dd-PyClone: Improving Clonal Subpopulation Inference from Single Cells and Bulk Sequencing Data

by

Sohrab Salehi

B.Sc. Computer Engineering, Amirkabir University of Technology, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Bioinformatics)

The University of British Columbia
(Vancouver)

December 2015

Abstract

Improving our understanding of intra-tumour heterogeneity in cancer has important clinical implications, including an opportunity to understand mechanisms behind relapses and drug resistance. Next generation bulk sequencing is a mature technology that has been used to study subclonal tumour populations at an aggregate level. Inference of populations from bulk sequencing requires sophisticated computational deconvolution methods. An alternative is to identify populations directly with single cell sequencing. However, single cell sequencing is a very error-prone process, and this impedes its ability to completely replace bulk sequencing for now.

In this work we present dd-PyClone, a statistical model to combine single cell and bulk sequencing data to study clonal subpopulation architecture and improve clustering assignment and cellular prevalence estimates of a set of genomic loci.

We introduce a single nucleotide variant and copy number aberration aware genotype simulation scheme based on a phylogenetic tree, termed the Generalized Dollo model. This model is an improvement over previous genotype generator models in that it also accounts for the evolutionary process before a rare event (here the single nucleotide variant) occurs.

We show that incorporating genomic loci co-occurrence patterns from single cell sequencing studies in inferring clonal subpopulation structure from bulk sequencing data is beneficial. Our method outperforms existing methods in simulation studies and performs comparably in real dataset benchmarking. We also show that our method is fairly robust as to the choice of hyperparameters and performs reasonably in presence of noise. We hope that our method will further the understanding of the evolutionary basis of cancer.

Preface

The project idea was conceived by Prof. Shah. The application of a distance dependent Chinese restaurant process to solve this problem was my idea. The generalized Dollo model in Chapter 7 is Prof. Bouchard-Côté's idea. The real datasets used in Chapter 7 was used with permission from authors.

The implementation of ddCRP inference algorithm is based on that in [2]. I have extended it to support the non-conjugate case relevant to this work.

There are no publications based on this work so far.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisors, Prof. Alexandre Bouchard-Côté and Prof. Sohrab P. Shah, for their unrelenting support during my studies.

I would also like to thank my committee member Prof. Samuel Aparicio for his contributions to the progress of my thesis, and Prof. Paul Pavlidis for chairing my defence.

A special thanks to Andrea Sollberger, Sharon Ruschkowsk, Jenny Cromarty, and Carolyn Lui for helping me manage my work.

I would like to acknowledge the Canadian Institute of Health Research (CIHR) for funding my studies.

Chapter 1

Introduction

In this chapter, we first introduce cancer as an evolutionary phenomenon. Next generation bulk and single cell sequencing and their associated computational methods are reviewed as the tools for quantifying the properties of this evolutionary system, including SNVs as genetic markers. We continue with a review of existing methods that infer clonal subpopulation structure from bulk sequencing SNV data. We then state our hypothesis in this work as follows: Combining single cell and bulk data in a unified statistical framework improves clustering and cellular prevalence estimates. Our main contribution is a computational method to implement, test, and verify our hypothesis.

1.1 Cancer as an evolutionary phenomenon

Cancer is an evolutionary phenomenon [23]. Cells accrue mutations in time, some of which confer a selective advantage on the bearer. This process leads to multiple cell subpopulations, each with unique genomic aberrations. This intra-tumour heterogeneity results in clinical complications, including relapses and drug resistance [36].

To find an optimal therapeutic intervention we first need to address a fundamental question, ”How does a tumour population respond to a specific perturbation?”, where a perturbation is any treatment policy, including surgery and chemotherapy.

The first step toward understanding this problem is to identify subpopulations or subclones present in a tumour sample as well as their prevalences. Somatic genetic markers, particularly single nucleotide variants (SNVs) are used to identify subclones [29]. Cells in the human body inherit their genomes from their parents. These inherited genomes harbour differences to other individuals that are called germline mutations. On the other hand, both normal and cancerous cells, all descendants of the initial zygote cell, accrue novel mutations not present in their common ancestor. These are somatic mutations [33].

Next generation sequencing has made measurement of these markers fast and cost-efficient [22]. Somatic mutations are essentially inferred by comparing DNA extracted from tumour tissue to the DNA from normal tissue from the same patient. Here we focus on next generation sequencing methods to measure SNVs.

1.2 Next generation sequencing

Briefly, first DNA molecules from millions of cells from a tissue sample are collected. This involves cell lysis through which we lose the assignment of genomes to cells, and consequently the mutation co-occurrence patterns. In a library preparation step, DNA molecules are fragmented and then are ligated by adapter sequences. This is often proceeded by an amplification phase where fragmented DNA sequences (templates), are amplified in a process called polymerase chain reaction (PCR). The amplified templates are sequenced in a sequencer and then analyzed to obtain their nucleic acid sequence in what is called the base calling step [15].

Then, base-called reads are mapped to a reference genome. At this point, several post-processing steps may be required. For example, the identification and removal of adapter sequences, reads of low quality, and contamination from foreign DNA [16].

1.2.1 Somatic single nucleotide variants

Methods such as [4, 7, 28, 30] are used to detect somatic SNVs. This is a challenging task, since many variants are either germline mutations or some artifact of the sequencing process. This is further complicated by the fact that in some cancers germline variations outnumber somatic variations by orders of magnitude. Sequencing artifacts are also prevalent and come from a number of different sources, including normal DNA contamination, strand bias, and low mapping quality.

Somatic variant calling methods work by essentially modelling the allelic distribution, taking into account various sources of error. They may involve preprocessing steps, or filters, such as removal of low quality reads and local realignments. Some post processing may also be applied to acquire higher confidence candidate variants, for instance, filtering the variants against a database of normal patients [4].

1.2.2 Targeted deep sequencing

A technique used to validate these mutation calls is targeted deep sequencing [31]. To ensure that the called SNVs are not an artifact of sequencing error, limited sections of the genome that contain selected SNVs are chosen and highly amplified and sequenced. This very high read coverage (number of reads covering each genomic locus) increases confidence in the validity of the called SNVs. This data is then used to generate allele counts. That is, at each genomic locus, how many reads map to the reference allele and how many reads map to the alternative allele.

1.2.3 Copy number aberrations

Another important genomic feature of cancer cells is their copy number state. Amplification and Loss of Heterozygosity (LOH) events have been shown to play a role in tumour progression [32].

NGS data can be used to identify Copy Number Aberration (CNA) events [24]. Computational methods based on NGS data mostly work by dividing the genome into different segments (bins) and use mean change in read coverage in each segment to assign an average copy number to the whole region. In tumour samples that contain multiple heterogeneous subpopulations, traditional CNA calling methods may fail to assign an integer copy number to a region. This is due to each subpopulation having potentially a different copy number state at the same region.

Methods such as [14] take tumour cellularity and different CN states for each subpopulation genotype into account, and they assign the most probable copy number state to each region based on a Hidden Markov Model.

1.2.4 Measuring clonal parameters

In NGS data we can only directly measure allele counts. Due to fragmentation and short read lengths, patterns of co-occurrence (phasing) are lost and direct observation of number of subpopulations, their genotypes, and prevalences (collectively called subclonal structure) is not possible.

To address this problem, computational methods have been developed to estimate these quantities or their surrogates. Since in general there is no bijection between allelic ratios and subclonal structures, it is often easier to estimate surrogate quantities for subclonal structure parameters.

Mutational cellular prevalence is a compound measure for genotype prevalence. It is defined as the fraction of cells that harbour a mutation at a specific genomic locus.

To infer cellular prevalence, methods such as PyClone [29] correct allele counts for CNVs and LOH events, and tumour cellularity to cluster genomic loci into groups with similar cellular prevalences. One of the main hypothesis in these methods is that there are no subclones with identical prevalences in the tumour, and thus a difference in cellular prevalence is due to belonging to different subclones. If this assumption is violated in the data, such subclones would get merged into a single group in the bulk deconvolution (over-clustering).

These mutation clusters could be considered as surrogate measures for subclonal genotypes, where we would expect that two mutations in the same cluster to co-occur in a genotype. In section 1.3 we review in more detail methods that infer subclonal structure from next generation sequencing SNV data.

1.3 Existing methods

Here we briefly review some of the existing methods that infer sub-clonal population composition from bulk next generation sequencing data.

1.3.1 Clomial

Clomial [37] accepts reference and variant allele counts from multiple subsections of a tumour as input. Number of genotypes should be set a priori. From this, it estimates both a genotype matrix, which indicates which genomic loci are mutated in each genotype, and a genotype prevalence matrix that shows what fraction of cells in a subsample belong to a particular genotype. The estimation problem is formulated as a variation of the Matrix Factorization problem where the allele counts matrix is factored into genotypes and genotype frequencies matrices. Inference is done using an Expectation Maximization algorithm.

It infers tumour cellularity, the fraction of cancerous cells in the tumour sample, from the data. Tumour cellularity is a measure of normal DNA contamination in the sample.

On the other hand, it assumes a diploid genotype and does not correct allele counts for copy number variations, nor does it take sequencing errors into account in its Binomial likelihood model. It is limited to situations where the number of clones are less than or equal to the number of samples (subsections of the tumour), since otherwise, the inference problem is under-constrained.

A major problem with Clomial is that it needs the number of genotypes to be set a priori. To mitigate this, Clomial poses the choice of the number of genotypes C as a model selection problem and suggests running the method with different values of C, selecting the one with the best Bayesian Information Criterion (BIC). Furthermore, to increase the chance of finding the global extremum, the authors suggest multiple restarts from different starting positions.

1.3.2 PyClone

The PyClone model has inspired our current work. As input, it accepts a fixed set of genomic loci with their allele counts and copy number states, as well as tumour cellularity. It uses a Dirichlet process mixture model [34] to jointly infer cellular prevalences and cluster assignments. PyClone assumes that at each genomic locus, the tumour has 3 subpopulations, namely, normal, reference, and variant subpopulations. Each subpopulation has a fixed copy number state with respect to a particular genomic locus.

It accounts for copy number states by defining a prior probability distribution over possible genotypes in each subpopulation at each genomic locus, and summing over these genotypes in the likelihood calculation. It uses a Beta-Binomial likelihood (emission) distribution to account for overdispersion in the sequencing data. PyClone's inference is based on a Markov chain Monte Carlo (MCMC) sampling scheme.

It also supports a multi-sample mode where it accepts as input sample specific tumour cellularity, copy number state, and allele count data. In multisample mode, PyClone will output one clustering result for mutations over all samples, and a per sample cellular prevalence estimate.

1.3.3 PhyloSub and PhyloWGS

PhyloSub [17] tries to infer cellular prevalences and cluster assignments of SNVs as well as their phylogenetic history. Its inputs are allele counts of a fixed set of SNVs, as well as associated copy number state estimates. It outputs clusters of SNVs and their cellular prevalences, in addition to a set of most likely phylogenetic trees. It achieves this by using a tree-structured stick breaking process [12]. This is a Bayesian non-parametric prior over partitions that has a hierarchical tree topology, where data points can be assigned to any node on the tree.

Heuristic rules are used to limit the space of possible phylogenetic trees. These rules come from a perfect and persistence phylogeny assumption [29]. That is, a mutation is only gained once over a tree, and when its gained, it cannot be lost or reverted.

In PhyloSub, phylogenetic trees are directed acyclic graphs (DAG) where nodes are SNV clusters and edges connect parental clusters to their immediate child clusters. PhyloWGS [6] is a successor to PhyloSub. It improves on PhyloSub in two respects. First, it handles CNAs differently by modelling them as a pseudo-SNV and thus inferring at what point on the phylogenetic tree they have occurred. Second, the authors claim that it is much faster than PhyloSub and therefore could be used to infer tumour subpopulation structure form whole genome sequencing data. Both models use a Binomial likelihood model and a MCMC algorithm for inference.

1.3.4 SciClone

SciClone's [18] input consists in variant allele frequencies (VAF) as well as copy number variations. It operates over genomic loci in copy neutral regions. The maximum number of clusters is another input of this model.

SciClone clusters genomic loci with similar VAFs. Similar to PyClone, it outputs cluster assignments for genomic loci and frequencies for mutational groups.

It models VAFs as a mixture of Beta distributions. Its inference framework is based on the variational Bayesian mixture modelling. To approximate the posterior distribution, it uses Gamma distributions with a Dirichlet distribution for mixture parameters.

It does not take into account copy number variations, nor does it correct for tumour cellularity. Variational Bayes methods will generally never converge to the true posterior (it could not even represent it).

1.4 Single cell sequencing

In general, bulk methods are limited. The fact that association of genotypes to cells is lost at cell lysis results in complications such as the phasing and over-clustering issues mentioned earlier. A potential solution to these problems is sequencing the genome of individual cells. Single cell sequencing (SCS) involves isolating a cell, amplifying its genome followed by base calling, and mapping to a reference genome [20].

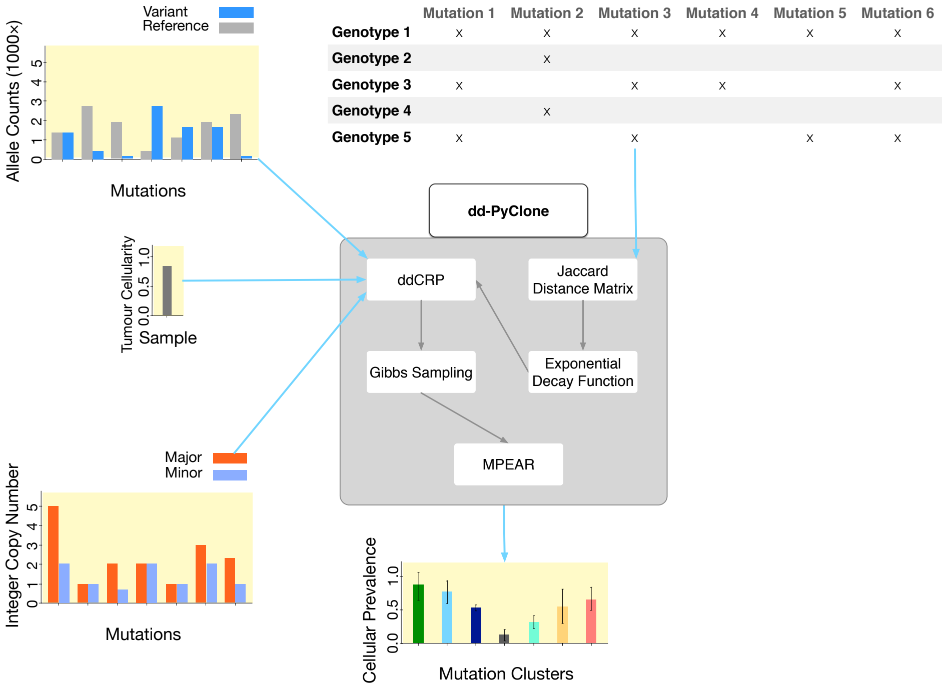
SCS is still in its infancy and suffers from a number of problems [20, 35]. First, it is very error-prone. Sources of error include allele dropout and uneven genome coverage that may result from a biased amplification step. Second, it is prohibitively expensive. Third is the issue of undersampling.

In SCS studies, single cells analyzed number in the lower hundreds [8, 19]. The probability of observing a representative cell from a very rare clone (cellular prevalence of less than 0.01%) is only about 0.01. Concretely, considering the above numbers and assuming a uniform sampling procedure and that the clones are uniformly distributed throughout the tumour, the power of the SCS experiment to test the existence of a very rare clone would be:

$\text{p(observing at least a single cell from a very rare clone}|\text{that clone exists in the tumour)} = 1 - \text{p(not observing any cells from the very rare clone}|\text{that clone exists)} = 1 - (1 - 0.0001)^{100} \approx 0.01$.

1.5 Main hypothesis, combining bulk data and SCS data

Our main hypothesis is ”Combining single cell and bulk data in a unified statistical framework improves clustering and cellular prevalence estimates.” Figure 1.1 shows the main components and workflow of our method. This work has the potential to overcome the limitations of both bulk and single cell methods. This is made more precise in chapter 2. We will experimentally validate our hypothesis in chapter 3 and demonstrate that if the majority of genotypes present in the tumour are observed, using them to guide clustering in bulk data will result in improved estimates.



	Mutation 1	Mutation 2	Mutation 3	Mutation 4	Mutation 5	Mutation 6
Genotype 1	x	x	x	x	x	x
Genotype 2		x				
Genotype 3	x		x	x		x
Genotype 4		x				
Genotype 5	x		x		x	x

Figure 1.1: This figure shows the workflow of our method, dd-PyClone. Inputs are genotypes from single cell genotyping experiment, tumour cellularity, and allele count estimates. As output, we infer clusters of genomic loci (mutation clusters) and their cellular prevalences. The details of the model, including definitions of ddCRP, distance matrix calculation, clustering, and inference routines are given in Chapter 2.

1.6 Summary

In this chapter we introduced cancer as an evolutionary system. We discussed strengths and limitations of next generation and single cell sequencing technologies and associated computational methods in quantifying intra tumour heterogeneity. We stated combination of bulk NGS and SCS as our objective in this work.

The rest of this document is organized as follows: Chapter 2 describes the assumptions of the model, its mathematical formulation, and the computational model to solve it. Chapter 3 introduces a simulator to generate SNV and CNA aware genotypes and reports our simulation studies as well as experiments on a real primary triple negative breast cancer dataset. Finally, Chapter 4 discusses a summary of the work as well as potential future research directions.

Chapter 2

Methods

In this chapter we formulate our hypothesis and establish relevant notations. We then introduce our modelling procedure in the context of a simplified problem. We then describe our main modelling method, the ddCRP that is a generalization of the traditional CRP. Using ddCRP, we construct an informed prior over partitions that encourage co-occurring genomic loci to cluster together. We proceed with describing our full model, dd-PyClone. We then introduce our inference procedure, a MCMC sampling scheme that uses Gibbs moves to update clustering and parameter assignments. Finally, we discuss limitations of our model and possible future extensions.

2.1 Problem statement

Here, we first establish relevant terms, notations, inputs, and outputs of our model. Then we formulate our problem in a mixture modelling framework.

2.1.1 Main hypothesis

We reiterate our main hypothesis here: Combining single cell and bulk data in a unified statistical framework improves clustering and cellular prevalence estimates.

2.1.2 Concepts and definitions

Given (i) variant allele counts and (ii) copy number state at each genomic locus, (iii) tumour cellularity, and (iv) single cell genotype data, our method infers (i) cellular prevalences and (ii) cluster assignments for those genomic loci. We define these terms below.

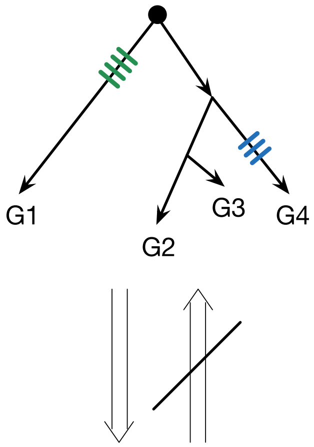
Inputs

Variant allele counts is the first input to the model. We assume that at each genomic locus i, a total of d_i reads map to the variant and normal alleles out of which b_i reads map to the variant allele. A related notion is the **variant allelic prevalence**, ξ, that is the expected fraction of reads that harbour the variant allele. This is computed as the number of mutated alleles in the variant subpopulations divided by the total number of alleles in all cells.

The second input to the model is the copy number state at each genomic locus. Note that copy number variations affect ξ. For instance in Figure 2.4, $\xi = \frac{2\times5}{2\times1+3\times3+3\times5} = \frac{5}{13}$.

The third input to the model is the **tumour cellularity** t, as the fraction of cancer cells in the sample. Hence the fraction of normal cells would be $1-t$. We assume it is estimated independently from our model.

The last input to our model is the genotype data. Let M denote the number of genotypes in the tumour sample and N be the number of genomic loci in our model. Genotype data is modelled as a binary matrix $\Delta \in \{0,1\}^{M\times N}$ with rows corresponding to genotypes and columns to genomic loci. Each entry $\Delta_{m,n}$ is equal to one if the genotype m is mutated at locus n. We assume in this work that genotype data is derived from single cell sequencing studies. Figure 2.1 illustrates our assumption about the relation between the genomic loci co-occurrence patterns and the underlying phylogenetic tree.



	G1	G4
Mut 1	x	
Mut 2	x	
Mut 3	x	
Mut 4	x	
Mut 5		x
Mut 4		x
Mut 7		x

Figure 2.1: A hypothetical phylogenetic tree with genotypes at leaves (top). The green and blue bars on the tree denote mutations that have happened together. A subset of the corresponding mutation co-occurrence patterns (bottom). Note that the bottom matrix shows a transposed version of the genotype matrix. While it always holds that if mutations are gained at the same site on the phylogenetic tree, then they will co-appear in the genotype matrix (the top-to-bottom arrow), the opposite is not always true (the bottom-to-top arrow). We are making the simplifying assumption that if mutations co-occur in a genotype matrix, then they have co-occurred in the underlying phylogenetic tree as well.

Outputs

The desired outputs are **cluster assignments** of genomic loci and their **cellular prevalences**. Cellular prevalence ϕ_i for a particular genomic locus i is defined as the fraction of cells in the sample that harbour a mutation at that genomic locus. For example, in Figure 2.4 cellular prevalence for the depicted genomic locus is $\frac{5}{8}$. Thus $1-\phi_i$, the fraction of cancer cells from the reference population, is $1-\frac{5}{8}=\frac{3}{8}$. We define the clonal prevalence of a genotype to be the fraction of cells in the tumour sample that match that genotype.

2.1.3 Notations

Let $X=\{x_1,x_2,...,x_N\}$ be the set of our N genomic loci, indexed by $\varpi=\{1,2,...,N\}$.

We adopt the notation $j:i$ for $j\leq i, j,i\in\mathbb{N}$ to denote $\{j,j+1,j+2,...,i\}$, a subset of successive integers.

We define a clustering of X as a partition T of its index set ϖ, that is $T=\{T_1,T_2,...,T_K\}$ such that $\sqcup_{k\in 1:K}T_k=\varpi$ where K is the number of partitions and $\sqcup$ denotes the disjoint union operator and each subset T_k is called a cluster.

We define x_A for $A\subset\varpi$ to be $\{x_i|i\in A\}$. For example x_{T_k} is the set of data points in cluster T_k and $x_{i:j}=\{x_i,x_{i+1},x_{i+2},...,x_j\}$.

Furthermore, let $T(.)\colon\mathbb{N}\to\mathbb{N}$ map data point indices to their clusters, that is $T(i)=k$ iff $i\in T_k$.

Partitions in a graph

Let $\mathbb{G}(\mathscr{V},\mathbb{E})$ denote an undirected graph $\mathbb{G}$ where $\mathscr{V}$ is the set of vertices and $\mathbb{E}$ is the set edges, i.e., a set of unordered pairs $\{u,v\}\subset\mathscr{V}$.

The set of edges $\mathbb{E}$ induces a partitioning on $\mathscr{V}$, where each connected component of $\mathscr{V}$ corresponds to a cluster. With a slight abuse of notation, let $T(\mathbb{E})=T(\mathbb{G}(\mathscr{V},\mathbb{E}))$ denote this partitioning and $T_{\mathbb{E}}^{k}$ denote its k-th cluster.

A directed graph $\mathscr{G}(\mathscr{V},\mathscr{E})$ consists in a set of vertices $\mathscr{V}$ and a set of directed edges $\mathscr{E}$ where each edge is an ordered pair of vertices.

For a directed graph $\mathscr{G}$, we define its underlying undirected graph $U(\mathscr{G})$ to be the graph obtained by replacing all directed edges in $\mathscr{G}$ with undirected ones.

Let $T(\mathscr{E})$ be the partitioning induced by $U(\mathscr{G})$, the underlying undirected graph of $\mathscr{G}$. Throughout this document the $\mathscr{G}$ corresponding to $\mathscr{E}$ is always apparent from the context, with $\mathscr{V}$ always being the set of our data points.

Let $T_{\mathscr{E}} : \mathbb{N} \rightarrow \mathbb{N}$ map vertex indices to their clusters, that is $T_{\mathscr{E}}(i) = k$ iff $i \in T_{\mathscr{E}}^{k}$.

2.1.4 Simplified generative model

For exposition, we start with a simplified generative model in which we describe the relationship between inputs and the outputs of our method. Assume we have a heterogeneous tumour that contains subpopulations from two distinct haploid genotypes, g_1 and g_2 with clonal prevalences of 30% and 70%, respectively. For simplicity we set the tumour cellularity in our sample to one ($t = 1$). Since this implies that the expected fraction of variant allele reads is equal to cellular prevalence at each genomic locus ($\xi = \phi$), we will ignore ξ and directly use ϕ in this subsection.

The possible cellular prevalences for any genomic locus i in this tumour are $\phi_i = \{\phi_i^1 = 0.0, \phi_i^2 = 0.3, \phi_i^3 = 0.7, \phi_i^4 = 1.0\}$. Since locus i is either not mutated in any of the genotypes (hence ϕ_1), only mutated in g_1 (corresponding to ϕ_2), only mutated in g_2 (therefore ϕ_3), or mutated in both g_1 and g_2 (meaning ϕ_4). These four cases represent our possible clusters.

To simulate the sequencing process for genomic locus i, we first pick its cluster. We use an auxiliary variable z_i as follows: $z_i \sim \text{Categorical}(w)$ where $w = w_{1:4}$ denotes the mixing weights, the proportion of clusters such that $\sum_{i=1}^{4} w_i = 1$.

The cellular prevalence for genomic locus i is now ϕ_{z_i}. **In the inference procedure, z_is and ϕ_{z_i}s constitute our desired outputs.** Next we simulate the number of variant alleles. Since according to our assumptions ϕ_{z_i} also denotes the expected proportion of variant reads in the sequencing experiment, we can relate it to the variant read counts b_i via a Binomial likelihood function as follows: $b_i \sim \text{Binom}(d_i, \phi_{z_i})$ where for now, we fix d_i, the total number of reads, to some appropriate constant value [1]. In the inference procedure, we observe b_i and d_i and

[1] It could vary from about 10× in a whole genomic sequencing to about 10,000× in an ultra deep sequencing experiment [31].

they are the inputs to our model. Put together, we have:

$$\begin{aligned} z_i &\sim \text{Categorical}(w) \\ b_i &\sim \text{Binom}(d_i, \phi_{z_i}) \end{aligned} \tag{2.1}$$

The two step process we described in equation 2.1 defines a mixture distribution as follows:

$$p(b_i) = \sum_{j=1}^{4} w_j \text{Binom}(d_i, \phi_j) \tag{2.2}$$

Here we have four possible mixture components or clusters. Cluster assignments for each datapoint (a genomic locus in our model), are determined by the indicator variables z_i-s that are sampled from a categorical distribution, our prior over partitions, since we assumed that (i) the possible values for the ϕ_i-s were finite and (ii) known. Neither of these assumptions hold in general, that is, the ϕ_i-s could be any real-valued number in $[0,1]$. To address this issue in a principled way, we introduce the Chinese Restaurant Process (CRP) in subsection 2.2.2.

Furthermore, co-occurrence patterns in g_1 and g_2 could be used to construct an informed prior over partitions of genomic loci. This can be done via a generalization of CRP, called ddCRP, that we introduce in subsection 2.2.1. Before describing our model, dd-PyClone, in section 2.3, we present an updated generative process for the simplified example that we considered here in subsection 2.2.3.

2.2 Distance dependent Chinese Restaurant Process

This section introduces the distance dependent Chinese Restaurant Process (ddCRP), the method we use to incorporate single cell genotyping data to construct an informed prior over partitions that encourages co-occurring genomic loci to cluster together. We begin by presenting the traditional CRP, a special case of ddCRP that is agnostic to the co-occurrence pattern of genomic loci.

2.2.1 Traditional CRP

ddCRP can be explained through an alternative representation of the Chinese Restaurant Process (CRP). We follow the notation in [2]. In the traditional CRP, customers enter a Chinese restaurant and opt to sit at a table where the probability of joining a table is proportional to the number of customers already sitting at that table.

Customers may also choose to sit at a new table with probability proportional to α, a model parameter. **In the Chinese restaurant metaphor, customers represent the genomic loci and tables represent clusters.**

Let z_i denote the table assignment for customer i and assume that customers $1 : i-1$ have occupied tables $1 : K$ and let n_k be the number of customers sitting at table k. Customer sitting configuration induces a partitioning of customer indices. CRP draws z_i as in equation 2.3.

$$p(z_i = k | z_{1:(i-1)}, \alpha) \propto \begin{cases} n_k & \text{for } k \leq K \\ \alpha & \text{for } k = K+1 \end{cases} \tag{2.3}$$

The CRP is an exchangeable process, that is, the order in which customers enter the restaurant does not affect the probability of a certain partition [34].

2.2.2 Alternative representation of Traditional CRP

Traditional CRP can equivalently be viewed as customers joining other customers instead of joining other tables. Let c_i denote the customer index with whom customer i is sitting and $C = c_{1:N}$. This defines a directed graph $\mathscr{G}(\mathscr{V}, \mathscr{E})$ with $\mathscr{V}$ the set of customer indices and $\mathscr{E}$ the set of ordered pairs (i, c_i).

As described in subsection 2.1.3, this induces $T_{\mathscr{E}} = T(C)$ a partitioning of customer indices. Each cluster corresponds to a table in the traditional representation.

Figure 2.2 shows an example C and its corresponding $T(C)$.

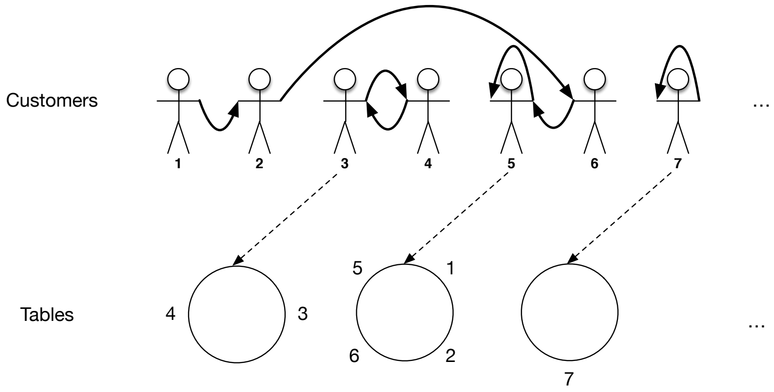


Figure 2.2: Induced table sitting $T(C)$ by a particular customer connection configuration C . Bold arrows show customer connections and dotted arrows point to equivalent table sittings. Customer 7 has a self loop and since she is not connected to any other customers, the corresponding table has only one customer.

In a generalization of this model, the probability for a customer i to connect to a customer j is proportional to a function of the distance between them. The distance matrix D encodes our knowledge about the data point's dissimilarity from a secondary source. In this work, this distance matrix is computed from the genotypes derived from single cell genotyping experiments (more details in subsection 2.3.1). The non-increasing decay function f takes non-negative finite values. This is summarized in equation 2.4.

$$p(c_i = j|D, \alpha) \propto \begin{cases} f(d_{i,j}) & \text{for } i \neq j \\ \alpha & \text{for } i = j \end{cases} \tag{2.4}$$

This defines the ddCRP model. We note that picking a constant decay function $f(x) = 1$ reduces ddCRP to traditional CRP, since in that case, equation 2.4 is identical to equation 2.3.

Unlike traditional CRP, ddCRP is not an exchangeable process. This means that the order in which customers enter the restaurant changes the probability of a particular partition. In our implementation, we randomly shuffle the order of customers at each iteration of the sampling algorithm. To investigate the effects of non-exchangeability, we ran our method over synthetic and real datasets with and without random reordering of customers in chapter 3, subsection 3.4.4.

We found that the method performs nearly identically with or without random customer reordering. This may imply that our method is not very sensitive to the order of customers.

2.2.3 Generative process for ddCRP mixture modelling

Now that we have seen how ddCRP can be used to construct an informed prior over partitions, we present the high level forward simulation algorithm for mixture modelling in ddCRP for the simplified example we considered in subsection 2.1.4:

1. For $i \in [1,N]$, draw $c_i \sim$ ddCRP(α, f, D).
   From this, derive $T(C)$, the corresponding table assignment.

2. For $i \in [1,K]$, draw $\phi_i \sim G_0$.

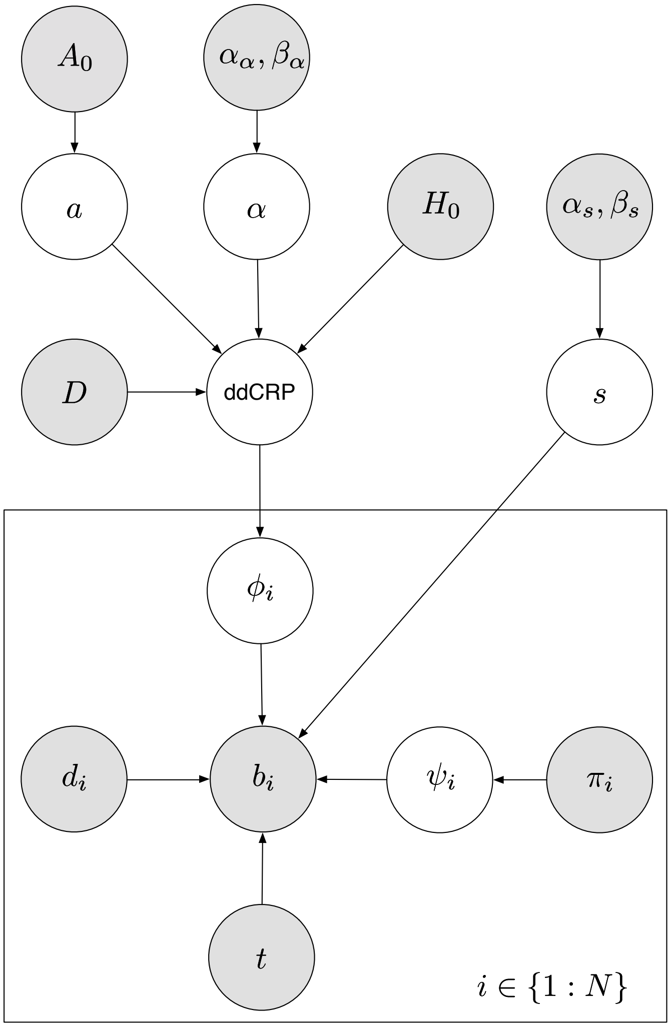
3. For $i \in [1,N]$, draw $b_i \sim F_i(\phi_{T_C(i)})$.

where α is a model parameter, f is a decay function, G_0 is the base distribution for the ϕ_i-s, F_i is the likelihood function relating expected number of reads to b_i to cellular prevalence ϕ_i as in equation 2.1, and $T_C(i)$ is the index of the table at which customer i is sitting.

Formally, in our simplified model, for each genomic locus $i \in [1,N]$, we want to infer ϕ_i and $T_C(i)$, given the model observations b_i and d_i. In section 2.3 we report a complete set of expected model inputs and outputs.

2.3 The dd-PyClone model

Here we introduce our model, dd-PyClone. Figure 2.3 summarizes dependency and distributional assumptions in dd-PyClone's model. Table 2.1 explains random variables used in this model.



(a) Probabilistic Graphical Model (PGM) of dd-PyClone

$$\alpha \sim \text{Gamma}(a_\alpha, b_\alpha)$$

$$H_0 = \text{Uniform}([0,1])$$

$$A_0 = \text{Uniform}([0,1])$$

$$a \sim A_0$$

$$D = \{d_{i,j}\}, d_{i,j} = \text{JaccardDist}(i,j), i,j \in \{1:N\}$$

$$f_a = \exp(-d_{i,j}/a)$$

$$\phi_i | f_a, D, H_0, \alpha \sim \text{ddCRP}(f_a, D, H_0, \alpha)$$

$$\psi_i | \pi_i \sim \text{Categorical}(\pi_i)$$

$$s | a_s, b_s \sim \text{Gamma}(a_s, b_s)$$

$$b_i | d_i, \psi_i, \phi_i, t, s \sim \text{BetaBinomial}(d_i, \xi(\psi_i, \phi_i, t), s)$$

$$\xi(\psi, \phi, t) = \frac{(1-t)\zeta(g_N)}{Z}\mu(g_N) + \frac{t(1-\phi)\zeta(g_R)}{Z}\mu(g_R) + \frac{t\phi\zeta(g_V)}{Z}\mu(g_V)$$

$$Z = (1-t)\zeta(g_N) + t(1-\phi)\zeta(g_R) + t\phi\zeta(g_V)$$

(b) Distributional assumptions of dd-PyClone

Figure 2.3: The complete dd-PyClone model. In the graphical model, the shaded nodes are observed and the rest of the nodes are not observed. In the inference step, the unobserved nodes will be inferred via Gibbs sampling. In particular, we are interested in inferring ϕ_i-s, the cellular prevalences for genomic loci and the induced clustering by the ddCRP.

Table 2.1: Notation reference for dd-PyClone's probabilistic graphical [Notation reference for dd-PyClone] model.

Variable	Description	Observed
A_0	Prior distribution over decay function's parameter a.	Yes
α_α	Shape hyperparameter over ddCRP distribution's α parameter.	Yes
β_α	Rate hyperparameter over ddCRP distribution's α parameter.	Yes
a	Decay function's parameter.	No
α	Model parameter for the ddCRP model.	No
H_0	Base measure for the ddCRP used to sample cellular prevalences for genomic loci.	Yes
α_s	Shape hyperparameter for the Beta-Bionomial precision parameter s.	Yes
β_s	Rate hyperparameter for the Beta-Bionomial precision parameter s.	Yes
D	Distance matrix over genomic loci. In this work, this is computed from single cell genotype analysis.	Yes
ddCRP	The distance dependent Chinese restaurant process with decay function f, distance matrix D, base measure H_0, and model parameter α.	No
s	Precision parameter for the Beta-Binomial emission model.	No
ϕ_i	Cellular prevalence for the genomic locus i.	No
d_i	Total number of reads that map to genome locus i.	Yes
b_i	Number of reads that map to variant allele at genomic locus i.	Yes
ψ_i	A vector (g_N^i, g_R^i, g_V^i) denoting genotype state at genomic locus i.	No
π_i	Prior over the genotype state for the genomic locus i.	Yes
t	Tumour cellularity.	Yes
N	Number of genomic loci.	Yes
M	Number of genotypes.	Yes

We assign each genomic locus to a customer. Throughout this document, we use genotype data from single cell genotyping studies to compute the distance between genomic loci. We note that this is not a requirement of the model, and other sources could be used to define dissimilarity between genomic loci.

2.3.1 Distance matrix

We have used the Jaccard distance to form the distance matrix $D \in [0,1]^{N\times N}$ between genomic loci. Jaccard distance is computed as $1 - \text{JaccardIndex}$ that is:

$$\begin{aligned} \text{JaccardDist}(A,B) &= 1 - \frac{|A \cap B|}{|A \cup B|} \\ |A \cap B| &= \sum_{i=1}^{M} (A_i \times B_i) \\ |A \cup B| &= \sum_{i=1}^{M} (A_i + B_i) \end{aligned} \tag{2.5}$$

where $A_{M\times 1}$ and $B_{M\times 1}$ are binary column vectors, each representing a genomic locus. Intuitively, this assigns a higher distance to genomic loci that co-occur less often in the single cell genotypes and vice versa. We note that our use of the Jaccard index to compute distances between genomic loci is related to the distance-based phylogenetic inference methods [9].

Let $\lambda = \{s, \alpha, a\}$ be the collection of hyperparameters in our model. For brevity, we first assume that these hyperparameters are fixed, and later we discuss their resampling scheme.

2.3.2 Bulk population assumptions

Similar to PyClone, we make the simplifying assumption that the clonal population in the bulk data comprises three subpopulations, namely, the normal, the reference, and the variant subpopulations. Figure 2.4 illustrates this assumption. To avoid confusion with the genotype states coming from the single cell sequencing study, we refer to the assumed copy number state of the subpopulations in the bulk data as **pseudo-genotypes**. This data is usually not available directly from the bulk data, and has to be inferred or accounted for in the inference procedure.

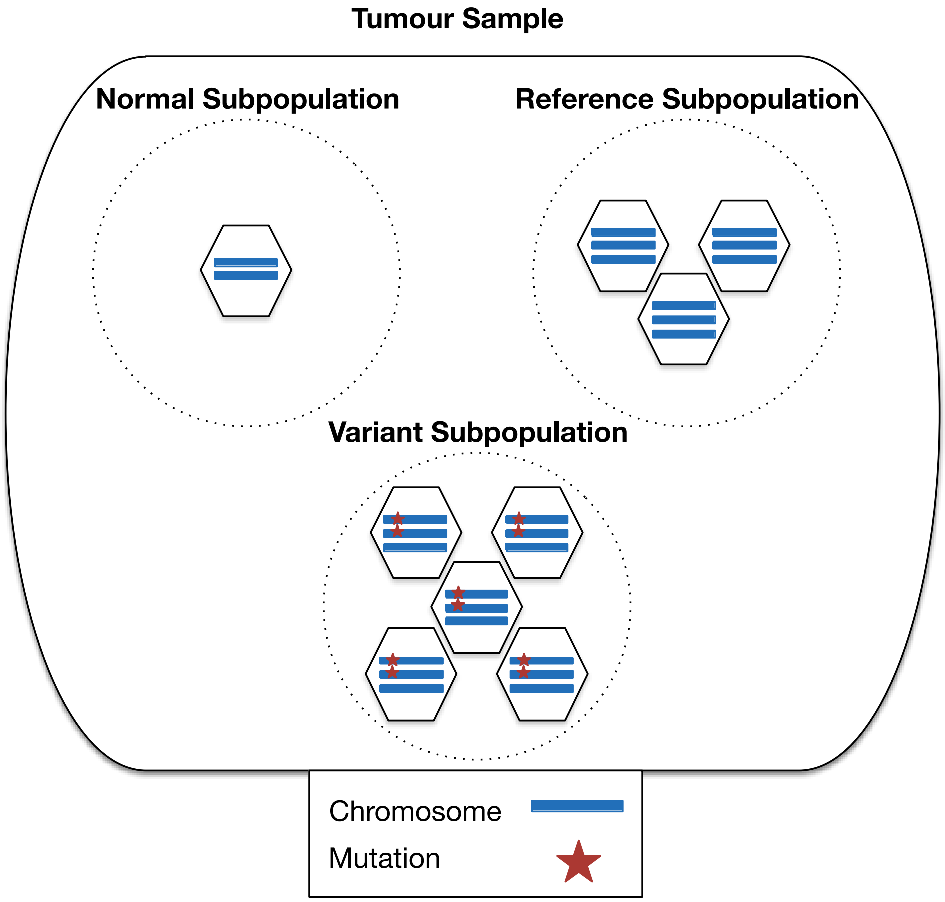


Figure 2.4: Our assumption about clonal architecture in the tumour, with respect to a particular genomic locus. In this example, normal subpopulation represents a collection of un-mutated diploid cells. Reference subpopulation comprises cells that have a copy number amplification event, but no single nucleotide mutations. Variant subpopulation is a collection of cells that have a SNV at the particular genomic locus.

2.3.3 Pseudo-genotype state priors

Let $\psi_i = (g^i_N, g^i_R, g^i_V) \in (\mathbb{N}_0 \times \mathbb{N}_0)^3$ represent the assumed **pseudo-genotype state** at each genomic locus i in the bulk data where $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Let g^i_N represent the normal pseudo-genotype N, g^i_R represent the reference pseudo-genotype R, and g^i_V

represent the variant pseudo-genotype V. Each g^i_S is a pair of non-negative integers that denote the copy number state for the pseudo-genotype $S \in \{N, R, V\}$ at the genomic locus i. For example, $g^i_N = (2,3)$ means that the normal pseudo-genotype in the bulk tumour sample has two copies of the reference allele and three copies of the variant allele at genomic locus i. Here $(0,0)$ denotes a homozygous deletion.

For $g \in \mathscr{G} = \mathbb{N}_0 \times \mathbb{N}_0$, let $\zeta : \mathscr{G} \to \mathbb{N}_0$ be the total copy number of pseudo-genotype g, We define $\mu(g)$, the probability of sampling a variant allele from a subpopulation with pseudo-genotype g as follows:

$$\mu(g) = \begin{cases} \varepsilon & \text{for } b(g) = 0 \\ 1 - \varepsilon & \text{for } b(g) = \zeta(g) \\ \frac{b(g)}{\zeta(g)} & \text{otherwise} \end{cases} \tag{2.6}$$

where ε is the sequencing error probability, the probability of observing a variant allele when sequencing a true reference allele.

We define the function $\xi(\psi, \phi, t)$ to capture the effects of pseudo-genotypes, cellular prevalence, and tumour cellularity as follows:

$$\xi(\psi, \phi, t) = \frac{(1-t)\zeta(g_N)}{Z}\mu(g_N) + \frac{t(1-\phi)\zeta(g_R)}{Z}\mu(g_R) + \frac{t\phi\zeta(g_V)}{Z}\mu(g_V) \tag{2.7}$$

where $Z = (1-t)\zeta(g_N) + t(1-\phi)\zeta(g_R) + t\phi\zeta(g_V)$ is the normalizing constant.

To compute the likelihood, we sum over possible values of ψ_i. Since the discrete space of Ψ values quickly becomes intractable, we only consider a limited number of pseudo-genotypes. This could be done via defining a prior π_i over ψ_i.

2.3.4 The Parental Copy Number (PCN) prior

Following [29], when copy number variation data in form of major and minor copy numbers is available, we have implemented a number of methods to elicit priors over pseudo-genotypes. We assume that copy number state at each genomic locus is reported as a pair of integers $(\bar{\zeta}_1, \bar{\zeta}_2)$, where the major copy number $\bar{\zeta}_1$, refers to the maximum of the two said integers and the minor copy number $\bar{\zeta}_2$ refers to the minimum of the pair and $\bar{\zeta} = \bar{\zeta}_1 + \bar{\zeta}_2$ is the total copy number. Here we describe the Parental Copy Number (PCN) strategy that is used in our experiments.

Let $\mathscr{P}$ denote the set of pseudo-genotype states that PCN scheme describes. We assign equal weight to the pseudo-genotype states in $\mathscr{P}$ and zero weight to any other pseudo-genotype state that is not a member of $\mathscr{P}$, that is a pseudo-genotype state $\psi_i \in \mathscr{P}$ has a weight equal to $\frac{1}{|\mathscr{P}|}$. The pseudo-genotype states with non-zero weights are $\mathscr{P} = \{\psi_1, \psi_2, \psi_3, \psi_4\}$ where

- $\psi_1 = (g_N = (2,0), g_R = (2,0), g_V = (\bar{\zeta}_1, \bar{\zeta}_2))$
- $\psi_2 = (g_N = (2,0), g_R = (2,0), g_V = (\bar{\zeta}_2, \bar{\zeta}_1))$
- $\psi_3 = (g_N = (2,0), g_R = (2,0), g_V = (\bar{\zeta} - 1, 1))$
- $\psi_4 = (g_N = (2,0), g_R = (\bar{\zeta}, 0), g_V = (\bar{\zeta} - 1, 1))$

These pseudo-genotype states adhere to the following conditions: $g_N = (2,0)$ so that the normal pseudo-genotype is diploid with respect to the reference alleles, and $\zeta(g_V) = \bar{\zeta}$ and $b(g_V) \in \{1, \bar{\zeta}_1, \bar{\zeta}_2\}$. The number of variant alleles $b(g_V)$ is at least one, in other words we do not consider genomic loci that are not mutated. When $b(g_V) \in \{\bar{\zeta}_1, \bar{\zeta}_2\}$, we set $g_R = g_N$ (as in ψ_1, ψ_2). For $b(g_V) = 1$, we consider two scenarios: (i) either the point mutation event has happened before the copy number event, in which case we set $g_R = g_N$ (see ψ_3), or (ii) the copy number event preceded the point mutation, where we choose g_R such that $\zeta(g_R) = \bar{\zeta}$ and $b(g_R) = 0$ (as in ψ_4).

We note that for some copy number configurations such as when $\bar{\zeta}_1 = \bar{\zeta}_2$ or $\bar{\zeta}_2 = 0$, some ψ_i values will be identical. For example, when total copy number is equal to one, the possible pseudo-genotype states in the PCN scheme are $\mathscr{P} = \{\psi_1 = (g_N = (2,0), g_R = (2,0), g_V = (0,1)), \psi_2 = (g_N = (2,0), g_R = (1,0), g_V = (0,1))\}$.

2.3.5 The likelihood function

Given the priors over pseudo-genotypes, the likelihood for each data point is:

$$p(b_i|\phi_i, d_i, \pi_i, t) = \sum_{\psi_i \in \mathscr{G}^3} p(b_i|\phi_i, d_i, \psi_i, t) p(\psi_i|\pi_i) \tag{2.8}$$

To address overdispersion, we have modelled the conditional distribution of variant allele counts b_i with a Beta-Binomial distribution, characterized in terms of mean and precision as follows:

$$p(b|d,m,s) = \binom{d}{b} \frac{B(b+sm, d-b+s(1-m))}{B(sm, s(1-m))} \tag{2.9}$$

where B is the Beta function. To reflect our assumptions over the sample sub-population structure, we set the mean value to a function of pseudo-genotypes, cellular prevalence, and cellularity for each data point, that is $m = \xi(\psi^n, \phi^n, t)$. All data points share the same precision s.

2.4 Inference

The main objective of this procedure is to infer the desired outputs of our model, namely for genomic locus i, induced cluster assignments by c_i and cellular prevalences ϕ_i. We use a Gibbs sampler to draw samples from the posterior distribution of the model. We initialize the sampler such that all customers are in their own clusters. Let c_{-i} be the customer connection configuration with customer i's outgoing connection removed. Let $x_i = (b_i, d_i)$ denote the observed data, namely, variant and total allele counts.

The full conditional distribution of c_i is:

$$p(c_i|c_{-i}, x_{1:N}, \lambda) \propto p(c_i|\lambda)p(x_{1:N}|c_i, c_{-i}, \lambda) \tag{2.10}$$

where $p(c_i|\lambda)$ is the same as equation 2.4 and λ is the set of all hyperparameters. Let x_{T_k} be the set of customers in cluster T_k or equivalently, the set of customers sitting at table k, then the likelihood term factors in:

$$p(x_{1:N}|c_{-i}, c_i = j, \lambda) = \prod_{T_k \in T(C)} p(x_{T_k}|\lambda) \tag{2.11}$$

where $T(C)$ is the partitioning induced by current customer connection configuration C. The term $p(x_{T_k}|\lambda)$ further expands as:

$$p(x_{T_k}|\lambda) = \int (\prod_{i \in T_k} p(x_i|\theta, \lambda))p(\theta|\lambda)d\theta \tag{2.12}$$

where the likelihood $p(x_i|\theta, \lambda) = p(b_i|\phi_i, d_i, \pi_i, t)$ is the same as equation 2.8.

Since our prior over cellular prevalences ϕ_i is non-conjugate to the likelihood, we resolve to a cached version of Griddy Gibbs method [25] to compute the above integral. At the end of each iteration (i.e., when all customers are reassigned), we sample ϕ_k, for each cluster k as follows:

$$\phi_k \sim p(\phi_k|x_{T_k}, \pi_{T_k}, t, \lambda) \propto p(\phi_{T_k}|\lambda)p(x_{T_k}|\phi_{T_k}, \lambda, \pi_{T_k}, t) \tag{2.13}$$

where $p(\phi_{T_k}|\lambda)$ is the probability density function of a uniform distribution.

This Gibbs sampler potentially displaces more customers at each step, and as such might have better mixing properties compared to the traditional CRP Gibbs sampler [2]. Figure 2.5 shows such a step in ddCRP.

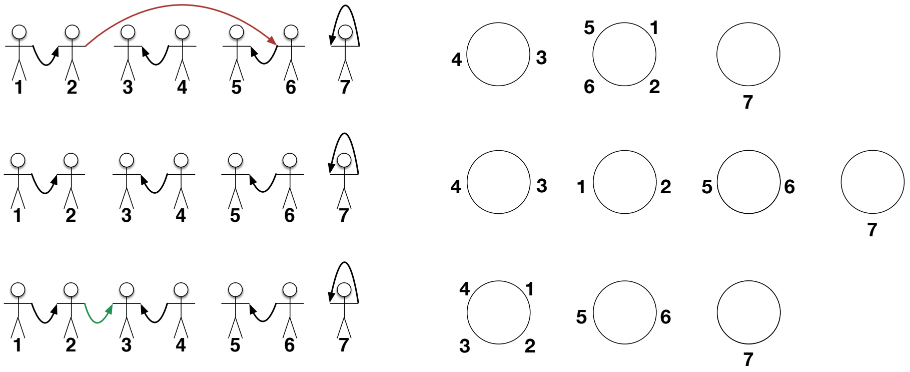


Figure 2.5: Possible moves by the sampler. Left column shows customer connection and right column shows induced table configuration at each step. We want to remove the outgoing connection of customer two, i.e., $c_2 = 6$ (top row, the red arrow). When this connection is removed, the second table is split into two tables, with customers one and two sitting at one table and customers five and six sitting at a new table (middle row). Customer three is picked as the new connection for customer two, i.e., $c_i^{\text{new}} = 3$, and this causes their respective tables to merge (bottom row, the green arrow).

2.5 Resampling hyperparameters

α and a are resampled using methods described in [2]. Briefly, we used the following Gibbs move to update the value of hyperparameter α given the customer connection configuration C:

$$p(\alpha|C) \propto \alpha^K [\prod_{i=1}^{N} (\alpha + \sum_{j \neq i} f(d_{ij}))]^{-1} p(\alpha) \tag{2.14}$$

where $K = \sum_i^N c_i = i$, i.e., the number of self-connections, and $p(\alpha) \sim \text{Gamma}(\alpha_0, \beta_0)$ is the Gamma prior over α with shape and rate parameters α_0 and β_0.

The decay function parameter a is updated using the following Gibbs move:

$$p(a|C,\alpha) \propto [\prod_{i:c_i \neq i} f(d_{ij},a)][\prod_{i=1}^{N}(\alpha + \sum_{j=1}^{i-1} f(d_{ij},a))]^{-1} p(a|\alpha) \tag{2.15}$$

where we assume a uniform prior on a independent of α.

Since the decay function is exponential in our model, we use the Griddy-Gibbs [25] approach to sample approximately from equation 2.15.

We use the method proposed in [21] for resampling s.

Gamma distributed priors are characterized using shape α and rate β parameters. Equation 2.16 shows the corresponding distribution function:

$$g(x;\alpha,\beta) = \frac{\beta^{\alpha}, x^{\alpha-1} e^{-x\beta}}{\Gamma(\alpha)} \tag{2.16}$$

where $\Gamma(\alpha)$ is the Gamma function.

By default, hyperparameter resampling is enabled in our experiments in this work, unless otherwise specified. We note that to explore the model's sensitivity to the value of hyperparameters, in some of our experiments in chapter 3, we disable hyperparameter resampling. We specify this in the description of those experiments.

2.6 Implementation

This model is implemented in R programming language and is available upon request. It is built upon the implementation of ddCRP in [2].

2.7 Limitations and future extensions

We note that our assumptions regarding the pseudo-genotypes in the bulk data may not agree with the genotype matrix derived from the single cell sequencing experiment. More specifically, in modelling the bulk data, we assume that the tumour consists of exactly 3 subpopulations (i.e., pseudo-genotypes). Equivalently, we assume the existence of three genotypes in the bulk data, where the first two are un-mutated and the third is mutated across all genomic loci. If the single cell sequencing study indicates the existence of a different number of genotypes, then the bulk and single cell sequencing assumptions about the genotypes do not match. We intend to explore a number of ways to mitigate this issue, including using the genotypes observed in the single cell sequencing experiments to inform the prior on the pseudo-genotypes in the bulk data.

The shortcomings of the single cell sequencing method, especially the gross undersampling problem may obstruct using genotypes inferred from this type of data in the likelihood of dd-PyClone. As implemented, our model only works with a single anatomical/spatial tumour sample. We aim to expand it to use multiple samples in future implementations. Our method uses a binarized version of the genotype matrix. It may be possible to use the original genotype matrix in the copy number space to calculate distances between genomic loci and potentially improve accuracy of our estimates.

2.8 Conclusion

In this chapter we gave a precise mathematical formulation of the main objective of this work, that is, how to use a binarized genotype matrix from single cell sequencing data to improve clustering and cellular prevalence estimation for genomic loci from bulk sequencing data.

We proposed a solution based on the distance dependent Chinese Restaurant Process (ddCRP), an infinite clustering framework that enables us to encourage co-occurring genomic loci to cluster together. We then described an inference method for our model based on a MCMC sampling scheme.

Chapter 3

Experiments

In this chapter we report our experimental results over synthetic and real datasets. We begin by describing methods pertaining to all of our experiments. To estimate clustering assignment and cellular prevalences from our MCMC samples we use max PEAR index and the mean over all samples respectively.

We then introduce the Generalized Dollo model, a strategy to generate genotypes via a stochastic process over a phylogenetic tree that simulates SNV and CNV events. We use the genotypes generated by this model to simulate the bulk data. Generating realistic simulated datasets is necessary to test the performance of our method since it is not yet possible to exactly quantify the CNV and SNV state in real datasets and therefore we cannot accurately assess the accuracy of our model using real datasets. We proceed to compare our method against existing methods. To gauge performance of results we use V-Measure index and mean absolute error.

Real data experiments come next. We introduce a dataset consisting in five timepoints of a Triple-Negative Breast Cancer (TNBC) xenograft experiment. We benchmark our method against other existing methods over this dataset and report the results. Finally, we present parameter sensitivity and MCMC convergence analysis results.

3.1 General information

3.1.1 Clustering summarization

To cluster genomic loci we first compute the posterior similarity matrix and then maximize the PEAR index to compute a point estimate [11] as implemented in the R package mcclust provided by the authors in [10]. We estimate the cellular prevalence for each genomic locus as the mean of after burn-in MCMC samples.

3.1.2 Clustering evaluation

Clustering performance is measured with respect to the ground truth data in two respects; first V-measure index that evaluates clustering completeness and homogeneity [27], and second, the accuracy of cellular prevalence estimates is reported as the average absolute error.

3.2 Simulated data

Here we introduce our simulation scheme. We first use the Generalized Dollo (GD) model to simulate genotypes. We then use these genotypes to simulate the bulk data. A binarized version of the genotypes is used to inform our prior in our method, while the bulk data constitutes the main input to our model and the competing methods. Figure 3.1 shows the high-level data simulation workflow.

3.2.1 Simulating genotypes

Generalized Dollo model

We used a variation of the Stochastic Dollo (SD) model, called Generalized Dollo Model (GD) to simulate synthetic data accounting for both SNVs and CNVs. SD is a stochastic process that models evolution of binary features (in our case, point mutations) along a phylogenetic tree. A feature could only be gained on one point on the tree, and could be lost multiple times on different branches, but when lost, it cannot be regained [1].

A limitation of SD is that it is restricted to binary features. For instance, it can

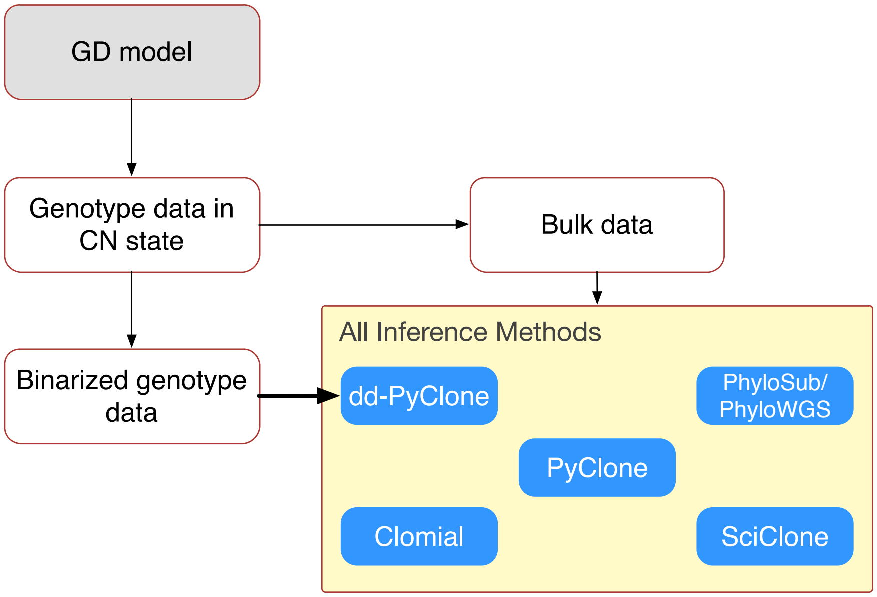


Figure 3.1: High-level data simulation workflow. First, the GD model is used to generate genotype data in copy number state (Δ^{CN}). Second, the genotype data is converted into bulk data. This is given as input to all the methods tested in this work to be used to infer the clustering assignment and cellular prevalences of genomic loci. Third, the genotype data in CN state is converted into a binary genotype matrix and supplied only to our method, dd-PyClone, whereby it is used to construct an informed prior over the partitions of genomic loci (the bold arrow).

only model presence and absence of a mutation at a certain genomic locus.

Multi State Stochastic Dollo (MSSD) model [1] relaxes this restriction by expanding the *present feature* state and allowing transition within this expanded state space. For example, MSSD allows transition and transversion point mutations in addition to deletion.

MSSD can only model evolution after a SNV has happened. This is not a correct assumption when modelling copy number variation events where we would like to be able to account for copy number changes before a point mutation has happened.

To resolve this problem, in addition to expanding the *present feature*, we also expand the *absence feature* and allow transition within these new states. This is the GD model. Once the system *gains the feature*, that is, it transits into the *present features* state subspace, it can make transitions within this subspace, but cannot go back to the previous state. Figure 3.2 illustrates SD, MSSD, and GD side by side on a specific phylogenetic tree for a particular genomic locus.

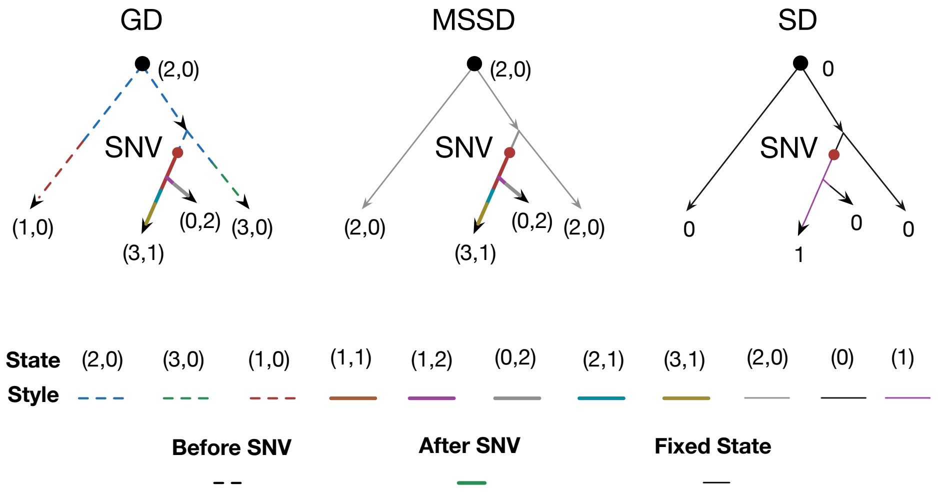
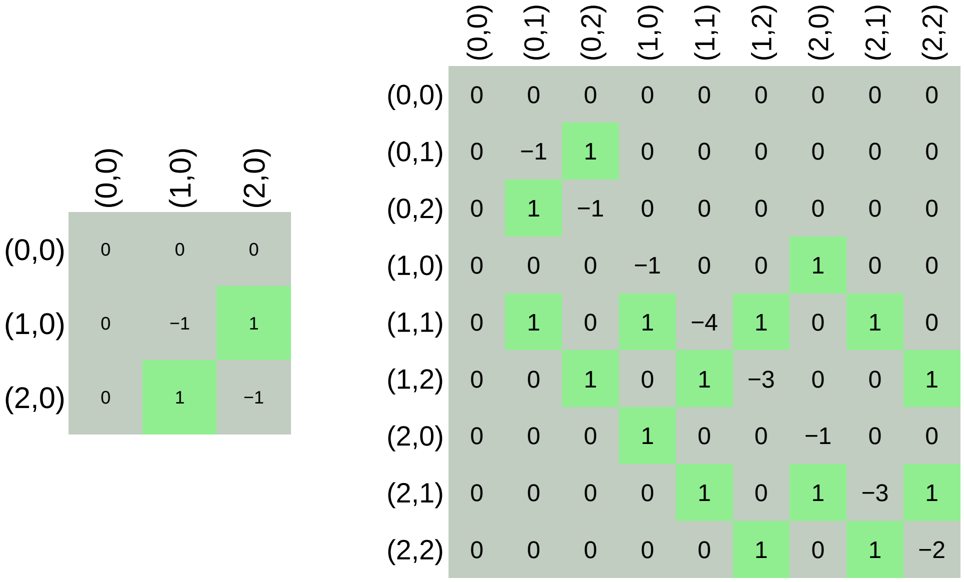


Figure 3.2: An instance of Generalized Dollo, Multi state stochastic Dollo and Stochastic Dollo models over a rooted phylogenetic tree for a single genomic locus side by side. We assume that a SNV has happened at the red dot on the tree. Dashed lines represent the GD model's run over the subtree before the SNV has happened. The thick solid lines represent the process after the SNV has happened. The thin solid lines represent a fixed state, i.e., the process can only handle a fixed state before the SNV gain event. The numbers and colours represent the state of the process (CTMC) at that point. GD can model multiple states on branches where SNV does not appear, while MSSD is forced to be in a fixed state in those positions. Hence the space of problems that GD models is a superset of that of SD.

GD model's setup

GD uses a Continuous-Time Markov Chain (CTMC) to simulate the evolution of genomic loci states along the paths of the phylogenetic tree. The state space of this CTMC consists of pairs $(c_1, c_2) \in \mathbb{N}_0 \times \mathbb{N}_0$ where $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ and c_1 and c_2 represent reference copy and variant copy numbers respectively. Rate matrix Q_1 controls the CTMC before the occurrence of the rare-event and rate matrix Q_2 controls the CTMC after the occurrence of the rare-event.

We design Q_1 and Q_2 such that a complete deletion, i.e., transitioning to state $(0,0)$ is not possible. Q_1 only allows transition between states that have zero variant copy number. This simulates the behaviour of the system before a SNV happens. We assume once a mutation is lost, it cannot be recovered, and enforce this assumption in Q_2 by not allowing transition from states with zero variant copy number zero to states with non-zero variant copy numbers.



	(0,0)	(1,0)	(2,0)
(0,0)	0	0	0
(1,0)	0	−1	1
(2,0)	0	1	−1

	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
(0,0)	0	0	0	0	0	0	0	0	0
(0,1)	0	−1	1	0	0	0	0	0	0
(0,2)	0	1	−1	0	0	0	0	0	0
(1,0)	0	0	0	−1	0	0	1	0	0
(1,1)	0	1	0	1	−4	1	0	1	0
(1,2)	0	0	1	0	1	−3	0	0	1
(2,0)	0	0	0	1	0	0	−1	0	0
(2,1)	0	0	0	0	1	0	1	−3	1
(2,2)	0	0	0	0	0	1	0	1	−2

Figure 3.3: Rate matrices for CTMC used on $\tau_{-p_{\text{SNV}}}$ (left) and $\tau_{p_{\text{SNV}}}$ (right). States represent are pairs representing reference and variant allele copy numbers. In this example, maximum allowed copy number for both reference and variant alleles is 2. States to which transition is possible are annotated green. Note that in both rate matrices, first row and column that represent transitioning from and to the complete deletion state, are all zero. This means that it is not possible to reach complete deletion in our model.

Figure 3.3 shows an example Q_{above} and Q_{below} rate matrices. The state space of the Q_{above} is a subset of that of Q_{below} since we do not allow transition to states where a SNV has happened.

Simulating from the GD model

To simulate data for each genomic locus from the GD model on a phylogenetic tree, we randomly pick a point on the tree to designate where the SNV has happened. We call the subtree rooted at SNV point the below-subtree and the remaining part of the tree, the above-subtree. We simulate GD on the above-subtree to determine the copy number state of the SNV point along with other point on this subtree. This accounts for evolution of the genotypes before a SNV has happened. We continue by simulating the GD on the below-subtree to determine the copy number state of decedents of the SNV point. This accounts for evolution of the genotypes after the SNV occurs.

To continue we first establish some notation following [3]. We define a phylogeny denoted by τ to be a continuous set of points and $\mathscr{G}(\mathscr{L},\mathscr{E})$ to be its topology where $\mathscr{L}$ are the leaves and $\mathscr{E}$ the edges. Let $H_i(v)$ denote the state of the CTMC for genomic locus i at point v. For an edge $e \in \mathscr{E}$, let $|e|$ denote its length.

Let τ_{-x} be the tree pruned at point x, that is, the tree with subtree rooted at x, removed. We write τ_x for the subtree of τ rooted at node x, and τ_{-x} the subtree pruned at node x, that is, the subtree with points in $\tau - \tau_x$. Let ρ be a normalized measure that assigns zero weights to absorbing states and equal weights to non-absorbing states. Then the state of the CTMC at the root $H_i(\Omega)$ is distributed according to a categorical distribution with parameter ρ.

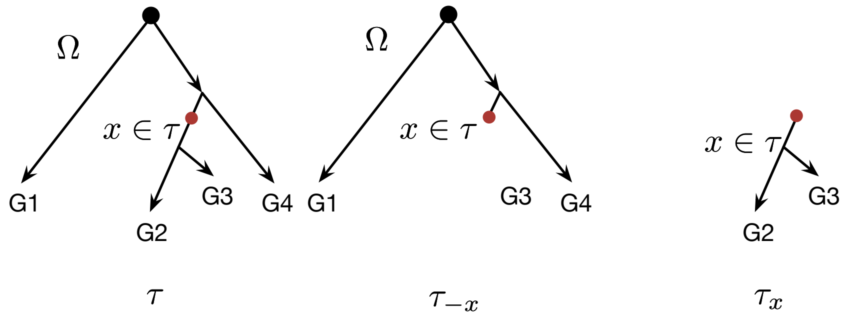


Figure 3.4: A rooted tree topology τ with root node Ω and SNV event at point x (left). $G1$, $G2$, $G3$, and $G4$ represent genotypes. τ_{-x} is the subtree pruned at x (middle). Subtree rooted at x is denoted by τ_x(right). To simulate from the Generalized Dollo model, for a specific genomic locus i, we first pick the SNV position on the tree, then simulate a CTMC on the pruned tree, and simulate another CTMC on the subtree.

Algorithm 1 shows the pseudo code to simulate from Generalized Dollo Model. As input we provide the SimulateGeneralizedDollo procedure with tree topology τ with M leaves and parameter μ, as well as rate matrices $Q_{\text{above}}, Q_{\text{below}}$.

Algorithm 1 Simulating From Generalized Dollo Model

```
procedure SIMULATEGENERALIZEDDOLLO($\tau, \mu, Q_{\text{ABOVE}}, Q_{\text{BELOW}}$)
    for $i$ in $1 : N$ do
        Simulate SNV edge $e_{\text{SNV}} \sim \nu(\tau, \mu)$
        Simulate SNV point $p_{\text{SNV}} \sim \text{Uniform}[0, |e_{\text{SNV}}|]$ on $e_{\text{SNV}}$.
        Simulate state of CTMC at the root node, $H_i(\Omega) \sim \text{Categorical}(u_i)$
        aboveStates ← sampleTreeCTMC($\tau_{-p_{\text{SNV}}}, Q_{\text{above}}$)
        belowStates ← sampleTreeCTMC($\tau_{p_{\text{SNV}}}, Q_{\text{below}}$)
        allStates ← allStates ∪ combine(aboveStates, belowStates)
    return allStates
```

where

$$X \sim \nu(\tau, \mu) \Leftrightarrow p(X = x) = \frac{1}{||\tau|| + 1/\mu} \times \begin{cases} |x| & \text{if } x \neq \Omega \\ 1/\mu & \text{otherwise} \end{cases} \tag{3.1}$$

where sampleTreeCTMC(τ, Q) simulates along a phylogeny, a substitution CTMC and a substitution-deletion CTMC for rate matrices Q_1 and Q_2 respectively.

Since ν has a point mass μ on Ω, there is a non-zero probability that a SNV happens at the root and hence $\tau_{-p_{SNV}} = \emptyset$. In this case sampleTreeCTMC$(\emptyset, Q)$ returns $\emptyset$. If a SNV does not happen at the root, then with probability one there are genotypes in the sample that have no variant allele copy at that genomic locus.

This will give us the copy number state of each genotype at each genomic locus. We summarize this into copy-number aware genotype matrix $\Delta^{CN} \in (\mathbb{N} \times \mathbb{N})^{M \times N}$. Each element of this matrix $\Delta^{\text{CN}}_{m,n}$ is a pair = (CN_R, CN_V) where CN_R and CN_V represent reference and variant allele copy numbers respectively at genomic locus n for the m-th genotype. The binary genotype matrix Δ comes from binarized Δ^{CN}. An element of Δ is equal to one if the second element of the corresponding element in Δ^{CN} is non-zero, and it is zero otherwise. Figure 3.5 shows the phylogeny and 10 genotypes each with 48 genomic loci generated from the GD model.

3.2.2 Simulating bulk data

We use the generated genotypes Δ^{CN} from the GD model to simulate the bulk data. This bulk data serves as the input to the competing methods in this work. Our method additionally takes as input a binarized version of the genotype data to inform its prior over partitions of genomic loci.

For each genomic locus i, the simulated bulk data consists in (i) variant and total allele counts (b_i, d_i), (ii) major and minor copy numbers $\bar{\zeta}_1^i$ and $\bar{\zeta}_2^i$, and (iii) tumour cellularity t. We set $t = 1$ for simulated experiments in this work.

Let $\Phi = \Phi_{1:M}$ where Φ_m is the clonal prevalence for genotype m, that is, the fraction of cells in the tumour sample that have genotype m and M be the total number of genotypes. Then ϕ_i, the clonal prevalence of genomic locus i in our sample would be $\Phi.\Delta[,i]$ and $\phi = \Phi.\Delta$. In this work, we set $\Phi_m = \frac{m}{\sum_{j=1}^{M} j}$.

To generate bulk data at genomic locus i, we first simulate d_i, the total number of reads mapping to i from a Poisson distribution with parameter 10,000. We then use the CN state of the most prevalent genotype from Δ^{CN} (here, it would be the M-th genotype) at locus i to set the major and minor CNs for the bulk. That is we set $\bar{\zeta}_1^i = \text{Maximum}(\Delta^{\mathrm{CN}}_{M,i})$ and $\bar{\zeta}_2^i = \text{Minimum}(\Delta^{\mathrm{CN}}_{M,i})$. To simulate the variant allele counts b_i we have to take into account the aggregate effect of all genotypes at locus i in Δ^{CN}. This means that the ψ_i-s will be slightly different from subsection 2.3.3 in chapter 2, that is, instead of containing normal, reference, and variant subpopulations $\psi_i = (g_N^i, g_R^i, g_V^i)$, it should contain normal, and all the genotypes from Δ^{CN}. With a slight abuse of notation, we denote this by $\psi_i^*(\Delta^{\mathrm{CN}}) = \psi_i^* = (g_N^i, g_1^i, g_2^i, ..., g_M^i)$. We also have to modify the definition of ξ to work with the new ψ^* as follows:

$$\xi^*(\psi_i^*, \Phi_i, t) = \frac{(1-t)\zeta(g_N)}{Z^*}\mu(g_N) + t\sum_{j=1}^{M} \frac{\Phi_j \zeta(g_m^i)}{Z^*}\mu(g_m^i) \tag{3.2}$$

where Z^* is the appropriate updated normalizing constant. Finally, we have $b_i \sim$ Beta-Binomial$(d_i, \xi^*(\psi_i^*, \phi_i, t), s)$ where we set $s = 1000$. Algorithm 2 summarizes the bulk data simulation procedure:

Algorithm 2 Simulate Bulk Data

```
procedure SimulateBulkData($\Phi, \Delta^{\text{CN}}, s, t$)
    for $i$ in $1:N$ do
        $d \leftarrow d \cup$ Simulate $d_i \sim \text{Pois}(10,000)$
        $\bar{\zeta}_1 \leftarrow \bar{\zeta}_1 \cup \text{Maximum}(\Delta^{\text{CN}}_{M,i})$
        $\bar{\zeta}_2 \leftarrow \bar{\zeta}_2 \cup \text{Minimum}(\Delta^{\text{CN}}_{M,i})$
        $b \leftarrow b \cup$ Simulate $b_i \sim \text{Beta-Binomial}(d_i, \xi^*(\psi_i^*, \Phi_i, t), s)$
    return $\{d, b, \bar{\zeta}_1, \bar{\zeta}_2, t\}$
```

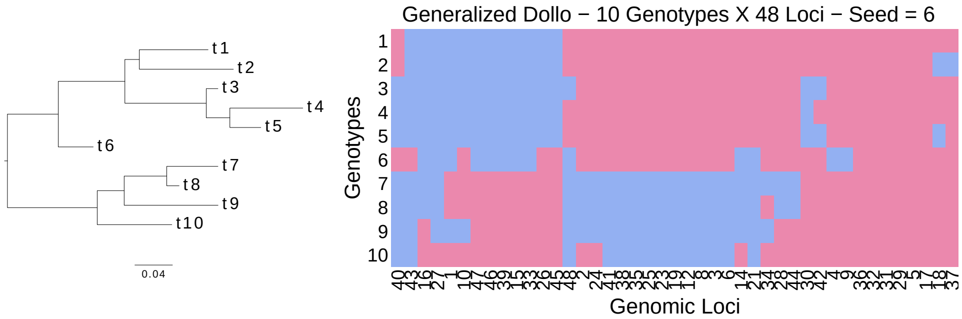


Figure 3.5: Transposed binarized simulated genotypes Δ (right) from Generalized Dollo process over a fixed phylogeny (left). The original genotype matrix Δ^{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

3.2.3 Benchmarking against existing methods

We benchmarked our method against existing methods over synthetic data. We simulated 10 synthetic datasets each with 10 genotypes over 48 genomic loci from the GD model. Figure 3.5 shows the genotype heat map and phylogeny used in one of the datasets. The rest of the figures are in the appendix A, section A.1. In these experiments, dd-PyClone was always supplied with the binarized genotype matrix. The other methods were given the bulk data simulated from the original genotype matrix in the copy number state. We refer the reader to Figure 3.1 for a schematic of the simulated data generation workflow. Figure 3.6 shows the performance results of this experiment.

Parameters for synthetic data generation

We used the following setup to generate synthetic data:

```
t = 1
d = 10,000
s = 1000
μ = 100
number of genotypes = 10
number of genomic loci = 48
Max Total Copy Number = 4
```

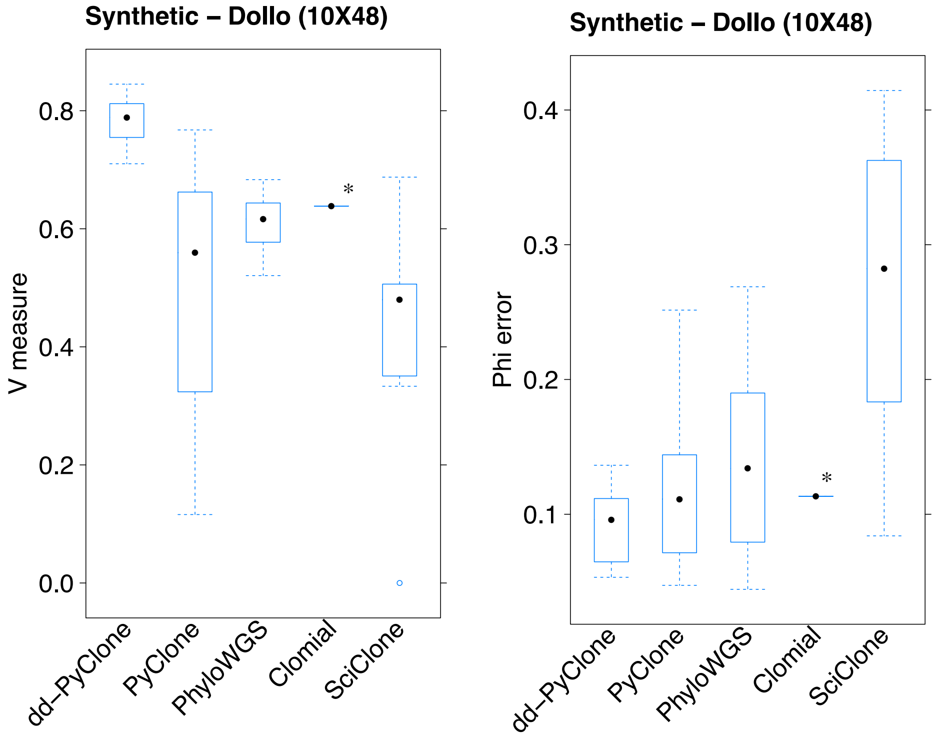


Figure 3.6: Performance results for dd-PyClone and existing methods over synthetic data. Right panel shows clustering assignments performance. Left panel shows cellular prevalence mean absolute error. *We were not able to run Clomial with more than one dataset, since in the rest of the datasets, the number of clusters are more than 10, and in this case, Clomial never converged.

Parameter setting in method comparison experiments

We ran PyClone version `0.12.3` for 10,000 iterations with a burn-in of 1000 and thinning of 1. Remaining parameters were set as follows:

```
num_iters:10,000
base_measure alpha=1
base_measure_ beta=1
concentration=1
prior shape = 1
prior rate = 0.001
```

```
density = pyclone_beta_binomial
beta_binomial_precison value = 1000
beta_binomial_prior shape = 1
beta_binomial_prior rate = 1
beta_binomial_precison proposal precision = 0.01
tumour_content = 1
error_rate = 0.001
```

We used Clomial version `1.4.0` and provided it with the correct number of clusters via its `c`. Remaining arguments were set as follows:

```
maxIt = 20
binomTryNum=10
c = True Number of Clusters
```

We downloaded PhyloWGS with git tag `smchet1-31-g57294e3` and used it with the default settings for the following parameters:

```
—mcmc–samples = 2500
—mh–iterations = 5000
```

Since this version of PhyloWGS did not output clonal frequencies, we edited the source code to extract these values. Furthermore, to simplify comparison with other methods, we provided PhyloWGS with an empty copy number file.

We used SciClone version `1.0.7`. We set the maximum number of clusters to its true value. The remaining parameters are as follows:

```
minimumDepth = 2
copyNumberMargins = 0.25
maximumClusters = True number of clusters.
```

3.3 Real dataset

3.3.1 Triple-negative breast cancer Xenograft data

To test our method over a real dataset, we used a subset of samples from a triple-negative breast cancer xenograft study [8] where breast cancer tissues from 55 patients were transplanted into highly immunodeficient mice to generate 30 xenograft lines. Over 3 years, these lines were passaged up to 16 generations.

Whole genome sequencing was performed over a subsample of this cohort to identify candidate genomic positions. It was followed by deep targeted amplicon sequencing of between 100 to 300 SNV positions per sample. 210 cells from five timepoints that span two samples were chosen for single cell genotyping, and about 48 SNV positions were targeted for each timepoint.

The results were post-processed to remove all positions labeled as non-somatic. This was further summarized into constituent genotypes. A consensus phylogenetic tree was inferred using MrBayes [26]. Cells were grouped into clades consisting of high probability branching splits. For each clade a consensus genotype was derived by taking the most prevalent genotype at each genomic locus. Figure 3.7 shows the inferred genotype matrix Δ for each sample. In each timepoint, we only kept genomic loci that were shared between the bulk and single cell genotype data. Inferred genotypes from the triple-negative breast cancer xenograft single cell genotyping study is shown in Figure 3.7.

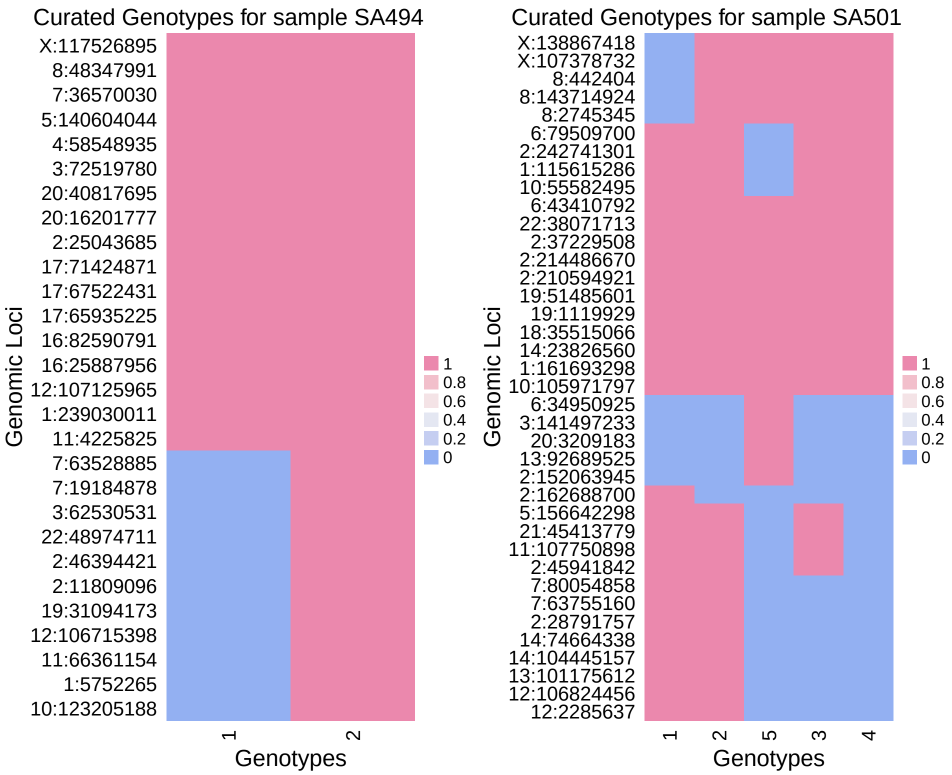


Figure 3.7: Binary genotype matrices for sample SA494 over 29 genomic loci (left) and sample SA501 over 38 genomic loci (right). These are manually curated from a single cell genotype sequencing experiment [8]. Briefly, MrBayes was used to infer a consensus phylogenetic tree over the single nuclei. Then they were grouped into clades according to high probability branching splits. Finally, each clade was assigned a consensus genotype by taking the mode genotype of the clade at each genomic locus.

3.3.2 Establishing the ground truth

Since exact clustering configuration and cellular prevalences of the genomic loci in the real dataset is unknown, we used multi-sample PyClone's result over 11 timepoints from sample SA501 and 4 timepoints from sample SA494 as our gold standard. PyClone in multi-sample mode borrows statistical strength across all timepoints to give better estimates of subclonal structure in individual timepoints.

The following timepoints were used for sample SA501:

`SA501T, SA501X1A, SA501X2A, SA501X2B, SA501X3A, SA501X3B, SA501X4A, SA501X4B, SA501X4C, SA501X4D, SA501X5A.`

The following timepoints were used for sample SA494:

`SA494X4, SA494X3, SA494X2, SA494T`

PyClone was run for 100,000 iterations with a burn-in period of 50,000 iterations. The rest of the settings were identical to synthetic simulation experiments as in listing 3.2.3. Cellular prevalence estimates are summarized in Figure 3.8. This results in an overall clustering and timepoint-based prevalence estimates for each genomic locus.

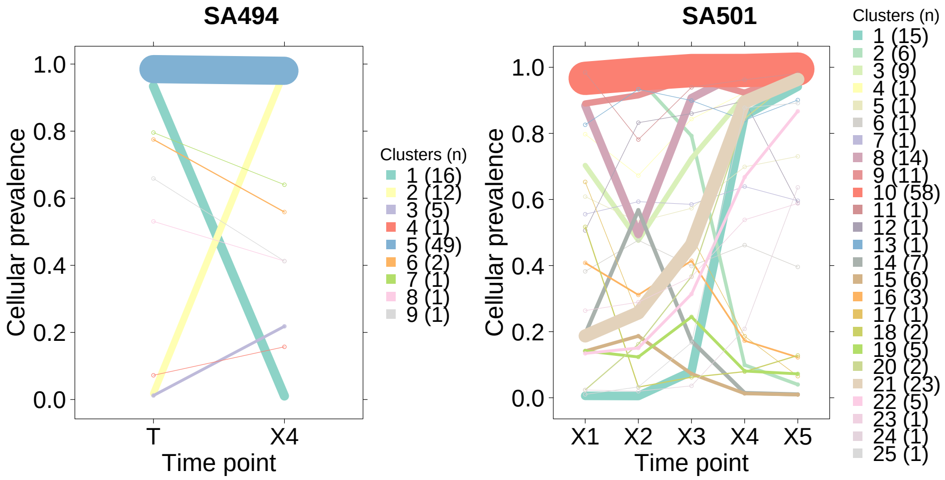


Figure 3.8: Clustering result for multi-sample PyClone over timepoints `SA501 X1, X2, X4`, and `SA494 T, X4`

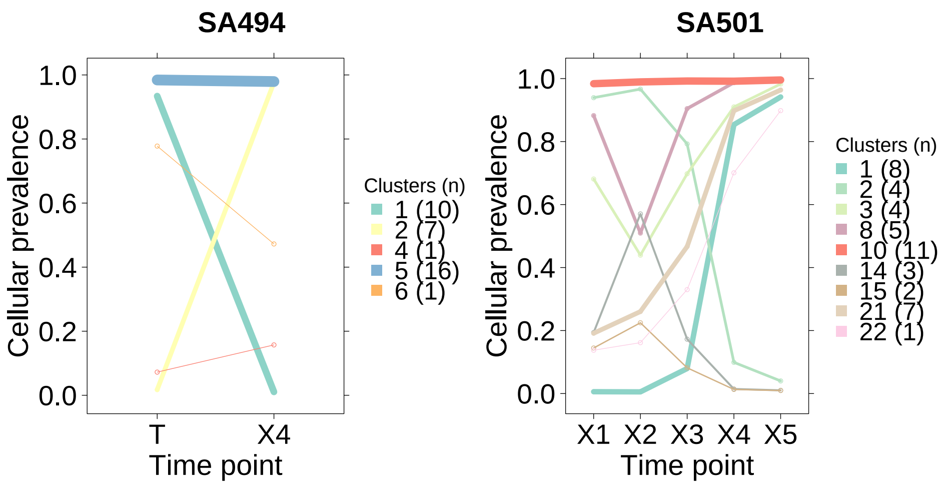


Figure 3.9: Clustering result for multi-sample PyClone over timepoints `SA501 X1, X2, X4`, and `SA494 T, X4` for genomic loci that overlap with those sequenced in the single genotype analysis.

We ran our method along with competing methods over timepoints `SA501 X1, X2, X4`, and `SA494 T, X4` for which we had matching single genotype sequencing data. Figure 3.10 shows the performance of each method against the golden standard.

3.4 Parameter sensitivity

In this section we report our simulation studies aimed at elaborating dd-PyClone's sensitivity to the choice of hyperparameters and noise level. Since hyperparameter a, the decay function parameter, is the distinguishing parameter of our model, we mostly focus on effects of its starting value on our model. To assess our model's robustness to noise, we introduce two types of noise, namely, point error and genotype loss. Finally, we examine our model under varying values of a and presence of noise simultaneously.

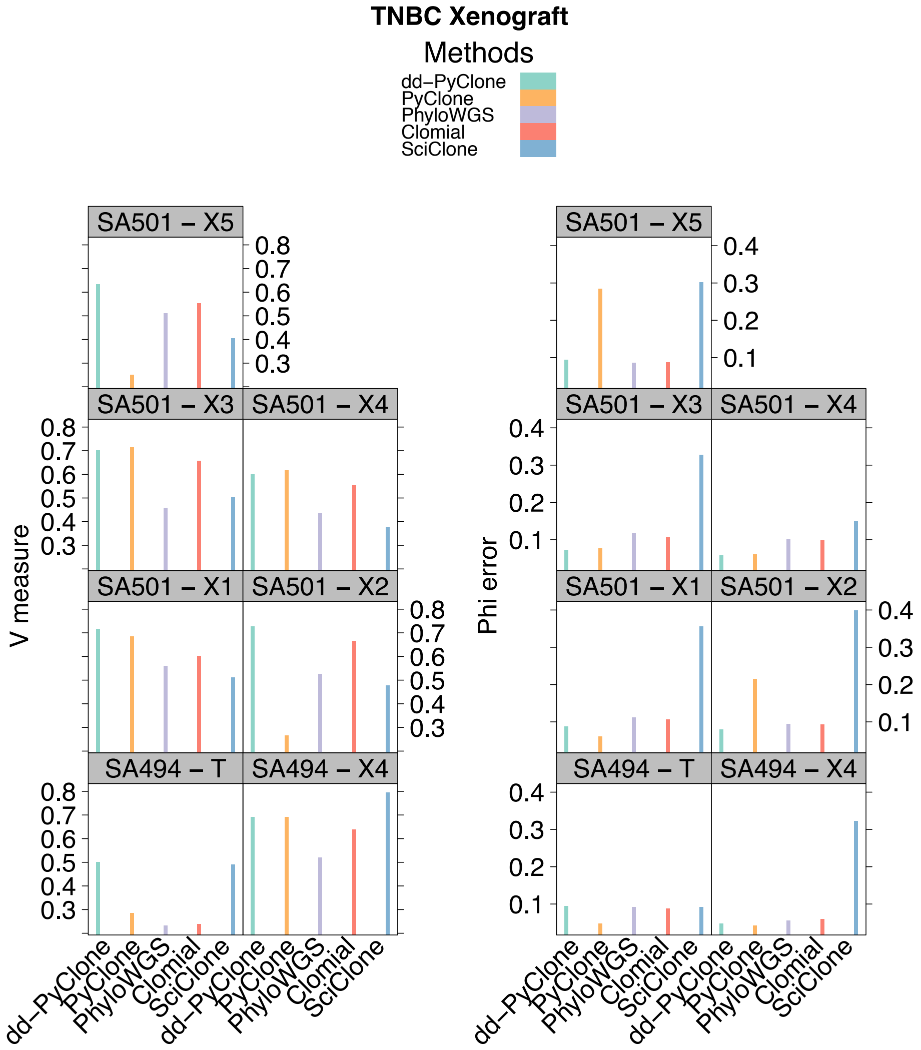


Figure 3.10: Performance results for dd-PyClone and existing methods over TNBC SA501 X1, X2, X4, and SA494 T, X4. Right panel shows clustering assignment performance. Left panel shows cellular prevalence approximation mean absolute error.

3.4.1 Sensitivity to value of a

Figure 3.11 shows the result of running our model with different starting values for the hyperparameter a. In these experiments we disabled resampling of hyperparameters a, α, and s, and fixed them at their starting value. We simulated 10 datasets from the GD model with 5 genotypes over 48 genomic loci. We ran our model 170 times for each dataset, with different initial values for hyperparameters, each time for 200 iterations. Each box plot shows the respective performance index for runs with an identical initial value of a and different values for s and α, each for 5 datasets.

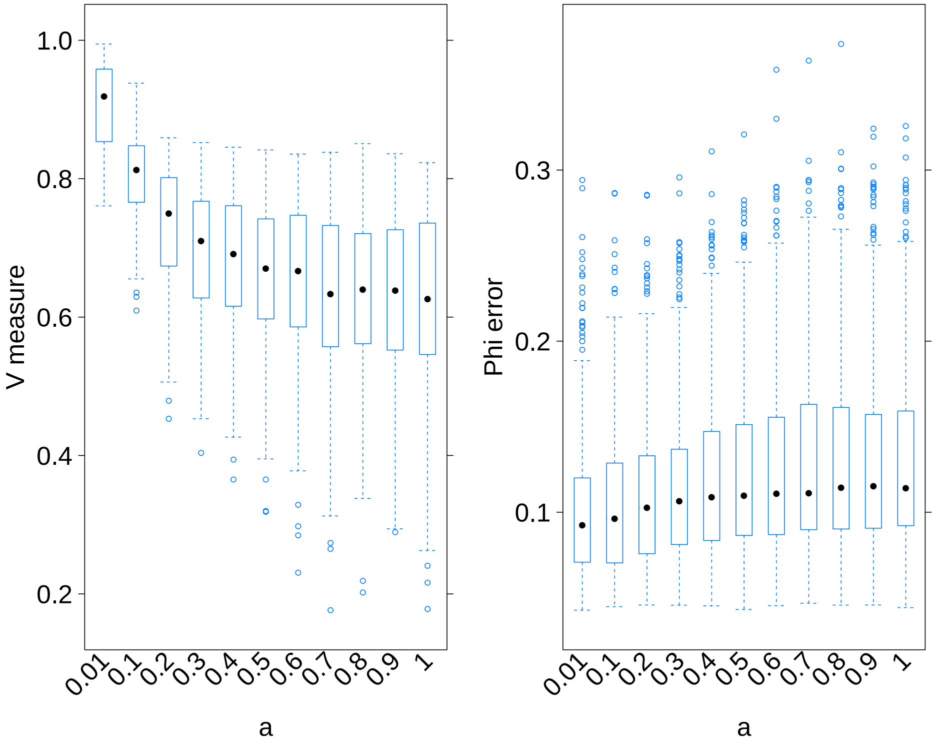


Figure 3.11: Performance over 10 synthetic datasets. Hyperparameter a is fixed at the specific value for each inference run. This result suggests that performance declines with increasing values of hyperparameter a.

3.4.2 Sensitivity to presence of noise

We consider two types of noise. A point noise that affects the status of a single genomic locus for certain genotypes, and a genotype loss noise, where one or multiple genotypes are completely lost. Let the original genotype matrix be $\Delta_{M\times N}$ where M is the number of genotypes and N is the number of genomic loci. In our experiments, we provided the noisy genotype matrix to our model.

Point noise

When a genotype is erroneously marked as mutated at a genomic locus, we say a point noise has occurred. In particular, we assume that a process independently operates on each element of Δ and flips its value with a probability p. This process could be due to false positives in calling SNVs.

Concretely, assume $f_p\colon \{0,1\}_{M\times N} \times [0,1] \to \{0,1\}_{M\times N}$ be a stochastic map parameterized by p corresponding to the point noise process. The filtered matrix would be a random binary matrix $R_{M\times N}$ with elements that follow the distribution in equation 3.3.

$$\Pr(R_{i,j} = k) = \mathbb{1}(\Delta_{i,j} = 0)p^k(1-p)^{(1-k)} + \mathbb{1}(\Delta_{i,j} = 1)p^{(1-k)}(1-p)^k \qquad (3.3)$$

In other words, we can view this process as first sampling a random binary matrix $F_{M\times N}$ each element of which is sampled independently from a Bernoulli distribution with parameter p, $F_{i,j} \overset{iid}{\sim} \text{Bern}(p)$. Then we combine this filtered matrix with the original genotype matrix Δ using an element-wise XOR operation to get filtered matrix R.

Figure 3.12 shows the result of providing our model with a noisy genotype matrix, under various probabilities of noise p. We simulated 10 datasets from the GD model with 10 genotypes and 48 genomic loci. We ran dd-PyClone for 500 iterations and enabled resampling of hyperparameters a, α, and s. Each box plot shows the respective performance index when dd-PyClone was supplied with a noisy genotype matrix with a particular p.

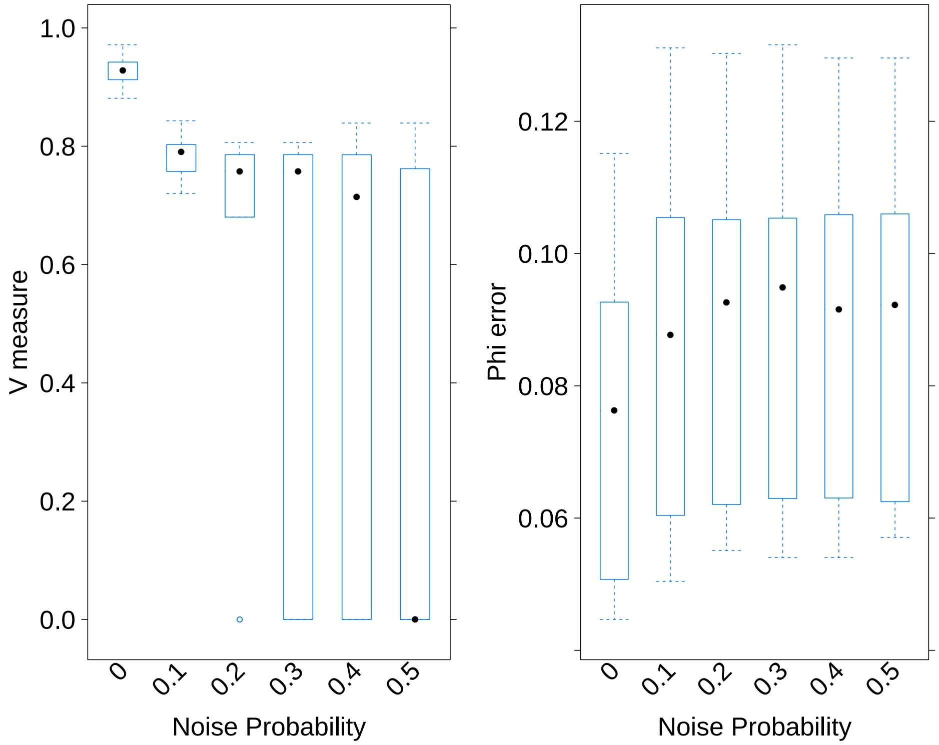


Figure 3.12: Effect of adding point noise on V measure index (left) and mean absolute error of cellular prevalences (right). This result implies that our method is sensitive to high levels of point noise.

Genotype loss

Now we turn our attention to effects of genotype loss. It may happen due to undersampling inherent in single cell genotyping (SCG) experiments. The number of isolated cells that are to be sequenced in SCG experiments is order of magnitudes less than the number of cells in the donor tumour. This may result in undersampling of some genotypes, with less prevalent genotypes being more prone to missing.

Figure 3.13 illustrates the effects of progressively removing more genotypes. We simulated 10 datasets from the GD model with 10 genotypes over 48 genomic loci. For each dataset, we ran dd-PyClone 11 times each for 500 iterations. Each time, we held out a number of genotypes from the original genotype matrix Δ and provided dd-PyClone with the resulting undersampled genotype matrix $\Delta^{'}$. This result implies that if more than half of high prevalence genotypes are observed, using them will improve clustering assignment and cellular prevalence estimates over the genotype-naive methods.

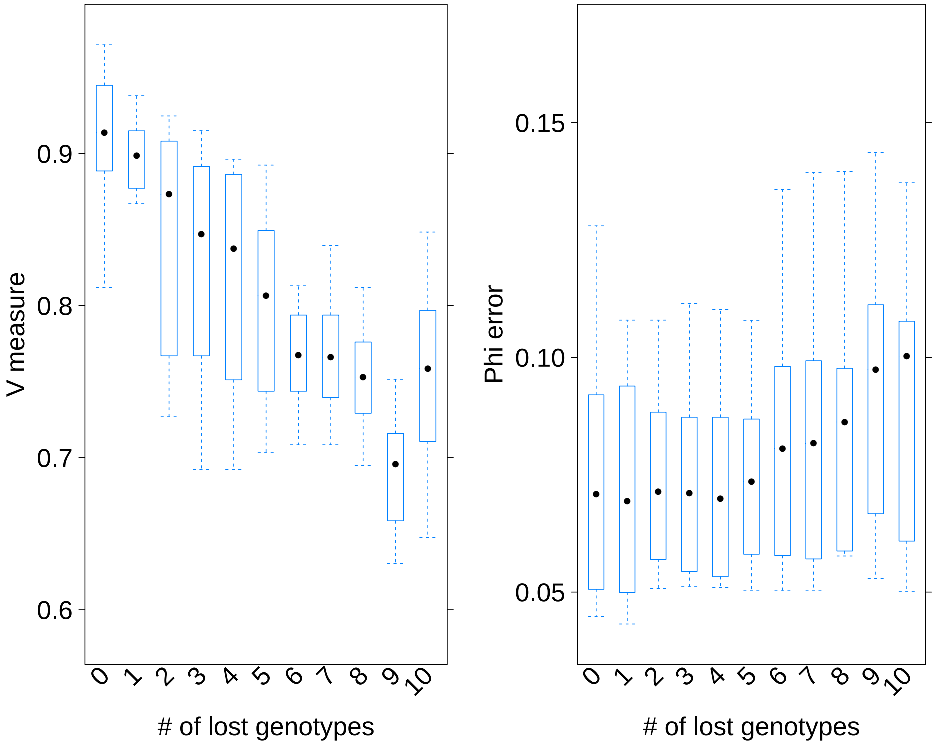


Figure 3.13: Effect of removing genotypes on V measure index (left) and mean absolute error of cellular prevalences (right). Zero genotype loss depicts the model performance under the original genotype matrix. Number of lost genotypes of 10 indicates the performance of the model with no genotypes supplied. In this case, the method essentially falls back to that of PyClone. We have included PyClone's performance over the same datasets with 500 MCMC iterations as a reference (right most box plot). As expected, with loss of genotypes, the method progressively performs worse since it is losing information. The rise in performance when all genotypes are lost could be attributed to the fact that undersampled genotypes are misleading and interfere with the signal in the bulk data.

We also tested the performance of the model under removal of individual genotypes. Figure 3.14 illustrates the effects of removing individual genotypes. The experimental configuration is exactly the same as the above experiment, except that we held out only one genotype from the original genotype matrix Δ and provided dd-PyClone with the resulting undersampled genotype matrix Δ'. This result implies that our method is robust regarding the removal of individual genotypes.

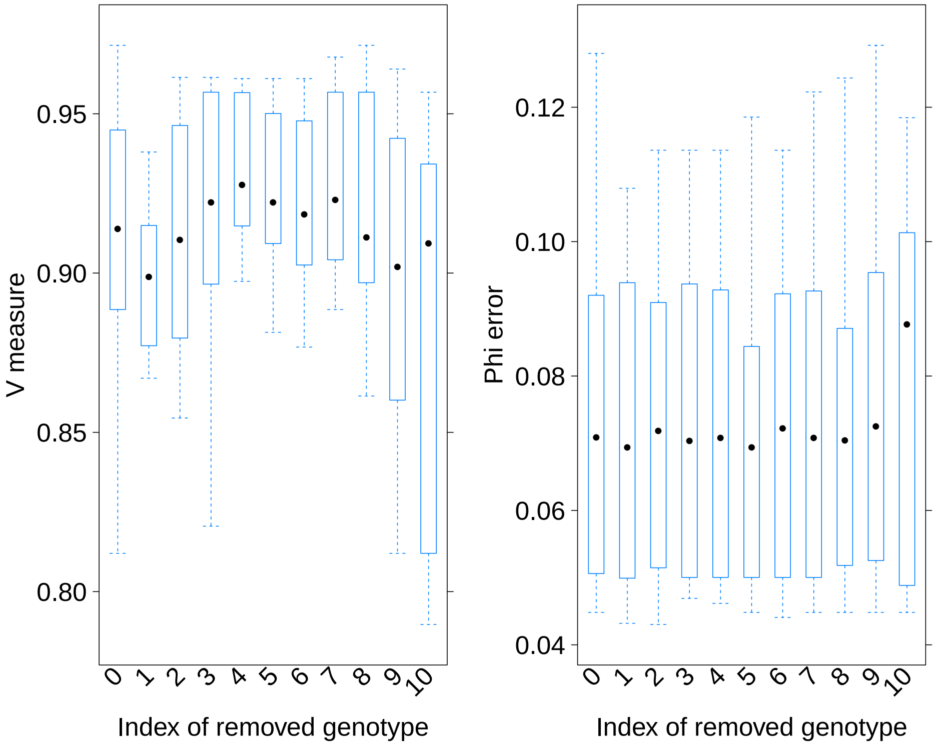
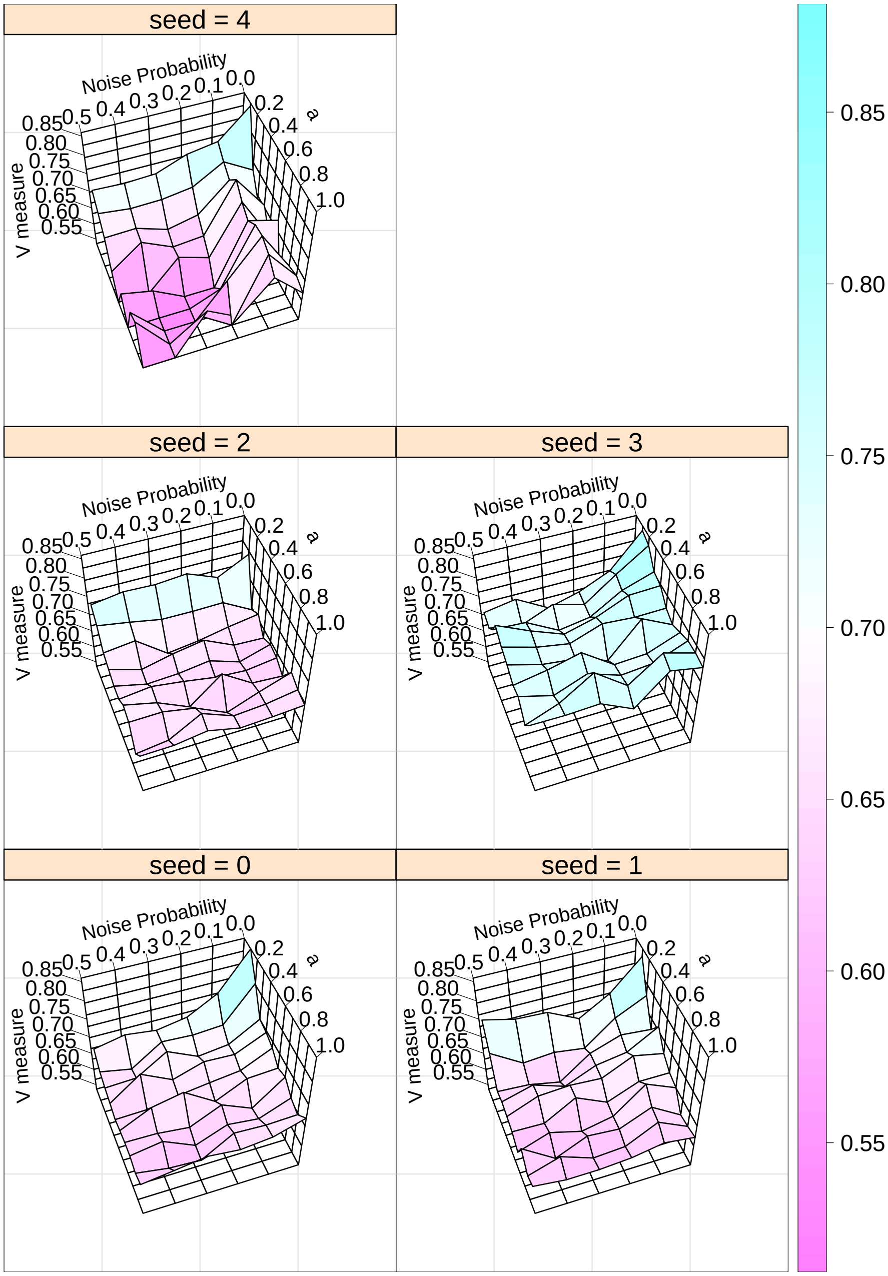


Figure 3.14: Effect of removing individual genotypes on V measure index (left) and mean absolute error of cellular prevalence (right). Zero genotype loss depicts the model performance under the original genotype matrix. Horizontal axes shows which genotype was removed from the genotype matrix supplied to the method. The model is robust to the loss of individual genotypes.

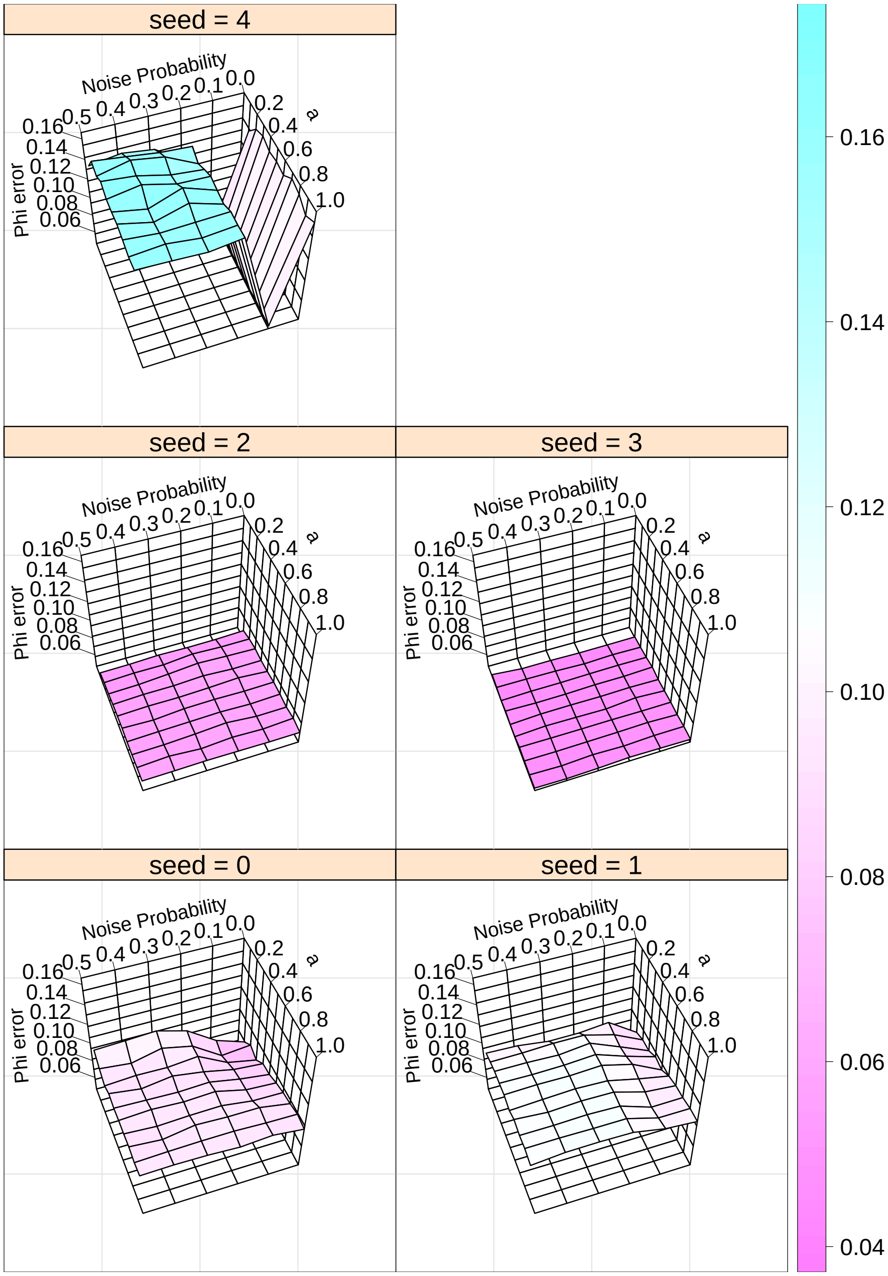
3.4.3 Sensitivity to a and noise

Here we examine the effects of simultaneously varying a and introducing noise. In our first experiment, we added point noise. We simulated five datasets from the

GD model with 10 genotypes over 48 genomic loci. For each dataset, we ran dd-PyClone for 200 iterations 60 times. Each time we fixed the hyperparameters a, α and s to a different starting value and disabled hyperparameter resampling. For each dataset, we introduced point noise with specified probability p to the original genotype matrix, and input the filtered genotype matrix to our model. Results for this experiment are shown in Figure 3.15. It implies that in presence of noise, the model is more sensitive to higher values of decay function parameter a and as a increases, model performance declines.



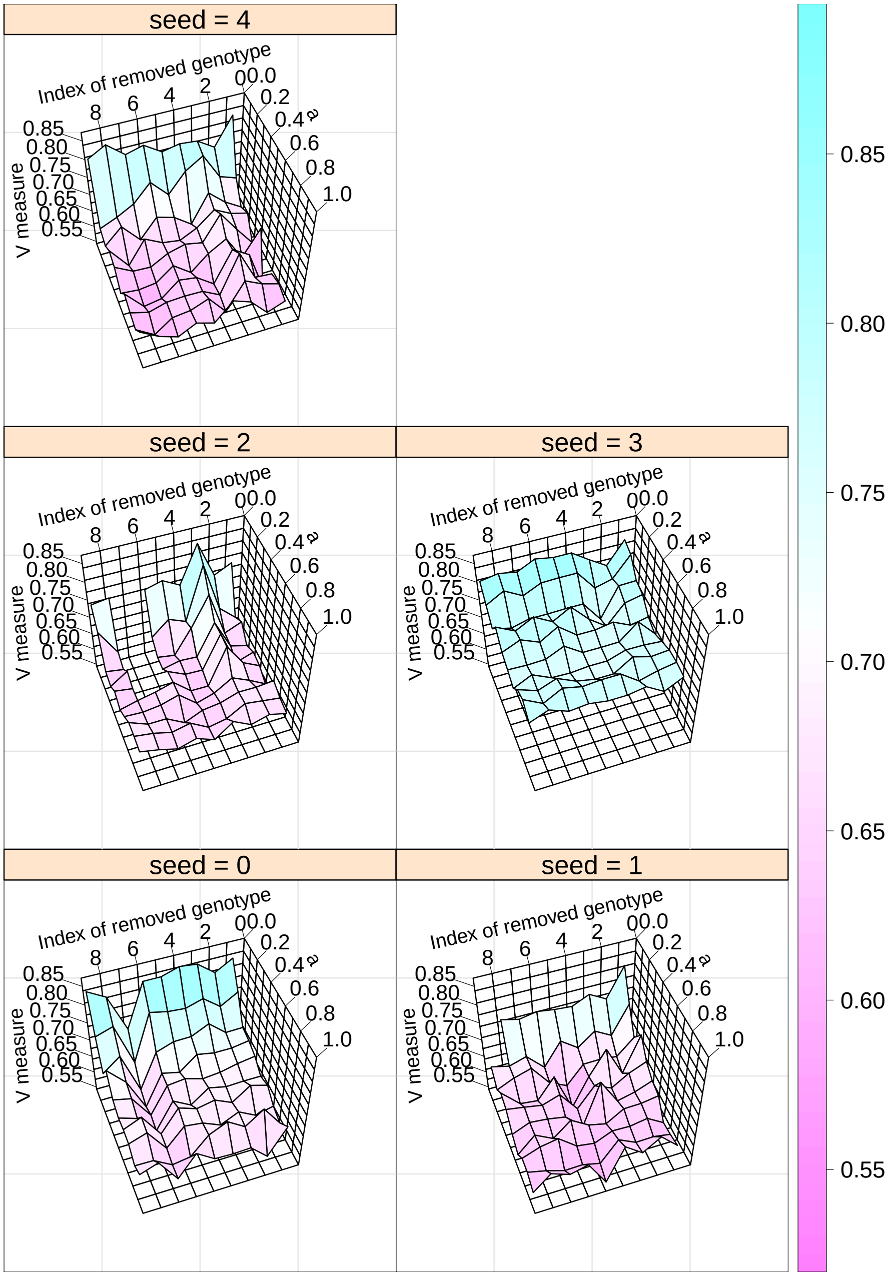
(a) V measure Index



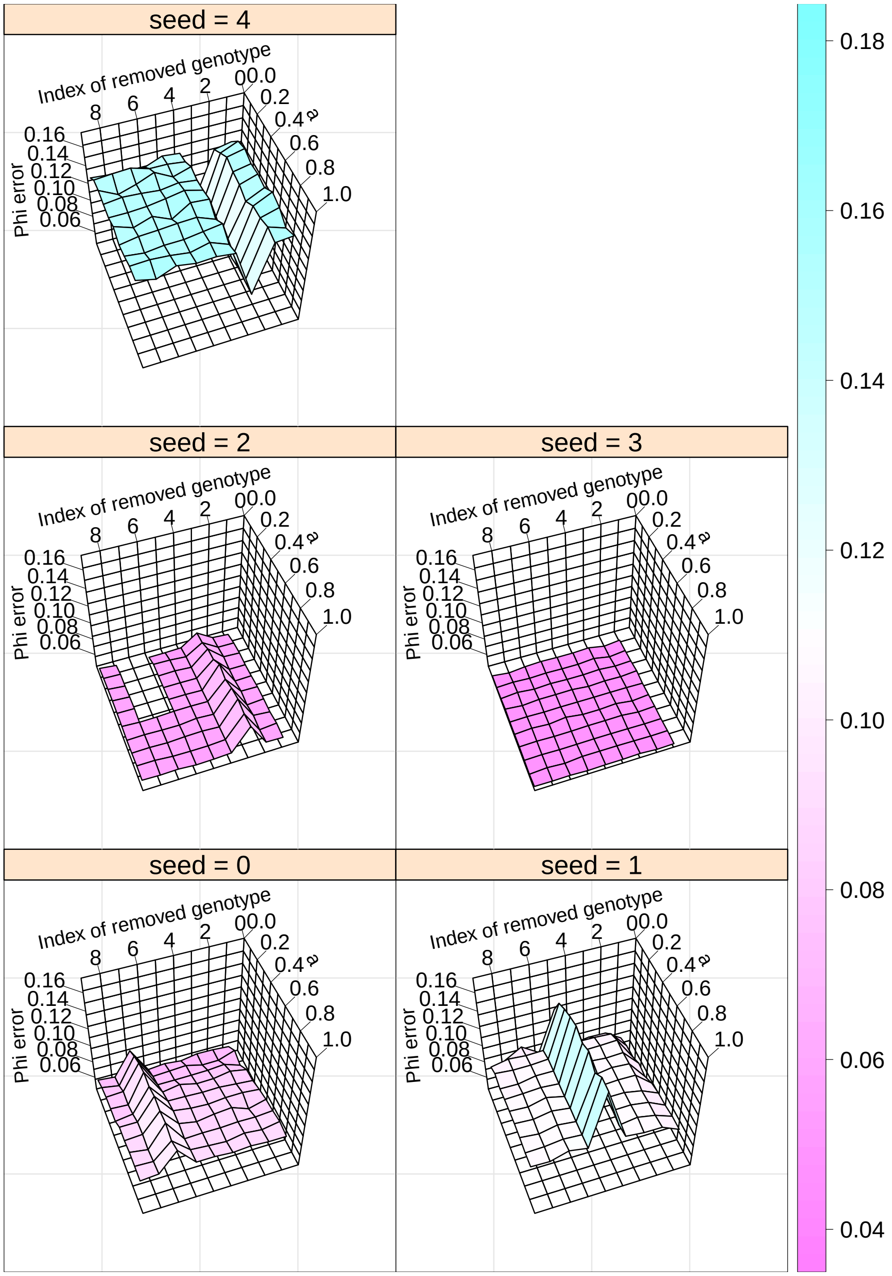
(b) Cellular prevalence error

Figure 3.15: Effect of adding random point noise and varying decay parameter a on V measure index (a) and mean absolute error of cellular prevalence estimates (b) for the five simulated datasets. Beta-Binomial precision parameter s and hyperparameter α are fixed at 1000 and 1 respectively. We note that V measure index is more sensitive to changes in value of a than the level of point noise. Heat map colours represent values in the vertical axis and are included to aid the eyes.

We examined two genotype loss scenarios: one where only a single genotype is lost, and one where progressively more genotypes are missed. Results for the first scenario are in Figure 3.16. Five datasets identical to the point noise experiment were generated. For each dataset, we held out the specified genotype and input the remaining as the genotype matrix to our model (i.e., a matrix with 9 genotypes in our experiments).



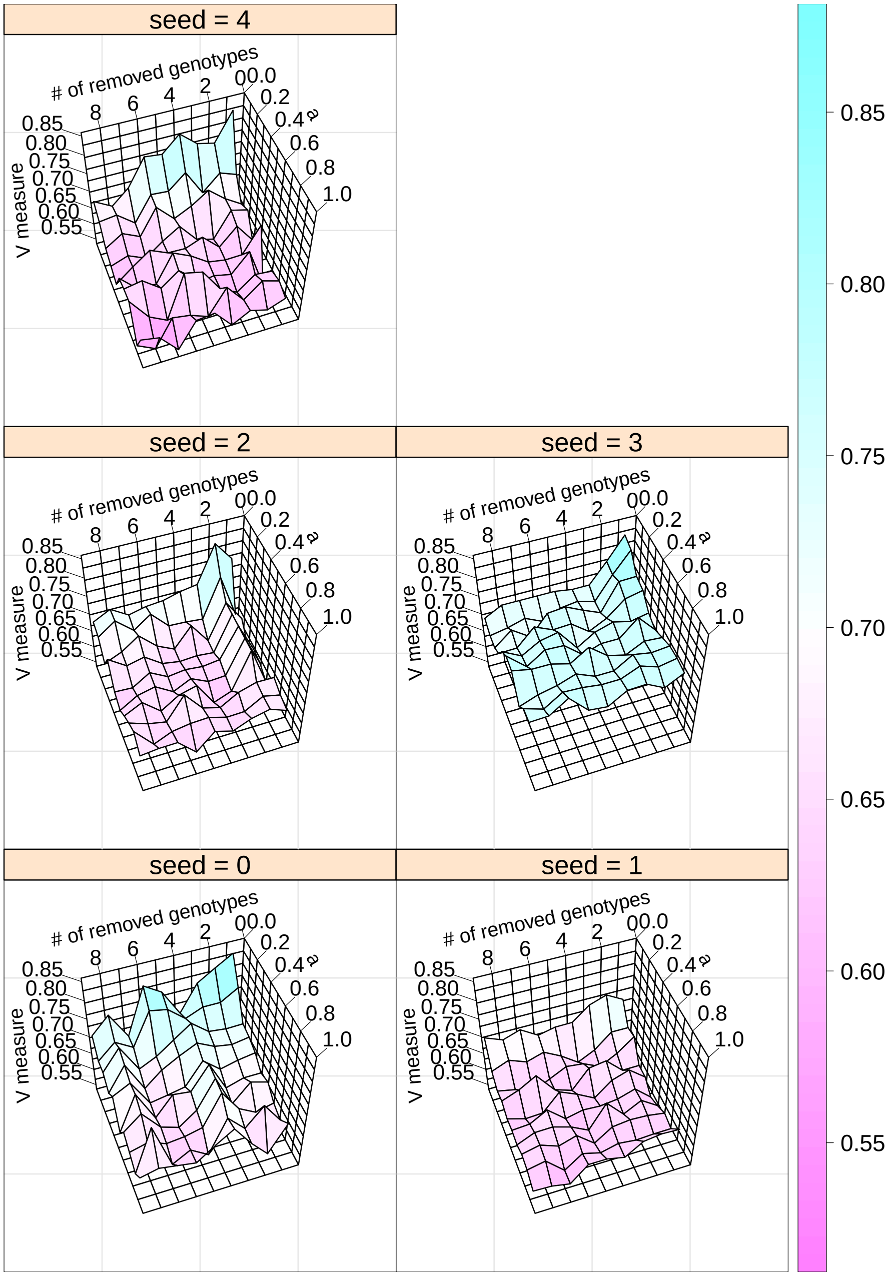
(a) V measure Index



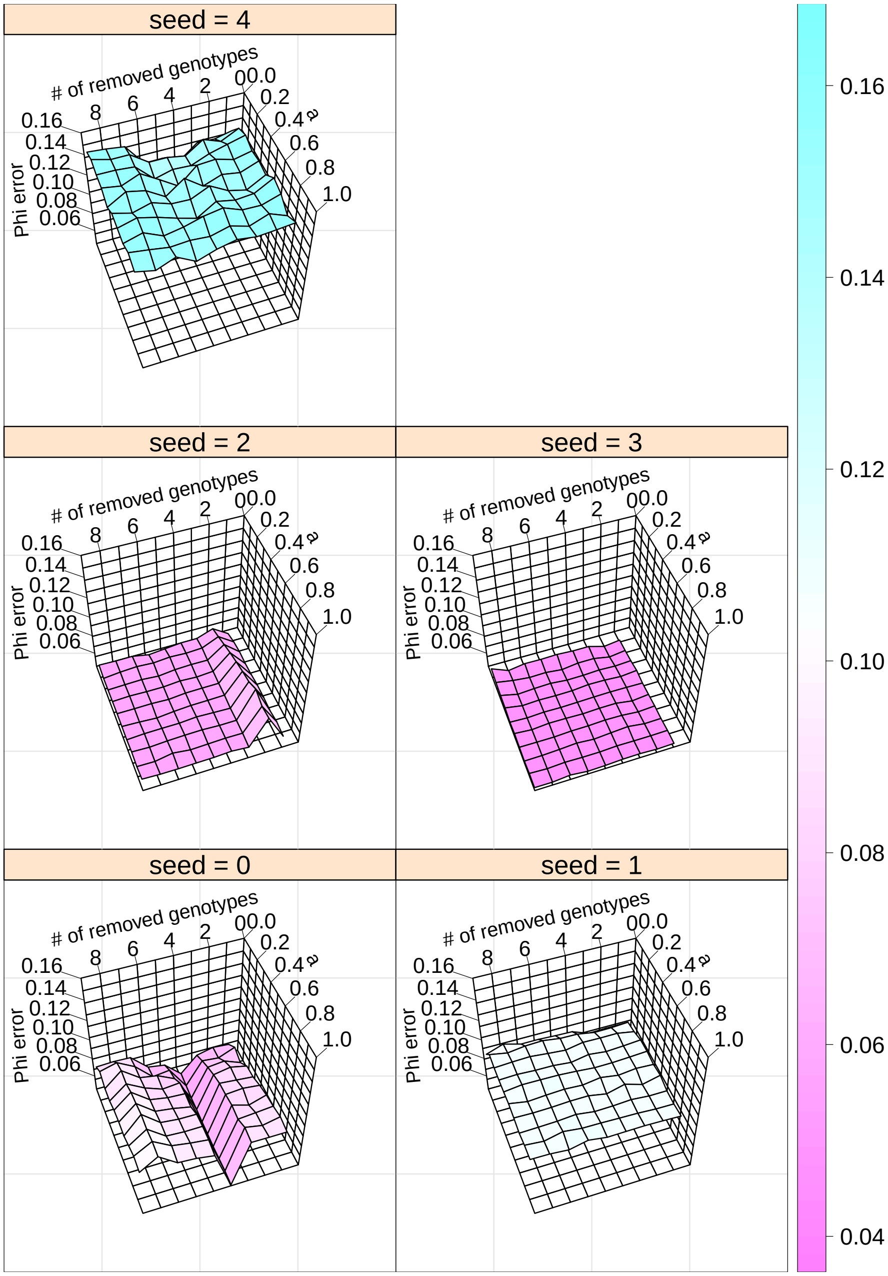
(b) Cellular prevalence error

Figure 3.16: Effect of removing single genotypes and varying hyperparameter a on V measure index (a) and mean absolute error of cellular prevalence estimates (b) for five simulated datasets. Genotypes are sorted in decreasing order of prevalence from right to left. Genotype 1 is the least prevalent and genotype 9 is the most prevalent. Beta-Binomial precision parameter s and hyperparameter α are fixed at 1000 and 1 respectively. We note that V measure index is more sensitive to changes in value of a than removal of single genotypes. Heat map colours represent values in the vertical axis and are included to aid the eyes.

In the second scenario, we progressively removed more genotypes. Figure 3.17 depicts these results. Except for genotype loss, the rest of experiment setup was identical to the first scenario. This result implies that the model is more sensitive to the value of the decay function parameter a than it is to genotype removal.



(a) V measure Index



(b) Cellular prevalence error

Figure 3.17: Effect of removing progressively more genotypes and varying decay parameter a on V measure index (a) and mean absolute error of cellular prevalence estimates (b) for five simulated datasets. Beta-Binomial precision parameter s and hyperparameter α are fixed at 1000 and 1 respectively. We note that V measure index is more sensitive to changes in value of a than removal of multiple low prevalence genotypes. Heat map colours represent values in the vertical axis and are included to aid the eyes.

3.4.4 Effect of non-exchangeability

As stated in the Methods chapter, ddCRP is not an exchangeable prior in general. Figure 3.18 shows the effect of random reordering of customers on our model. It implies that our model is not significantly sensitive to the order of customers.

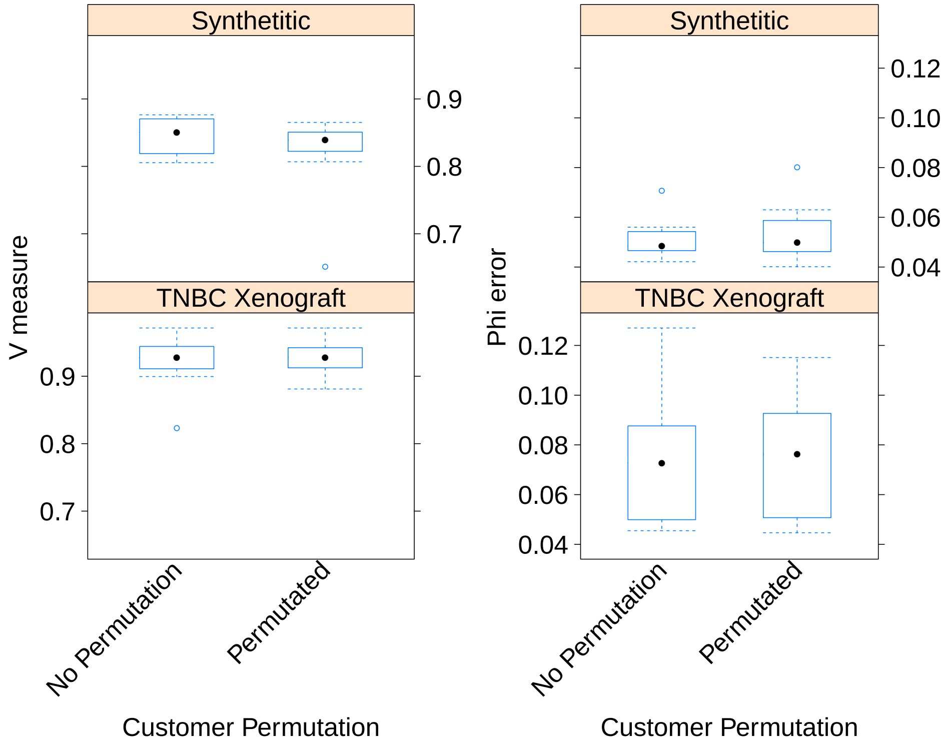


Figure 3.18: Effect of random reordering of customers in each iteration of the sampler vs keeping the order of customers fixed on V measure index (left) and mean absolute error of cellular prevalence (right). The top row depicts results over a synthetic dataset (all settings identical to subsection 3.4.2 except that there is no added artificial noise). The bottom row shows result over a real dataset.

3.5 Computational aspects

Computing the Distance Matrix takes $\mathscr{O}(N^2M)$. Computing the clustering result takes $\mathscr{O}(N^2)$. The complete analysis with 100 MCMC runs on a personal laptop with 2.4 GHz Intel Core 2 Duo and 8GB of RAM memory, for a dataset of 48 genomic loci and 10 genotypes takes about 5 minutes.

3.6 Convergence diagnostics

Following [29] to assess convergence of the MCMC chain for TNBC Xenograft samples SA501 and SA494, we ran 3 chains for 10,000 iterations with random seeds and visually inspected Posterior Similarity Matrices (PSM) to ensure similarity. Figure 3.19 shows the PSM for time point X4 in sample SA494. The rest of the figures are in the appendix, section A.2. These experiments imply that the chains have converged.

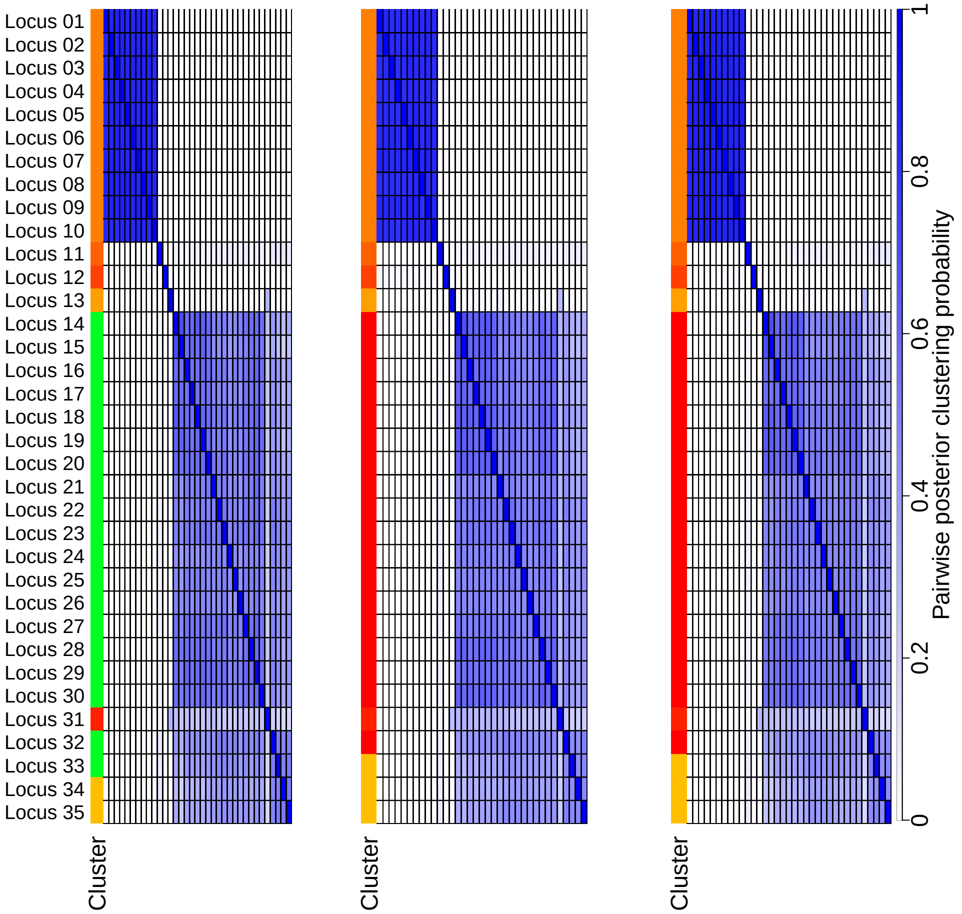


Figure 3.19: 10,000 runs from 3 different seeds over the Xenograft sample SA494 timepoint X4.

3.7 Conclusion

In this chapter, we first introduced a SNV and CNV aware genotype simulation scheme based on a phylogenetic tree, termed the Generalized Dollo model, from which we also simulated the bulk datasets.

We have shown that our method outperforms existing methods on both clustering assignment and cellular prevalence estimates in simulated datasets from the GD model. Furthermore, we have demonstrated that our method performs comparably well with existing methods in a benchmark over a real dataset. We have also shown that our method is fairly robust to the choice of hyperparameters and performs reasonably in presence of noise.

Chapter 4

Conclusion

Understanding tumour subpopulation structure is essential in understanding how tumours start, grow, and develop resistance to treatment [13]. Next generation sequencing and, more recently, single cell sequencing have been used to study this intra-tumour heterogeneity.

Our method sits at the intersection of bulk and single cell sequencing technologies. It leverages genotype co-occurrence patterns extracted from SCS to improve clustering and cellular prevalence estimates in bulk sequencing data.

4.1 Significance and contribution

In this work we introduced a novel method to incorporate single cell genotyping data with bulk sequencing data in the study of subclonal architecture. We presented a new genotype simulator, the Generalized Dollo model. It enables us to account for both copy number and single nucleotide variations while respecting the Dollo parsimony principle. Moreover, it models evolution before the occurrence of a SNV, resulting in a more realistic simulated dataset.

We have shown that our method outperforms existing methods on both clustering assignment and cellular prevalence estimates in simulated experiments. Furthermore, we have demonstrated that our method performs comparably well with existing methods in a benchmark over real datasets. We have also shown that our method is fairly robust to the choice of hyperparameters and performs reasonably in presence of noise.

Thus we have confirmed the hypothesis we posited in the introduction chapter, that is, that co-occurrence patterns from single genotyping assays, when enough genotypes have been captured, in conjunction with deep sequencing bulk data, may improve cellular prevalence estimates of genomic loci.

4.2 Limitations

We note that our assumptions regarding the clonal subpopulation in the bulk data may not agree with the input genotype matrix Δ from the single cell genotyping experiment. More specifically, we assume in modelling of the bulk data that there are only three possible genotypes per genomic locus, namely, normal, variant, and reference subpopulations. If there are fewer or greater than three genotypes present in Δ, then the two assumptions are in conflict. We note that this is not an inherent problem in the model and could be fixed in future implementations.

The current inference algorithm uses a cache-based Griddy-Gibbs method to deal with non-conjugate distributions. This may potentially impair accuracy and impose a high memory footprint.

Our experiments indicate that our method is sensitive to undersampling of genotypes. That is, performance in both clustering assignment and cellular prevalence estimation decline when some of the existing genotypes are not observed. In particular, we observed that if less than half of the genotypes are observed, our model performs worse than some of the genotype-naive methods.

4.3 Potential applications

Until single cell sequencing technology is mature enough, that is, sufficiently cost-efficient and more accurate, it could be used in conjunction with bulk sequencing data to obtain improved estimates of tumour subclonal properties.

One scenario is the longitudinal sampling of cancer patients and model systems to more accurately profile evolutionary dynamics. This is a step on the route to uncovering properties of clonal fitness that in turn are required for quantitative descriptions of phenotypic traits conferring selective advantages. We are undertaking such experiments in our labs so that computational methods to decipher the subclonal structure of heterogeneous tumours, including ours, can be directly applied in the coming years.

We note that genotype matrix inferred from single cell genotyping studies is the main intended source to compute distance between genomic loci in dd-PyClone. However, any genomic loci co-occurrence indicator data from parallel studies could be used with our model. One example is co-occurring and anti-co-occurring mutations network [5].

4.4 Future research

There are a number of ways in which our model could be improved. First, instead of a binarized genotype matrix, the original single cell genotype matrix in copy number space could be used to compute distance between genomic loci. Second, we posit that it may be possible to use the bulk sequencing data to estimate missing or noisy values in the single cell genotype matrix. Third, these could be incorporated into a phylogenetic reconstruction algorithm that jointly infers evolutionary history, genotypes, and prevalences from bulk and single cell sequencing data.

4.5 Final word

In summary, we have introduced a novel way to combine single cell genotyping and bulk sequencing data to study clonal subpopulation architecture and have shown that it outperforms existing methods in simulation studies and performs comparably in real dataset benchmarking. We hope that our method will help in understanding the evolutionary basis of cancer.

Bibliography

[1] A. V. Alekseyenko, C. J. Lee, and M. A. Suchard. Wagner and dollo: a stochastic duet by composing two parsimonious solos. *Systematic biology*, 57(5):772–784, 2008. → pages 32, 33

[2] D. M. Blei and P. I. Frazier. Distance dependent chinese restaurant processes. *The Journal of Machine Learning Research*, 12:2461–2488, 2011. → pages iii, 16, 28, 29

[3] A. Bouchard-Côté and M. I. Jordan. Evolutionary inference via the Poisson indel process. *Proceedings of the National Academy of Sciences*, 10.1073/pnas.1220450110, 2013. → pages 37

[4] K. Cibulskis, M. S. Lawrence, S. L. Carter, A. Sivachenko, D. Jaffe, C. Sougnez, S. Gabriel, M. Meyerson, E. S. Lander, and G. Getz. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature biotechnology*, 31(3):213–219, 2013. → pages 2, 3

[5] Q. Cui. A network of cancer genes with co-occurring and anti-co-occurring mutations. *PLoS One*, 5(10):e13180, 2010. → pages 70

[6] A. G. Deshwar, S. Vembu, C. K. Yung, G. H. Jang, L. Stein, and Q. Morris. Phylowgs: Reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biology*, 16(1):35, 2015. doi:10.1186/s13059-015-0602-8. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4359439/. → pages 6

[7] J. Ding, A. Bashashati, A. Roth, A. Oloumi, K. Tse, T. Zeng, G. Haffari, M. Hirst, M. A. Marra, A. Condon, et al. Feature-based classifiers for somatic mutation detection in tumour–normal paired sequencing data. *Bioinformatics*, 28(2):167–175, 2012. → pages 2

[8] P. Eirew, A. Steif, J. Khattra, G. Ha, D. Yap, H. Farahani, K. Gelmon, S. Chia, C. Mar, A. Wan, et al. Dynamics of genomic clones in breast cancer patient xenografts at single-cell resolution. *Nature*, 2014. → pages 7, 44, 45

[9] J. Felsenstein. Distance methods for inferring phylogenies: a justification. *Evolution*, pages 16–24, 1984. → pages 22

[10] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97: 611–631, 2002. → pages 32

[11] A. Fritsch, K. Ickstadt, et al. Improved criteria for clustering based on the posterior similarity matrix. *Bayesian analysis*, 4(2):367–391, 2009. → pages 32

[12] Z. Ghahramani, M. I. Jordan, and R. P. Adams. Tree-structured stick breaking for hierarchical data. In *Advances in neural information processing systems*, pages 19–27, 2010. → pages 6

[13] M. Greaves and C. C. Maley. Clonal evolution in cancer. *Nature*, 481(7381): 306–313, 01 2012. URL http://dx.doi.org/10.1038/nature10762. → pages 68

[14] G. Ha, A. Roth, J. Khattra, J. Ho, D. Yap, L. M. Prentice, N. Melnyk, A. McPherson, A. Bashashati, E. Laks, et al. Titan: inference of copy number architectures in clonal cell populations from tumor whole-genome sequence data. *Genome research*, 24(11):1881–1893, 2014. → pages 3

[15] D. G. Hert, C. P. Fredlake, and A. E. Barron. Advantages and limitations of next-generation sequencing technologies: A comparison of electrophoresis and non-electrophoresis methods. *Electrophoresis*, 29(23):4618–4626, 2008. → pages 2

[16] B. P. Hodkinson and E. A. Grice. Next-generation sequencing: a review of technologies and tools for wound microbiome research. *Advances in wound care*, 4(1):50–58, 2015. → pages 2

[17] W. Jiao, S. Vembu, A. Deshwar, L. Stein, and Q. Morris. Inferring clonal evolution of tumors from single nucleotide somatic mutations. *BMC Bioinformatics*, 15(1):35, 2014. ISSN 1471-2105. doi:10.1186/1471-2105-15-35. URL http://www.biomedcentral.com/1471-2105/15/35. → pages 6

[18] C. A. Miller, B. S. White, N. D. Dees, M. Griffith, J. S. Welch, O. L. Griffith, R. Vij, M. H. Tomasson, T. A. Graubert, M. J. Walter, et al. Sciclone: inferring clonal architecture and tracking the spatial and temporal patterns of tumor evolution. 2014. → pages 6

[19] N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito, et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90–94, 2011. → pages 7

[20] N. E. Navin. Cancer genomics: one cell at a time. *Genome Biol*, 15:452, 2014. → pages 7

[21] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000. doi:10.1080/10618600.2000.10474879. URL http://amstat.tandfonline.com/doi/abs/10.1080/10618600.2000.10474879. → pages 29

[22] R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song. Genotype and snp calling from next-generation sequencing data. *Nature Reviews Genetics*, 12 (6):443–451, 2011. → pages 2

[23] P. C. Nowell. The clonal evolution of tumor cell populations. *Science*, 194 (4260):23–28, 1976. → pages 1

[24] L. Oesper, G. Satas, and B. J. Raphael. Quantifying tumor heterogeneity in whole-genome and whole-exome sequencing data. *Bioinformatics*, 2014. doi:10.1093/bioinformatics/btu651. URL http://bioinformatics.oxfordjournals.org/content/early/2014/10/29/bioinformatics.btu651.abstract. → pages 3

[25] C. Ritter and M. A. Tanner. Facilitating the gibbs sampler: The gibbs stopper and the griddy-gibbs sampler. *Journal of the American Statistical Association*, 87(419):861–868, 1992. doi:10.1080/01621459.1992.10475289. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1992.10475289. → pages 27, 29

[26] F. Ronquist, M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Höhna, B. Larget, L. Liu, M. A. Suchard, and J. P. Huelsenbeck. Mrbayes 3.2: efficient bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*, 61(3):539–542, 2012. → pages 44

[27] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420, 2007. → pages 32

[28] A. Roth, J. Ding, R. Morin, A. Crisan, G. Ha, R. Giuliany, A. Bashashati, M. Hirst, G. Turashvili, A. Oloumi, et al. Jointsnvmix: a probabilistic model for accurate detection of somatic mutations in normal/tumour paired next-generation sequencing data. *Bioinformatics*, 28(7):907–913, 2012. → pages 2

[29] A. Roth, J. Khattra, D. Yap, A. Wan, E. Laks, J. Biele, G. Ha, S. Aparicio, A. Bouchard-Cote, and S. P. Shah. Pyclone: statistical inference of clonal population structure in cancer. *Nat Meth*, 11(4):396–398, 04 2014. URL http://dx.doi.org/10.1038/nmeth.2883. → pages 2, 4, 6, 24, 66

[30] C. T. Saunders, W. S. Wong, S. Swamy, J. Becq, L. J. Murray, and R. K. Cheetham. Strelka: accurate somatic small-variant calling from sequenced tumor–normal sample pairs. *Bioinformatics*, 28(14):1811–1817, 2012. → pages 2

[31] S. P. Shah, A. Roth, R. Goya, A. Oloumi, G. Ha, Y. Zhao, G. Turashvili, J. Ding, K. Tse, G. Haffari, et al. The clonal and mutational evolution spectrum of primary triple-negative breast cancers. *Nature*, 486(7403): 395–399, 2012. → pages 3, 14

[32] A. Shlien and D. Malkin. Copy number variations and cancer. 2009. → pages 3

[33] M. R. Stratton, P. J. Campbell, and P. A. Futreal. The cancer genome. *Nature*, 458(7239):719–724, 2009. → pages 2

[34] Y. W. Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2010. → pages 5, 16

[35] Y. Wang and N. E. Navin. Advances and applications of single-cell sequencing technologies. *Molecular cell*, 58(4):598–609, 2015. → pages 7

[36] T. A. Yap, M. Gerlinger, P. A. Futreal, L. Pusztai, and C. Swanton. Intratumor heterogeneity: seeing the wood for the trees. *Science translational medicine*, 4(127):127ps10–127ps10, 2012. → pages 1

[37] H. Zare, J. Wang, A. Hu, K. Weber, J. Smith, D. Nickerson, C. Song, D. Witten, C. A. Blau, and W. S. Noble. Inferring clonal composition from multiple sections of a breast cancer. 2014. → pages 4

Appendix A

Supporting Materials

A.1 Simulated genotypes from the GD model

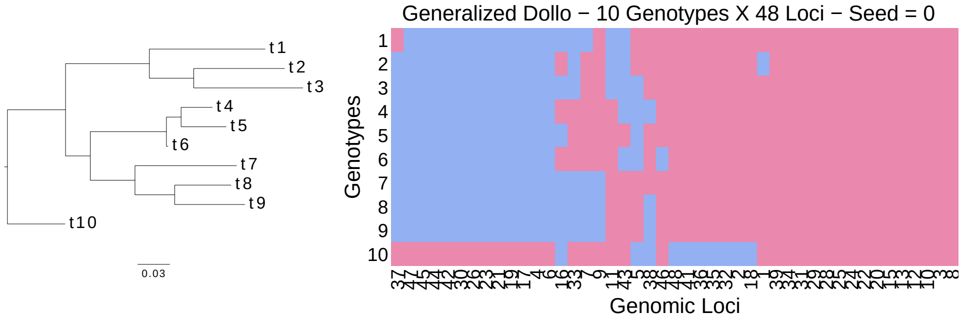


Figure A.1: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

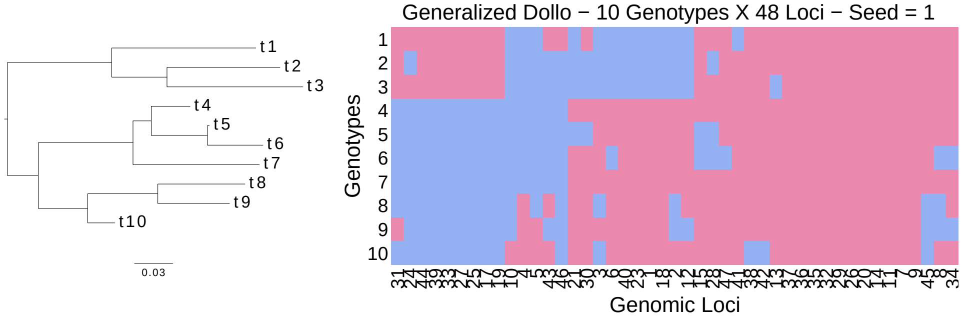


Figure A.2: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

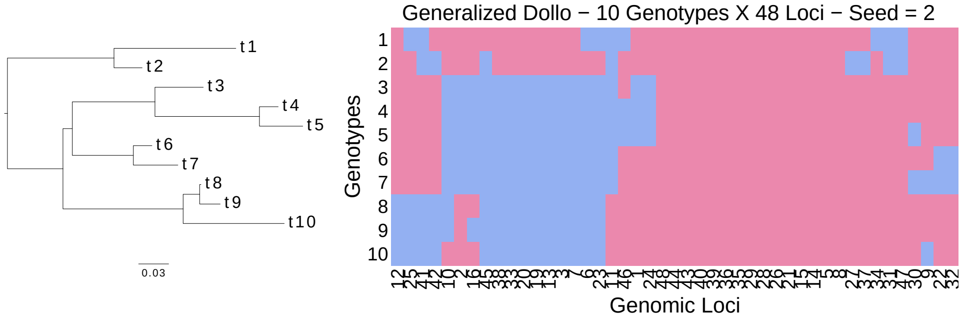


Figure A.3: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

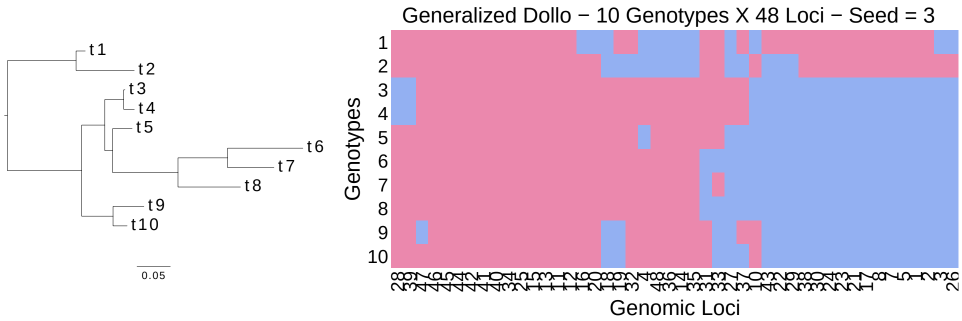


Figure A.4: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

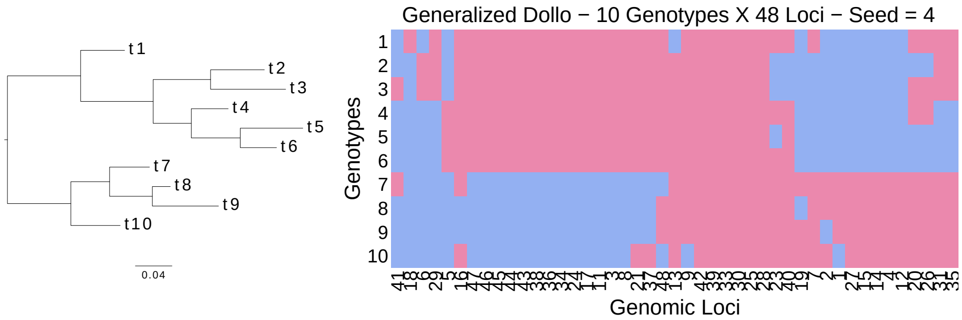


Figure A.5: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

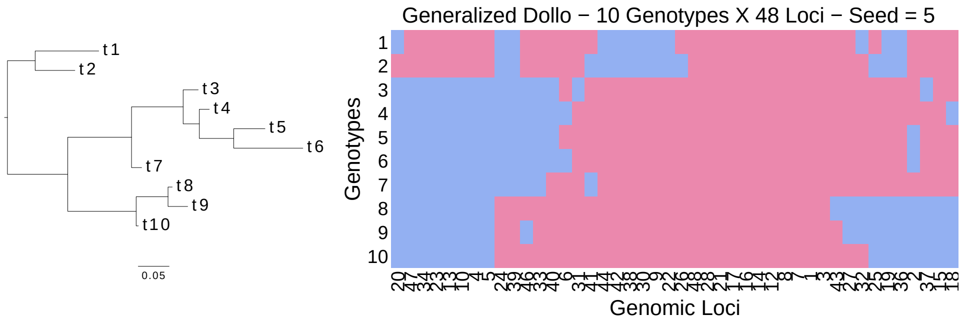


Figure A.6: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

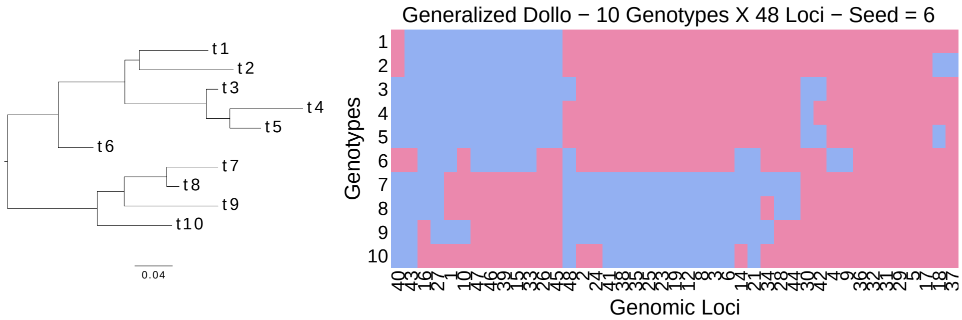


Figure A.7: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

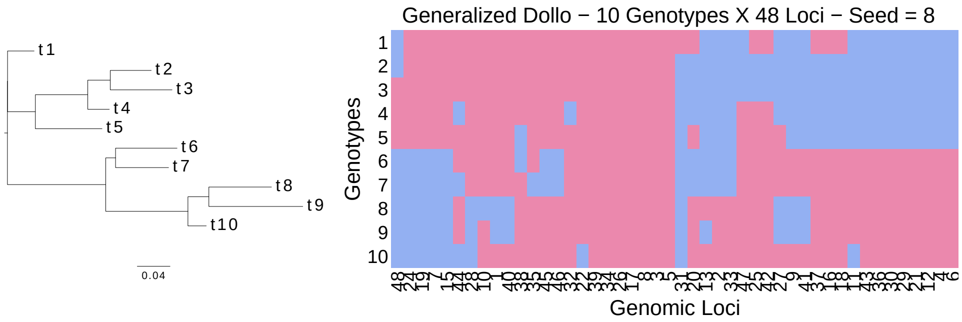


Figure A.8: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

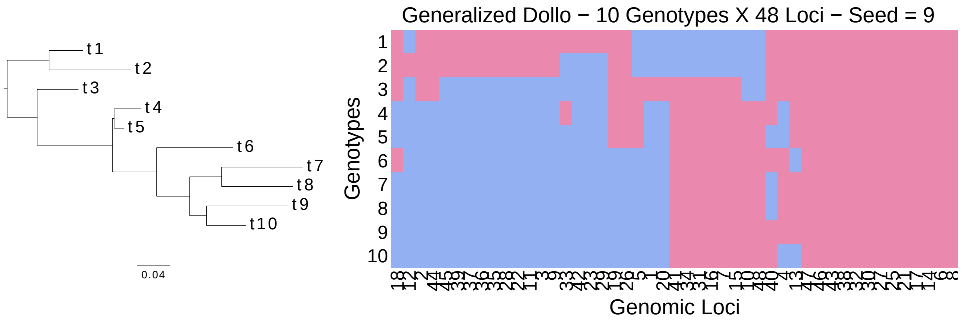


Figure A.9: Transposed binarized simulated genotypes X (right) from Generalized dollo process over a fixed phylogeny (left). The original genotype matrix X_{CN} is in copy number space. We binarize it by setting entries with non zero variant allele copy number to one (coloured red) and setting entries with variant allele copy number of zero to zero (coloured blue).

A.2 Convergence analysis results

The following figures each show PSM from 3 chains over the Xenograft TNBC real dataset.

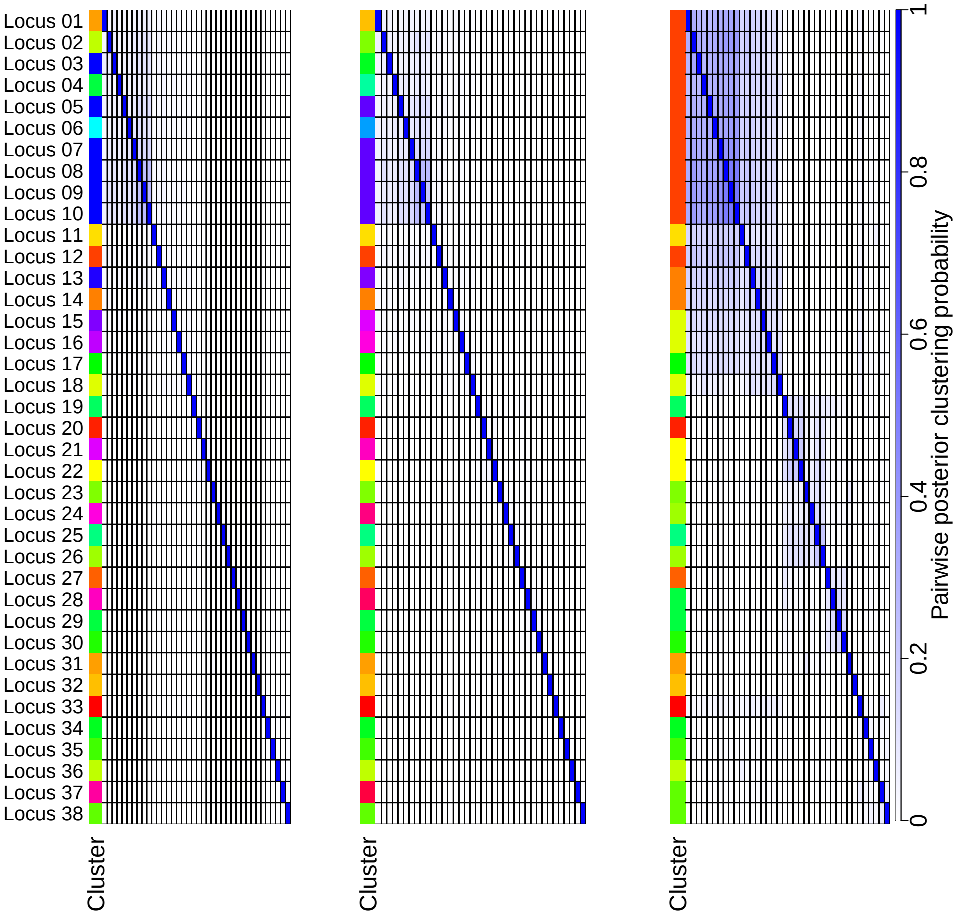


Figure A.10: 10,000 runs from 3 different seeds over the Xenograft sample SA501 timepoint X1.

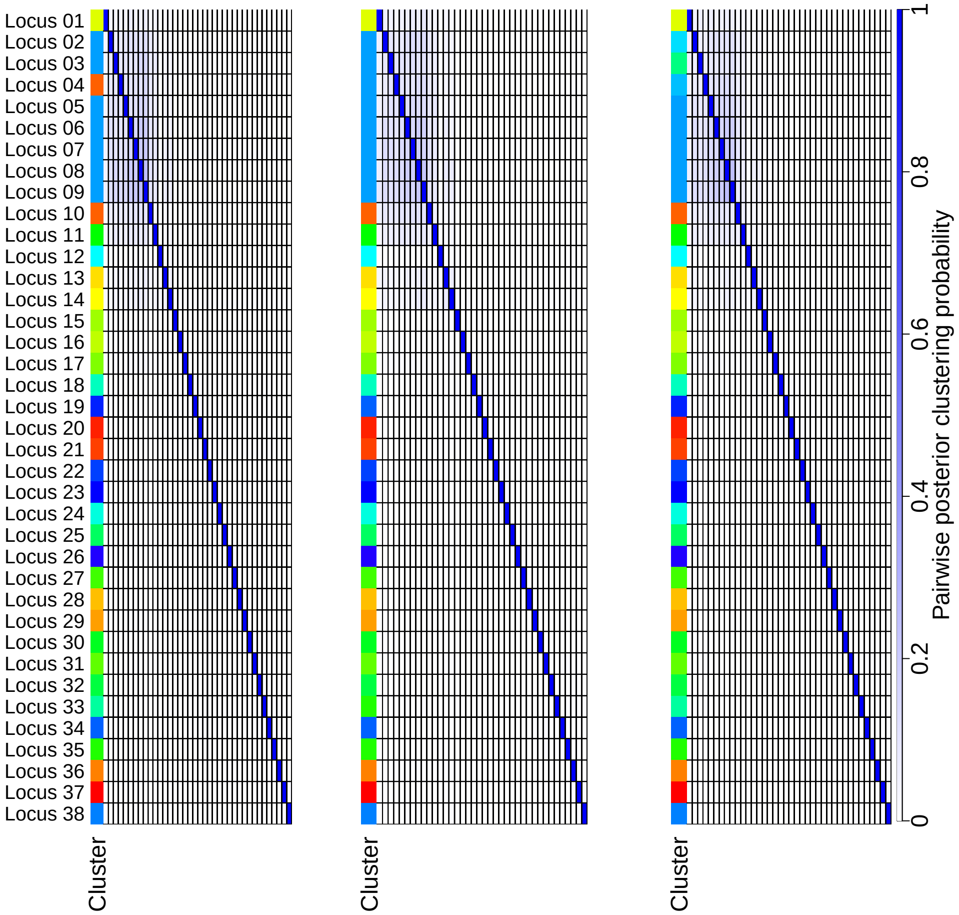


Figure A.11: 10,000 runs from 3 different seeds over the Xenograft sample SA501 timepoint X2.

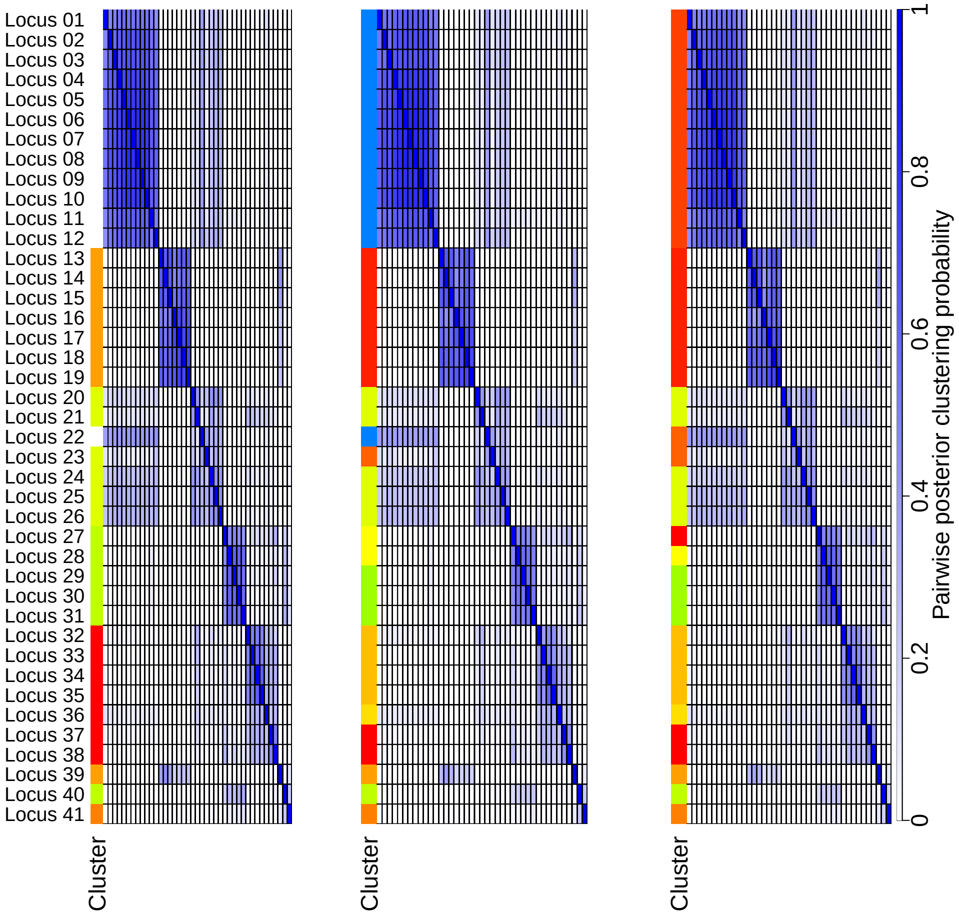


Figure A.12: 10,000 runs from 3 different seeds over the Xenograft sample SA501 timepoint X3.

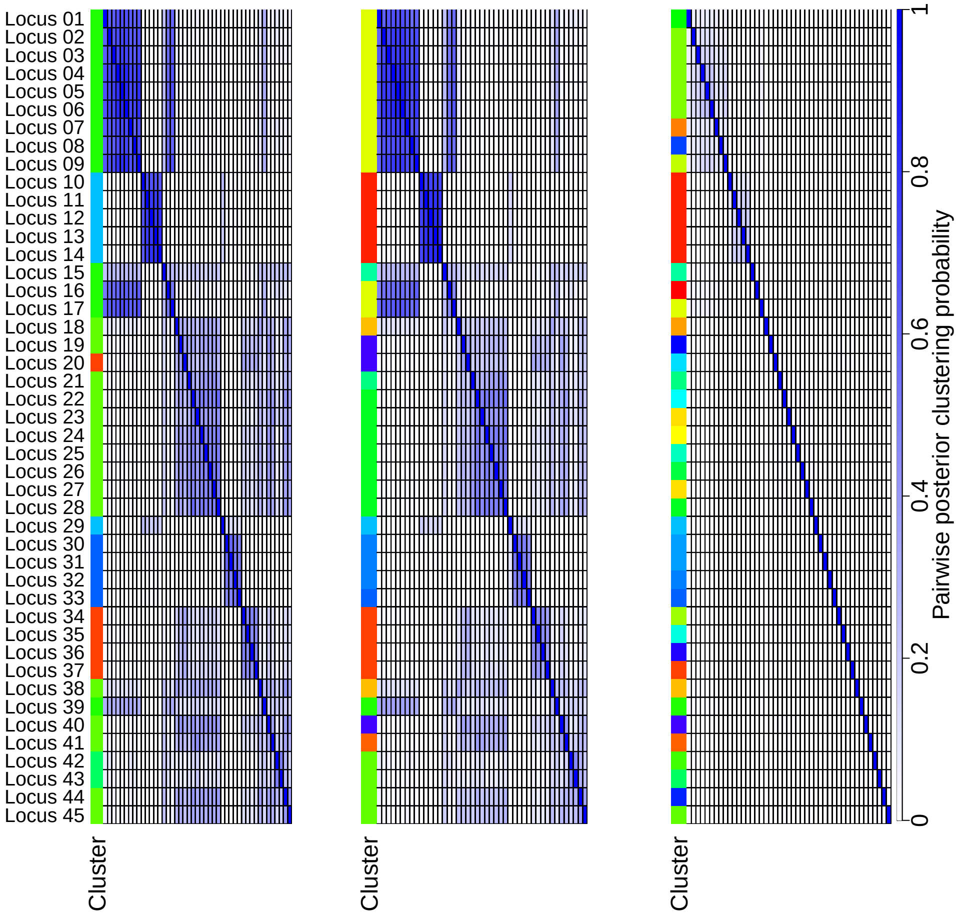


Figure A.13: 10,000 runs from 3 different seeds over the Xenograft sample SA501 timepoint X4.

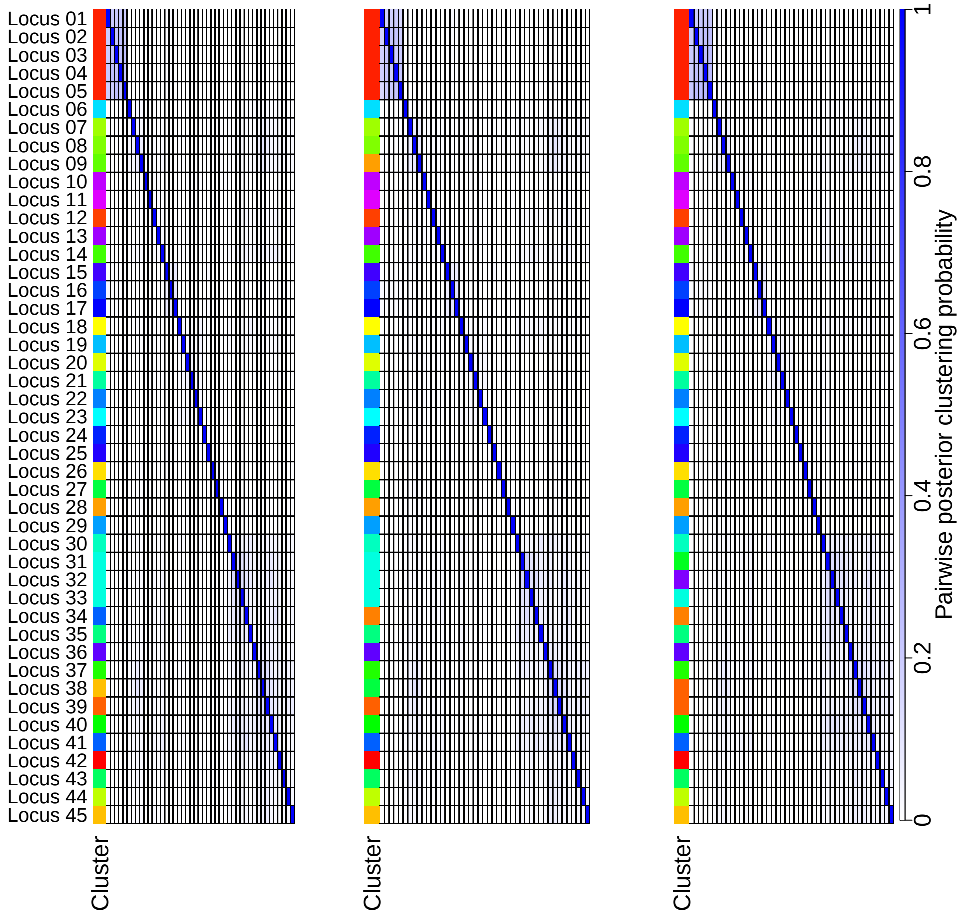


Figure A.14: 10,000 runs from 3 different seeds over the Xenograft sample SA501 timepoint X5.

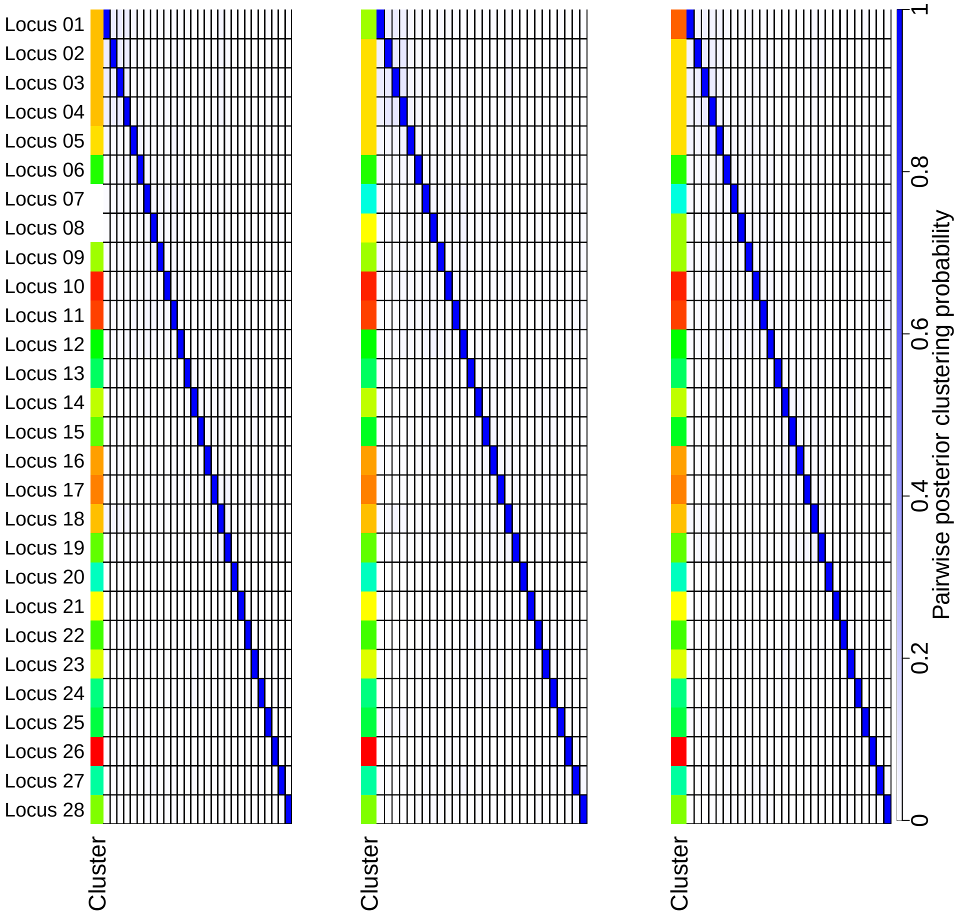


Figure A.15: 10,000 runs from 3 different seeds over the Xenograft sample SA494 timepoint T.