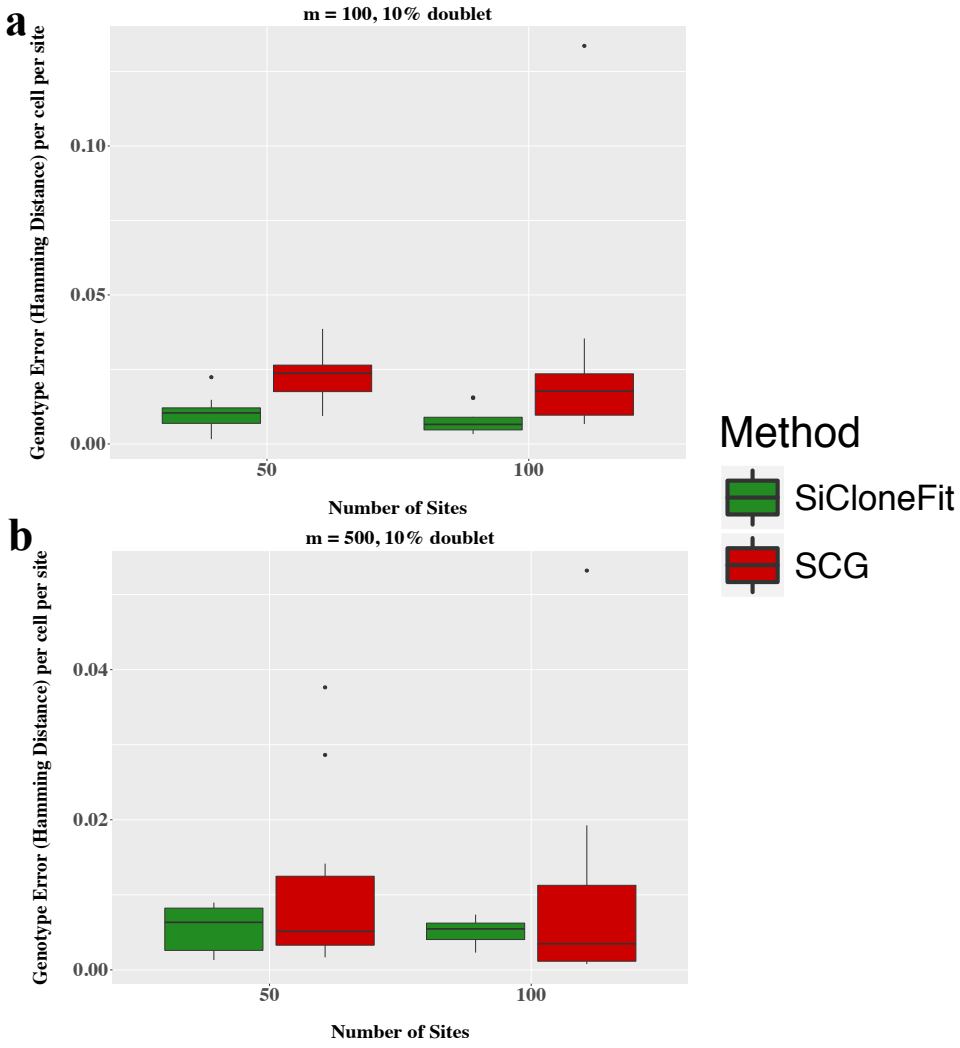
Supplemental Figure S19: **Scalability of SiCloneFit for large number of cells.** Performance of SiCloneFit for datasets containing large number of cells. On the x-axis, we have results corresponding to $m = 500$ and $m = 2000$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering from the ground truth. (b) Tree reconstruction error in inferring the clonal phylogeny. (c) Genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



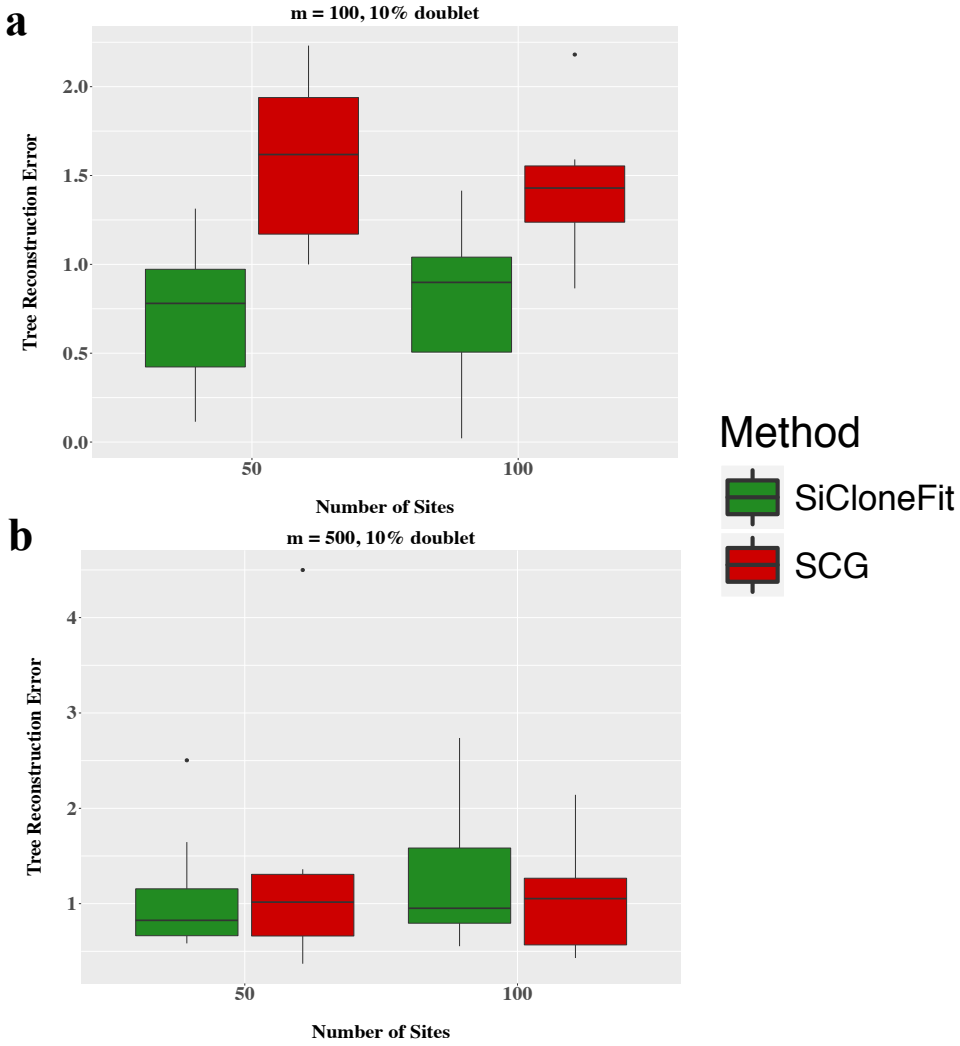
Supplemental Figure S20: **Scalability of SiCloneFit for large number of genomic sites.** Performance of SiCloneFit for datasets with large number of mutation sites. On the x-axis, we have results corresponding to $n = 100$ and $n = 400$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Clustering accuracy measured in terms of adjusted rand index that compares the inferred clustering from the ground truth. (b) Tree reconstruction error in inferring the clonal phylogeny. (c) Genotyping error measured in terms of hamming distance per cell per site between the true genotype matrix and inferred genotype matrix.



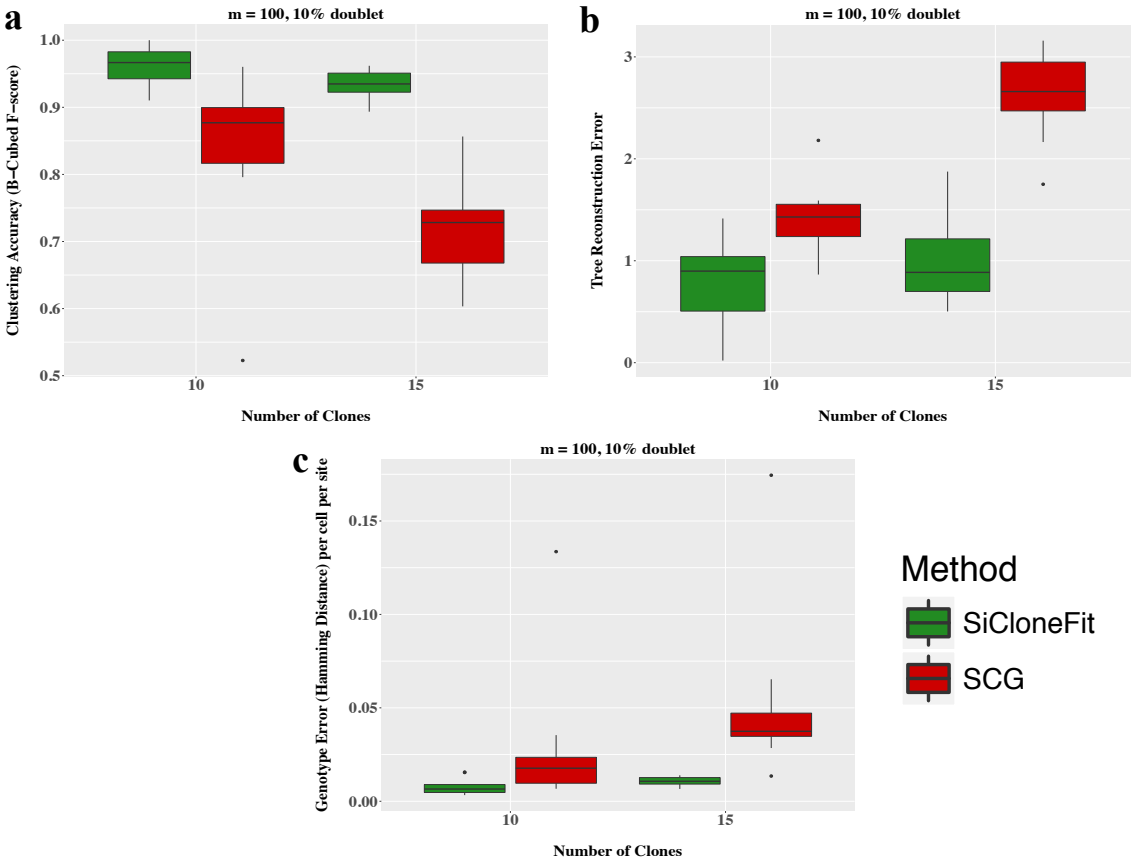
Supplemental Figure S21: **Clustering accuracy on datasets containing doublets with varying number of cells.** SiCloneFit's clustering accuracy is compared against that of SCG for datasets containing doublets. The y-axis denotes the clustering accuracy measured in terms of B-Cubed F-score that compares the inferred clustering from the ground truth. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. The top panel shows the results for $m = 100$ and the bottom panel shows the results for $m = 500$.



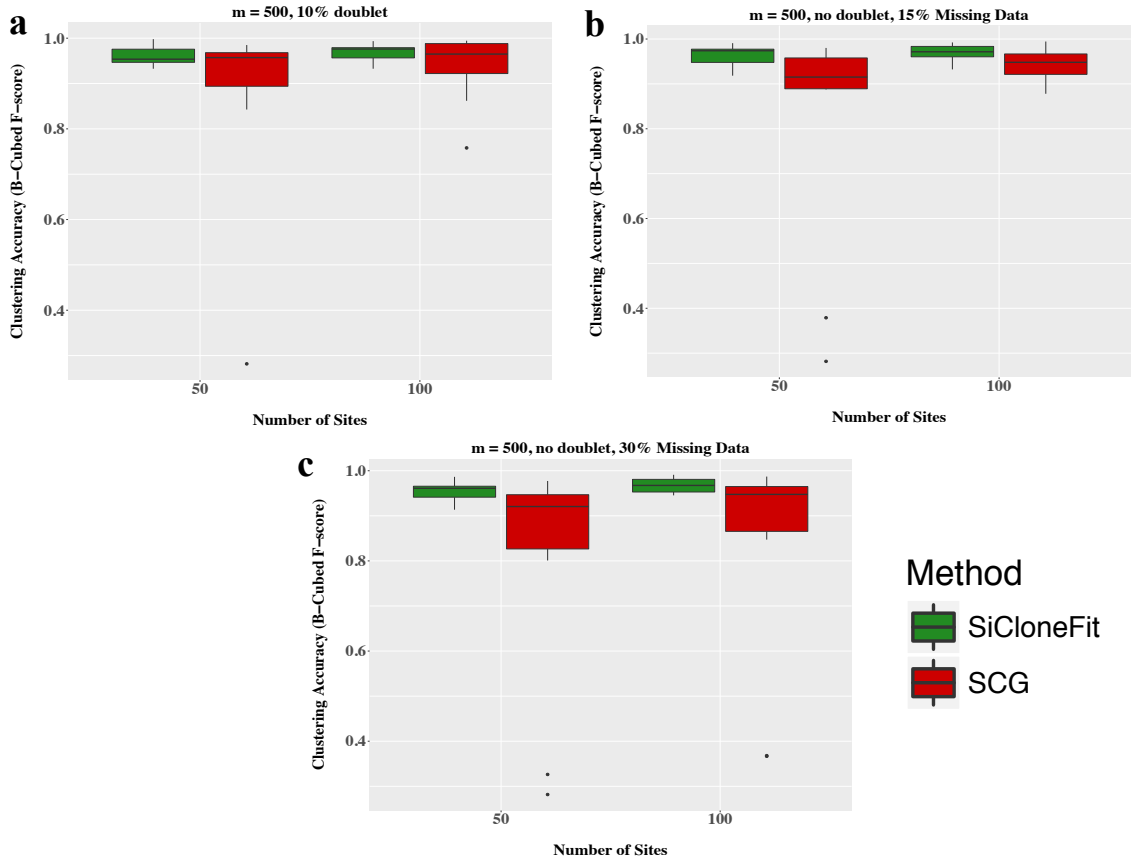
Supplemental Figure S22: **Genotyping performance on datasets containing doublets with varying number of cells.** SiCloneFit's genotyping performance is compared against that of SCG for datasets containing doublets. The y-axis denotes the genotyping error measured in terms of hamming distance between the true genotype matrix and inferred genotype matrix excluding the inferred doublets. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. The top panel shows the results for $m = 100$ and the bottom panel shows the results for $m = 500$.



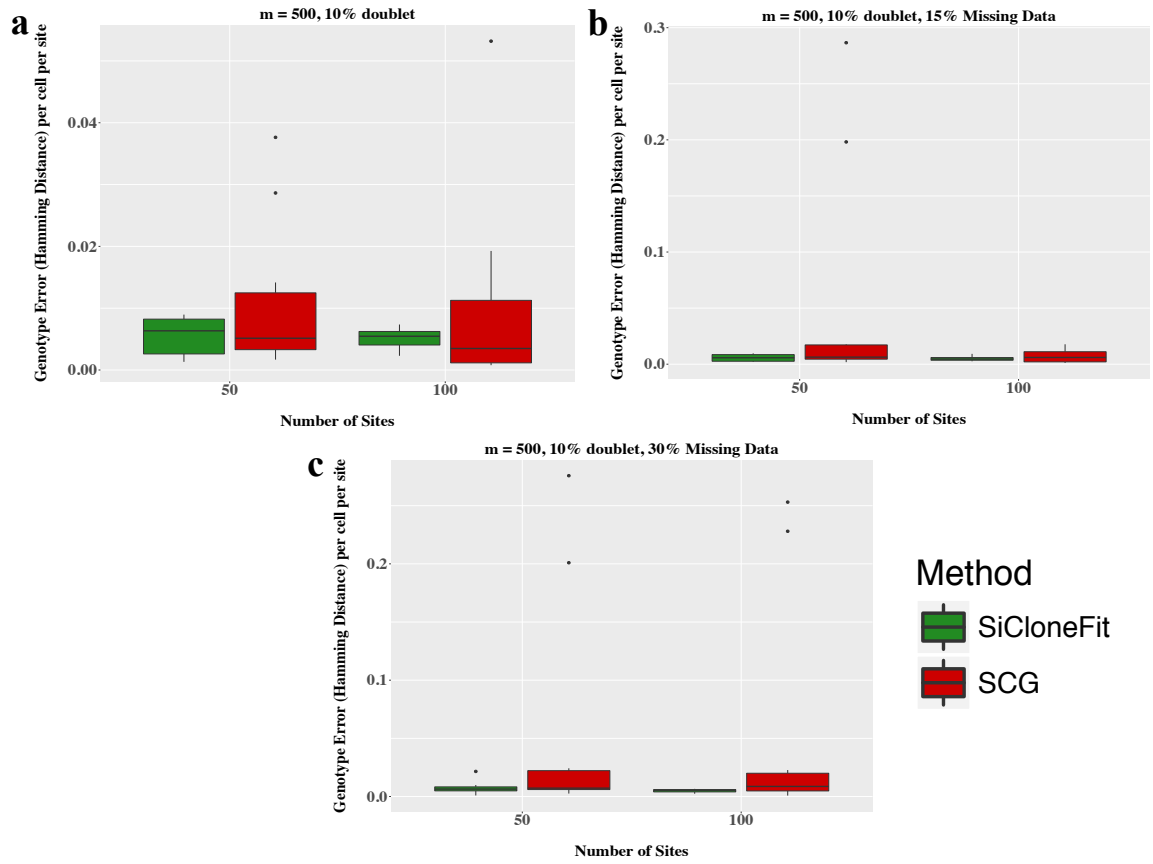
Supplemental Figure S23: **Performance in inferring clonal phylogeny on datasets containing doublets with varying number of cells.** SiCloneFit's performance in inferring clonal phylogeny is compared against that of SCG for datasets that contain doublets. The y-axis denotes the tree reconstruction error measured in terms of pairwise cell shortest-path distance between the true clonal phylogeny and inferred clonal phylogeny excluding the inferred doublets. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. The top panel shows the results for $m = 100$ and the bottom panel shows the results for $m = 500$.



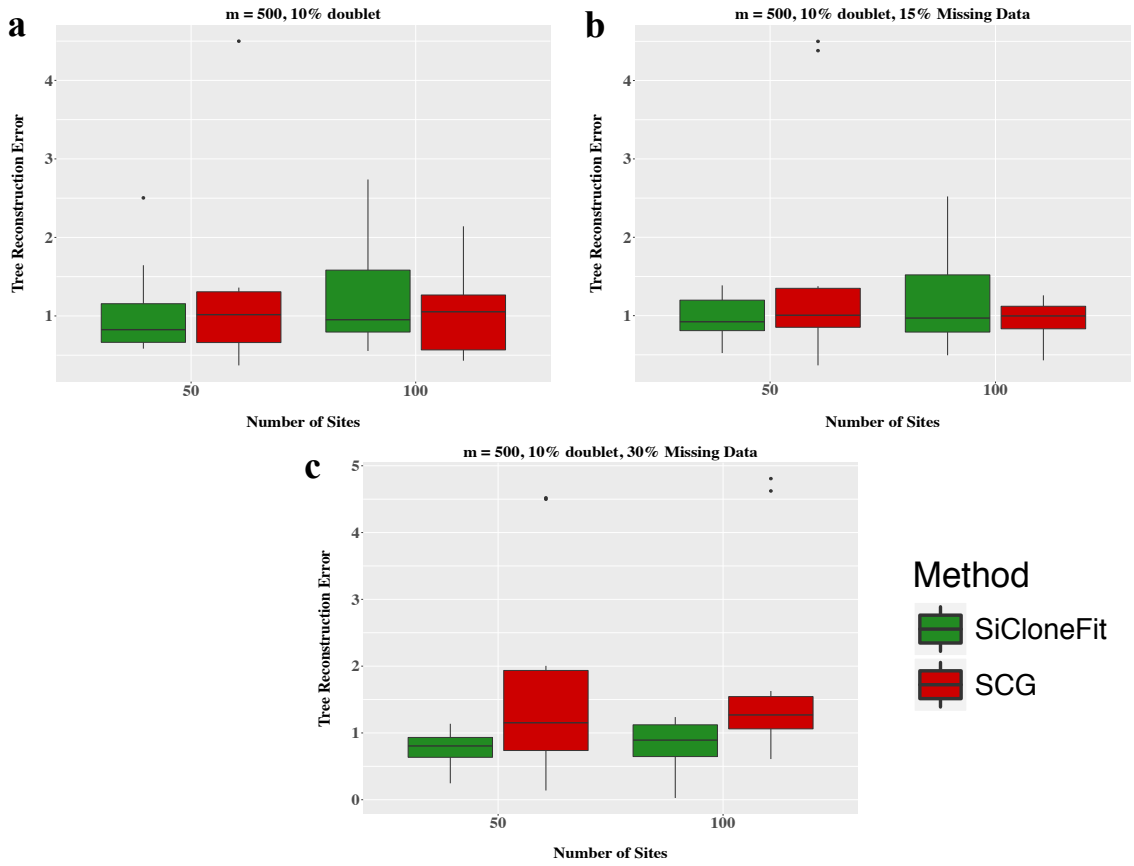
Supplemental Figure S24: **Performance comparison on datasets containing doublets with varying number of clones.** SiCloneFit's performance is compared against that of SCG on datasets containing doublets for varying number of clones. On the x-axis, we have results corresponding to $K = 10$ and $K = 15$. The number of cells was set to $m = 100$, and the number of sites was set to $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Comparison of clustering accuracy measured in terms of B-Cubed F-score that compares the inferred clustering from the ground truth. (b) Comparison based on the performance in inferring the clonal phylogeny. (c) Comparison based on the genotyping error measured in terms of hamming distance between the true genotype matrix and inferred genotype matrix.



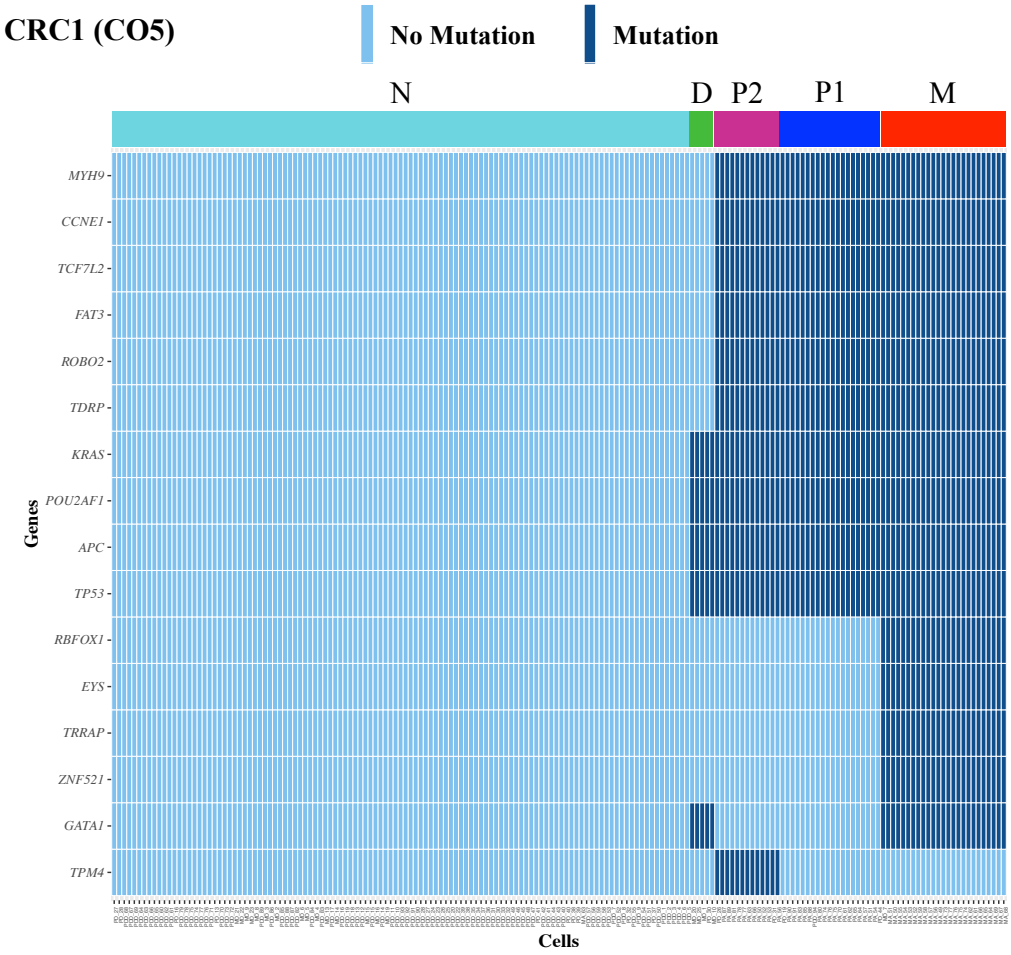
Supplemental Figure S25: **Clustering accuracy on datasets containing doublets and missing entries.** SiCloneFit's clustering accuracy is compared against that of SCG for datasets that contain doublets as well as missing values. The y-axis denotes the clustering accuracy measured in terms of B-cubed F-score that compares the inferred clustering from the ground truth excluding inferred doublets. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for the datasets without any missing data. (b) Results for the datasets with 15% missing data and (c) Results for the datasets with 30% missing data.



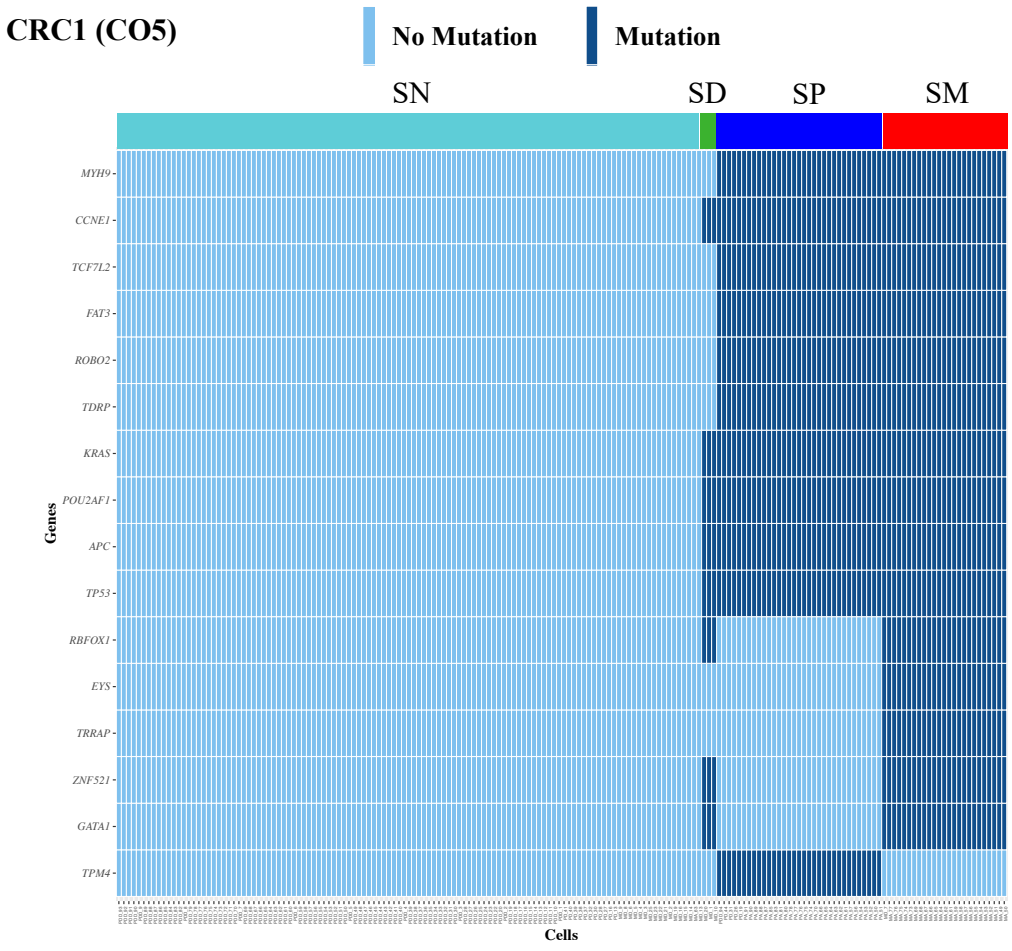
Supplemental Figure S26: **Genotyping performance on datasets containing doublets and missing entries.** SiCloneFit's genotyping performance is compared against that of SCG for datasets that contain doublets as well as missing values. The y-axis denotes the genotyping error measured in terms of hamming distance between the true genotype matrix and inferred genotype matrix. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for the datasets without any missing data. (b) Results for the datasets with 15% missing data and (c) Results for the datasets with 30% missing data.



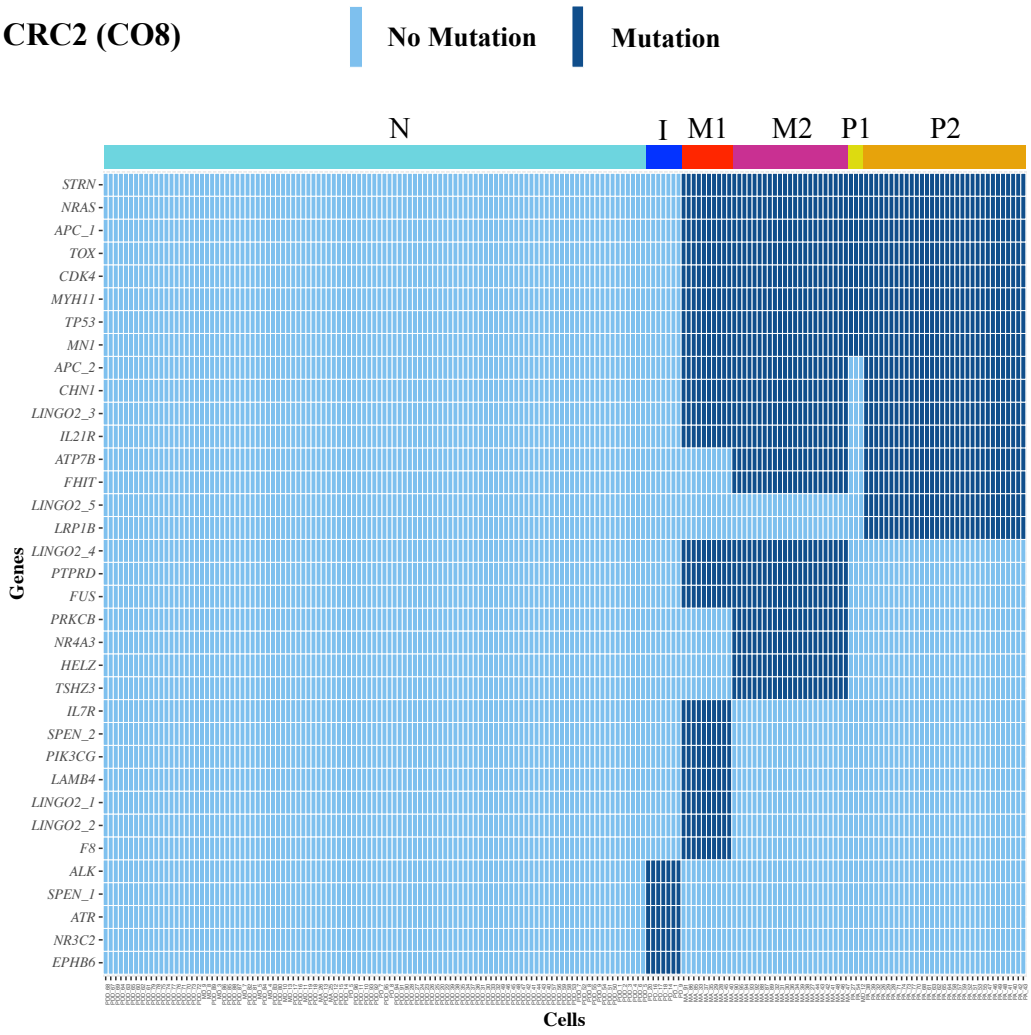
Supplemental Figure S27: **Performance in inferring clonal phylogeny on datasets containing doublets and missing entries.** SiCloneFit's performance in inferring clonal phylogeny is compared against that of SCG for datasets that contain doublets as well as missing values. The y-axis denotes the tree reconstruction error measured in terms of pairwise cell shortest-path distance between the true clonal phylogeny and inferred clonal phylogeny. On the x-axis, we have results corresponding to $n = 50$ and $n = 100$. Each box plot summarizes results for 10 simulated datasets with varying clonal phylogeny and varying size of clonal clusters. (a) Results for the datasets without any missing data. (b) Results for the datasets with 15% missing data and (c) Results for the datasets with 30% missing data.



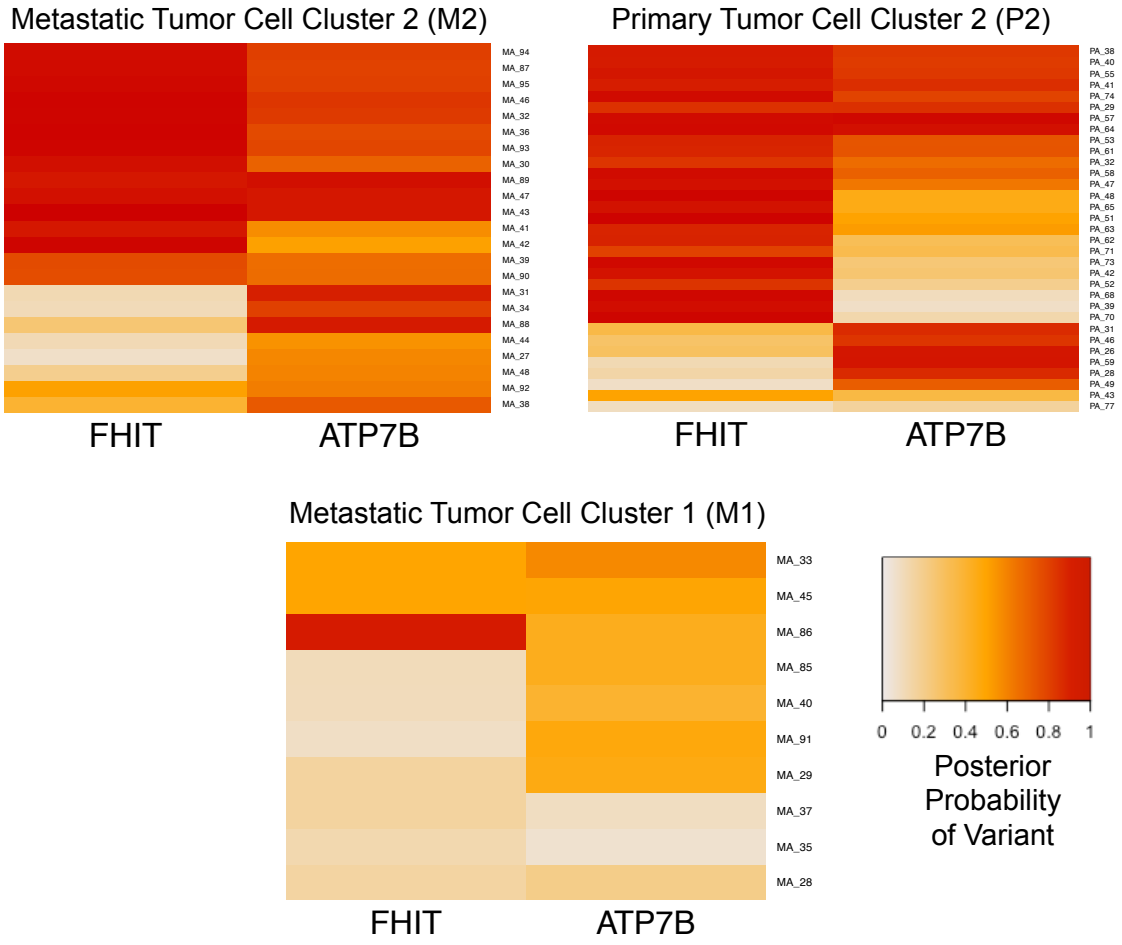
Supplemental Figure S28: **Inferred genotypes of cells from the posterior samples obtained using SiCloneFit for metastatic colorectal cancer patient CRC1.**



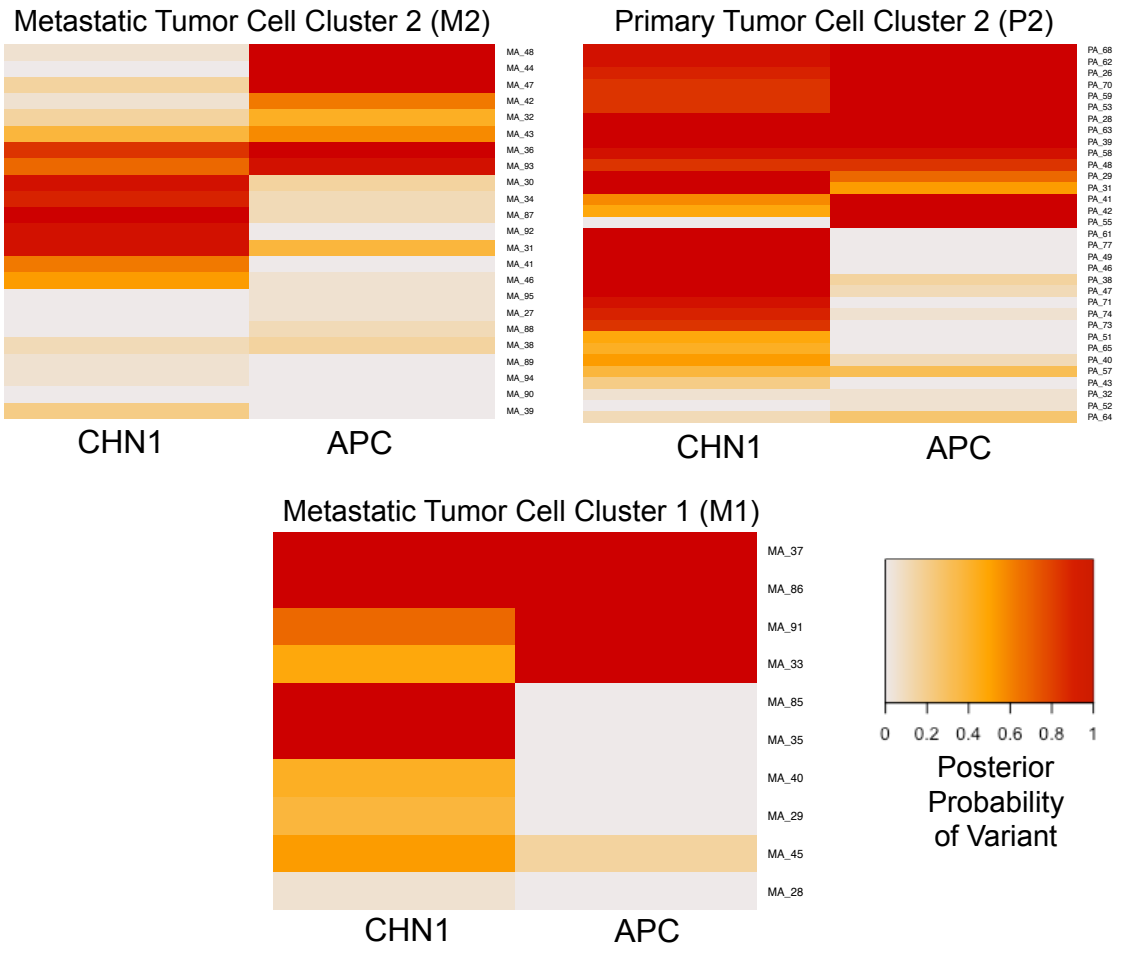
Supplemental Figure S29: **Clonal genotypes of cells inferred using SCG for metastatic colorectal cancer patient CRC1.**



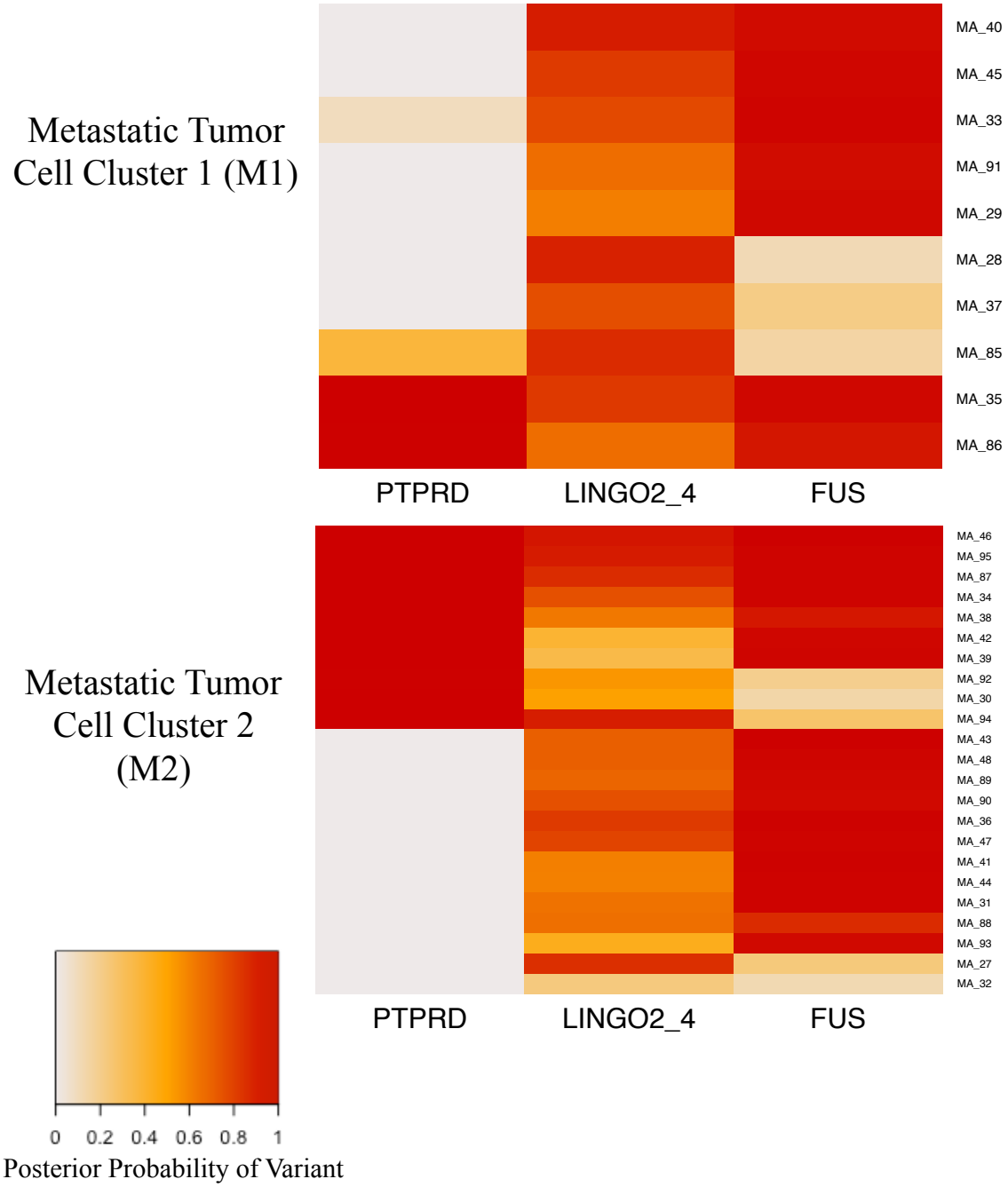
Supplemental Figure S30: **Inferred genotypes of cells from the posterior samples obtained using SiCloneFit for metastatic colorectal cancer patient CRC2.**



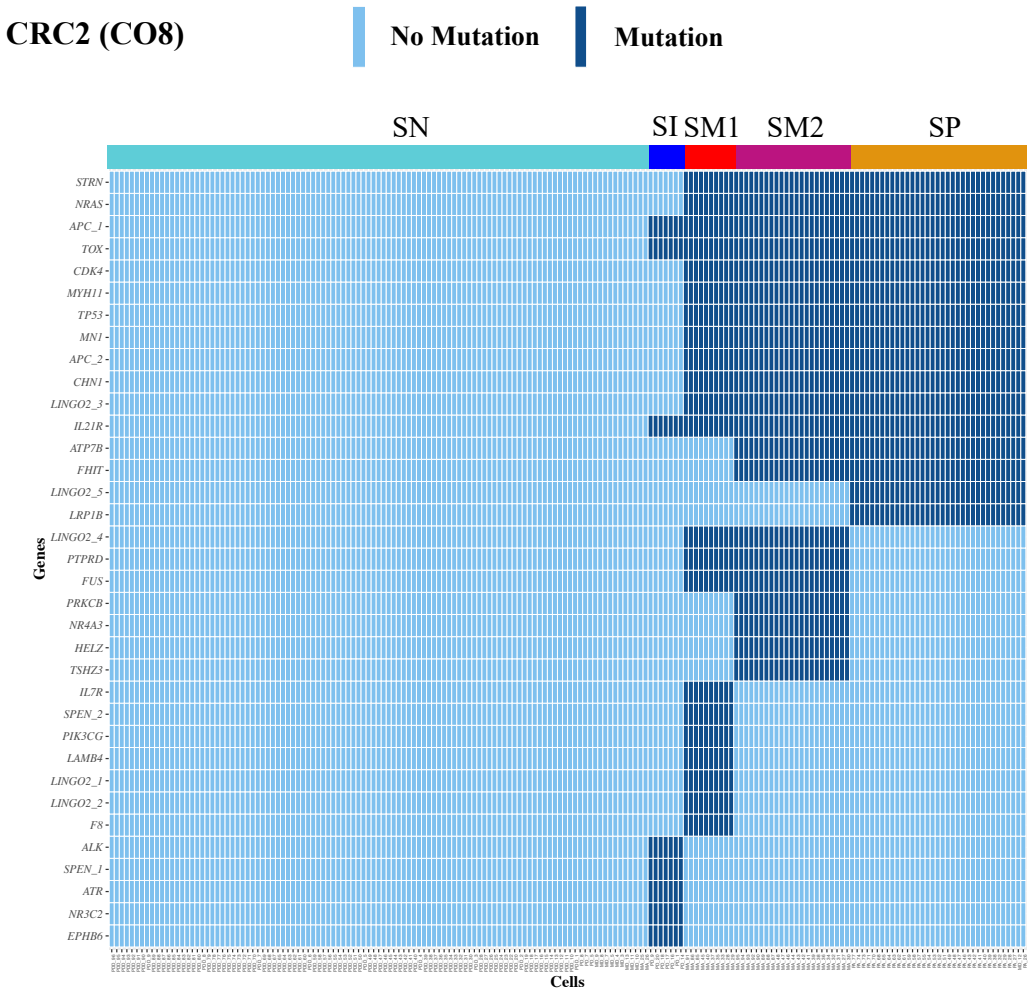
Supplemental Figure S31: **Probability heatmap of the *FHIT* and *ATP7B* mutations in CRC2.** Heatmaps of the posterior probabilities of two bridge mutations (*FHIT*, and *ATP7B*) in patient CRC2 are listed for the primary and metastatic tumor clusters. Both SiCloneFit and SCITE identify these two mutations as 'bridge mutations' between the two metastatic divergence events. Heatmaps for the two metastatic subclones (M1 and M2) and the primary subclone (P2) are shown separately. Each variant is colored in the heatmap based on the corresponding posterior probability value.



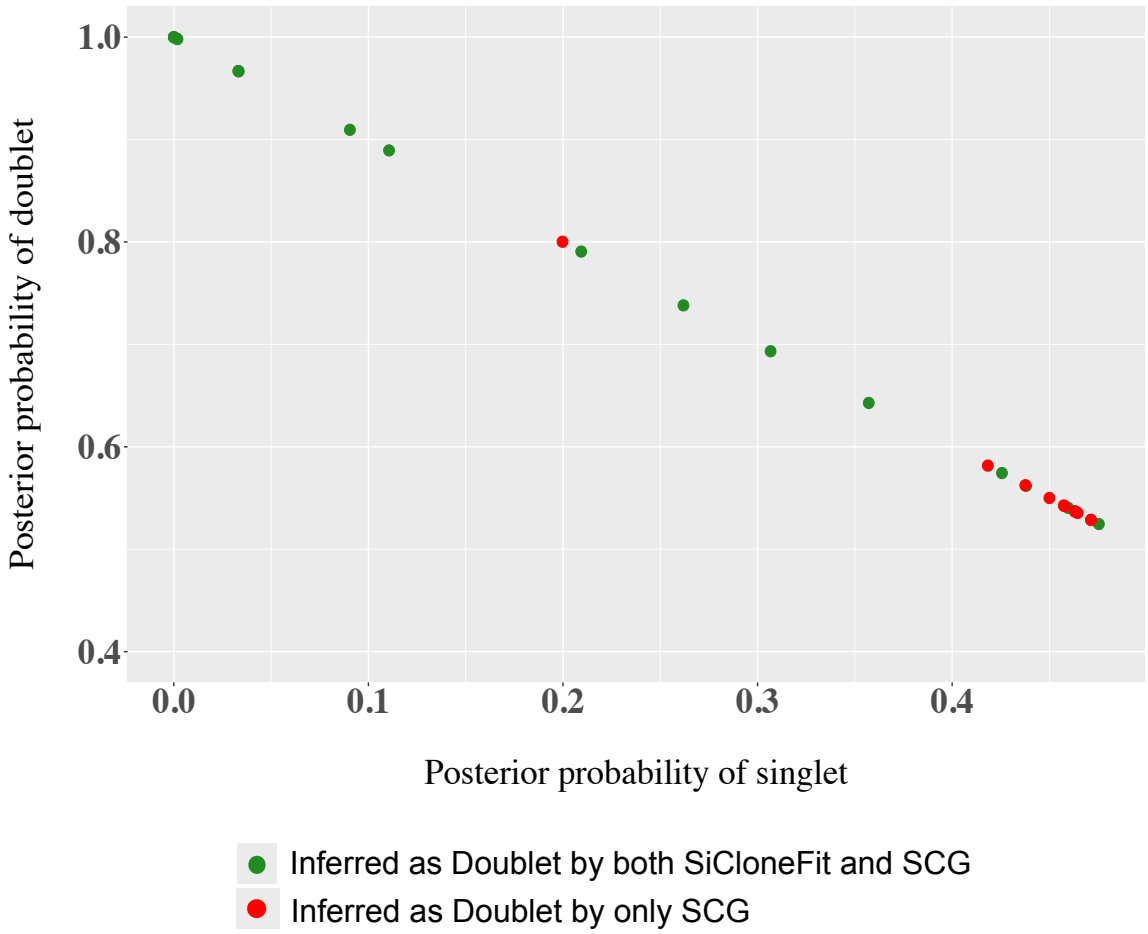
Supplemental Figure S32: **Probability heatmap of the *CHN1*, and *APC* mutations in CRC2**. Heatmaps of the posterior probabilities of two mutations (*CHN1*, and *APC*) in patient CRC2 are listed for the primary and metastatic tumor clusters. These two mutations were identified as 'bridge mutations' in the original study when using SCITE, however, SiCloneFit placed them to occur before any metastatic divergence (classifying them as 'non-bridge'). Heatmaps for the two metastatic subclones (M1 and M2) and the primary subclone (P2) are shown separately. Each variant is colored in the heatmap based on the corresponding posterior probability value.



Supplemental Figure S33: **Probability heatmap of the recurrent mutations in CRC2**. Heatmaps of the posterior probabilities of the three recurrent mutations (*PTPRD*, *FUS* and *LINGO2*) in patient CRC2 are listed for the metastatic tumor cells. Heatmaps for the two metastatic subclones (M1 and M2) are shown separately. Each variant is colored in the heatmap based on the corresponding posterior probability value.



Supplemental Figure S34: **Clonal genotypes of cells inferred using SCG for metastatic colorectal cancer patient CRC2.**



Supplemental Figure S35: **Posterior probabilities of the doublets inferred by SCG and SiCloneFit from the high grade serous ovarian cancer dataset.** The dataset consisted of 370 cells and 43 somatic mutations. The posterior probabilities are computed by SCG.

4 Supplemental Tables

Supplemental Table S1: Indices used in SiCloneFit model (Supplemental Fig. S1).

Index	Range	Description
i	$\{1, \ldots, n\}$	Index of SNV site
j	$\{1, \ldots, m\}$	Index of Single-cell sample
k	$\{1, \ldots, \infty\}$	Index of Clone (cluster)

Supplemental Table S2: Variables used in SiCloneFit model (Supplemental Fig. S1).

Variable	Range	Description
α_0	$(0, \infty)$	Model parameter for the Chinese Restaurant Process (CRP) model
c_j	$\{1, \ldots, \infty\}$	cluster indicator for cell j
$\mathcal{T}$	all trees on $\lvert\boldsymbol{c}\rvert$ leaves	Clonal Phylogenetic Tree
$\mathcal{M}_\lambda (= \{\lambda_r, \lambda_l\})$	$[0, 1]$	Parameters of the model of evolution
G_{ki}	$\{0, \ldots, \lvert g_t\rvert\}$	True genotype of clone k for genomic locus i
D_{ij}	$\{0, \ldots, \lvert g_o\rvert\}$	Observed genotype of i^{th} SNV from single cell j. Observed as the input data inferred from a variant caller.
α	$[0, 1]$	False-positive error rate
β	$[0, 1]$	False-negative error rate

Supplemental Table S3: Hyper-parameters used in SiCloneFit model (Supplemental Fig. S1).

Hyper-parameter	Description
a, b	Hyper-parameters for Prior distribution of α_0
a_α, b_α	Hyper-parameters for Prior distribution of α
a_β, b_β	Hyper-parameters for Prior distribution of β
a_M, b_M	Hyper-parameters for Prior distribution of $\mathcal{M}_\lambda$

Supplemental Table S4: Error model distribution for ternary genotype.

		D_{ij}		
		0	1	2
$G_{c_j i}$	0	$1-\alpha-\frac{\alpha\beta}{2}$	α	$\frac{\alpha\beta}{2}$
	1	$\frac{\beta}{2}$	$1-\beta$	$\frac{\beta}{2}$
	2	0	0	1

Supplemental Table S5: Error model distribution for binary genotype.

		D_{ij}	
		0	1
$G_{c_j i}$	0	$1-\alpha$	α
	1	β	$1-\beta$

Supplemental Table S6: Expected genotype state after combining two genotypes using the binary operator $\oplus$.

$\oplus$	$g=0$	$g=1$	$g=2$
$g=0$	0	1	1
$g=1$	1	1	1
$g=2$	1	1	2

Supplemental Table S7: New variables used in extended SiCloneFit model for handling doublets (Supplemental Fig. S2).

Variable	Range	Description
c_j^1	$\{1,\ldots,\infty\}$	Primary cluster indicator for cell j
c_j^2	$\{1,\ldots,\lvert\boldsymbol{c^1}\rvert\}$	Secondary cluster indicator for cell j.
Y_j	$\{0,1\}$	Bernoulli variable indicating if cell j is a singlet (0) or doublet (1)
δ	$[0,1]$	Doublet rate
a_δ, b_δ		Hyper-parameters for Prior distribution of δ

References

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. *Inf. Retr.*, 12(4):461–486, August 2009.

[2] Tamara Broderick, Jim Pitman, and Michael I. Jordan. Feature Allocations, Probability Functions, and Paintboxes. *Bayesian Anal.*, 8(4):801–836, 12 2013.

[3] George Casella, Christian P. Robert, and Martin T. Wells. *Generalized Accept-Reject sampling schemes*, volume Volume 45 of *Lecture Notes–Monograph Series*, pages 342–347. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2004.

[4] Alexander Davis, Ruli Gao, and Nicholas Navin. Tumor evolution: Linear, branching, neutral or punctuated? *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 1867(2):151 – 161, 2017. Evolutionary principles - heterogeneity in cancer?

[5] Michael D. Escobar and Mike West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[6] Joseph Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.

[7] Arno Fritsch and Katja Ickstadt. Improved criteria for clustering based on the posterior similarity matrix. *Bayesian Anal.*, 4(2):367–391, Jun 2009.

[8] Mel Greaves and Carlo C Maley. Clonal evolution in cancer. *Nature*, 481(7381):306, 2012.

[9] PETER J. GREEN. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[10] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[11] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, Dec 1985.

[12] Katharina Jahn, Jack Kuipers, and Niko Beerenwinkel. Tree inference for single-cell data. *Genome Biology*, 17(1):1–17, 2016.

[13] Marco L. Leung, Alexander Davis, Ruli Gao, Anna Casasent, Yong Wang, Emi Sei, Eduardo Vilar, Dipen Maru, Scott Kopetz, and Nicholas E. Navin. Single-cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer. *Genome Research*, 2017.

[14] Marco L. Leung, Yong Wang, Charissa Kim, Ruli Gao, Jerry Jiang, Emi Sei, and Nicholas E. Navin. Highly multiplexed targeted DNA sequencing from single nuclei. *Nat. Protocols*, 11(2):214–235, Feb 2016. Protocol.

[15] Shaoping Ling, Zheng Hu, Zuyu Yang, Fang Yang, Yawei Li, Pei Lin, Ke Chen, Lili Dong, Lihua Cao, Yong Tao, et al. Extremely high genetic diversity in a single tumor points to prevalence of non-Darwinian cell evolution. *Proceedings of the National Academy of Sciences*, 112(47):E6496–E6505, 2015.

[16] Steven N. Maceachern and Peter Müller. Estimating Mixture of Dirichlet Process Models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998.

[17] Andrew McPherson, Andrew Roth, Emma Laks, Tehmina Masud, Ali Bashashati, Allen W Zhang, Gavin Ha, Justina Biele, Damian Yap, Adrian Wan, et al. Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer. *Nature genetics*, 48(7):758, 2016.

[18] E. W. Meeds, D. A. Ross, R. S. Zemel, and S. T. Roweis. Learning stick-figure models using nonparametric Bayesian priors over trees. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[19] Nicholas Navin. Cancer genomics: one cell at a time. *Genome Biology*, 15(8):452–465, 2014.

[20] Radford M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[21] Jim Pitman. Combinatorial Stochastic Processes. *Ecole d'Eté de Probabilités de Saint-Flour XXXII*, 2006.

[22] Edith M. Ross and Florian Markowetz. OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome Biology*, 17(1):1–14, 2016.

[23] Andrew Roth, Andrew McPherson, Emma Laks, Justina Biele, Damian Yap, Adrian Wan, Maia A. Smith, Cydney B. Nielsen, Jessica N. McAlpine, Samuel Aparicio, Alexandre Bouchard-Cote, and Sohrab P. Shah. Clonal genotype and population structure inference from single-cell tumor sequencing. *Nat Meth*, 13(7):573–576, Jul 2016. Brief Communication.

[24] Raazesh Sainudiin and Amandine Veber. A Beta-splitting model for evolutionary trees. *Royal Society Open Science*, 3, 2016.

[25] Sohrab Salehi, Adi Steif, Andrew Roth, Samuel Aparicio, Alexandre Bouchard-Côté, and Sohrab P. Shah. ddClone: joint statistical inference of clonal populations from single cell and bulk tumour sequencing data. *Genome Biology*, 18(1):44, Mar 2017.

[26] K.P. Schliep. phangorn: phylogenetic analysis in R. *Bioinformatics*, 27(4):592–593, 2011.

[27] Jeet Sukumaran and Mark T. Holder. DendroPy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010.

[28] Marc J Williams, Benjamin Werner, Chris P Barnes, Trevor A Graham, and Andrea Sottoriva. Identification of neutral tumor evolution across cancer types. *Nature genetics*, 48(3):238, 2016.

[29] Hamim Zafar, Anthony Tzen, Nicholas Navin, Ken Chen, and Luay Nakhleh. SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome Biology*, 18(1):178, Sep 2017.