Efficient Bayesian inference of phylogenetic trees from large scale, low-depth genome-wide single-cell data

Fatemeh Dorri*[1], Sohrab Salehi*[2], Kevin Chern[2], Tyler Funnell[3], Marc Williams[3], Daniel Lai[4,5], Mirela Andronescu[4,5], Kieran R. Campbell[6,7], Andrew McPherson[3], Samuel Aparicio[4,5], Andrew Roth[4,5], Sohrab Shah[3], and Alexandre Bouchard-Côté[2]

[1]Department of Computer Science, University of British Columbia

[2]Department of Statistics, University of British Columbia

[3]Computational Oncology, Department of Epidemiology and Biostatistics, Memorial Sloan Kettering Cancer Center

[4]Department of Molecular Oncology, BC Cancer Research Centre

[5]Department of Pathology and Laboratory Medicine, University of British Columbia

[6]Lunenfeld-Tanenbaum Research Institute, Mount Sinai Hospital

[7]Department of Molecular Genetics, University of Toronto

May 5, 2020

Abstract

A new generation of scalable single cell whole genome sequencing (scWGS) methods [Zahn et al., 2017, Laks et al., 2019], allows unprecedented high resolution measurement of the evolutionary dynamics of cancer cells populations. Phylogenetic reconstruction is central to identifying sub-populations and distinguishing mutational processes. The ability to sequence tens of thousands of single genomes at high resolution per experiment [Laks et al., 2019] is challenging the assumptions and scalability of existing phylogenetic tree building methods and calls for tailored phylogenetic models and scalable inference algorithms. We propose a phylogenetic model and associated Bayesian inference procedure which exploits the specifics of scWGS data. A first highlight of our approach is a novel phylogenetic encoding of copy-number data providing an attractive statistical-computational trade-off by simplifying the site dependencies induced by rearrangements while still forming a sound foundation to phylogenetic inference. A second highlight is an innovative phylogenetic tree exploration move which makes the cost of MCMC iterations bounded by $O(|C|+|L|)$, where $|C|$ is the number of cells and $|L|$ is the number of loci. In contrast, existing off-the-shelf likelihood-based methods incur iteration cost of $O(|C|\ |L|)$. Moreover, the novel move considers an exponential number of neighbouring trees whereas off-the-shelf moves consider a polynomial size set of neighbours. The third highlight is a novel

*Equal contribution

mutation calling method that incorporates the copy-number data and the underlying phylogenetic tree to overcome the missing data issue. This framework allows us to realistically consider routine Bayesian phylogenetic inference at the scale of scWGS data.

1 Introduction

Cell population dynamics in cancer is a hallmark underpinning the progressive evolution of disease states and treatment resistance. Unremitting cell division is also a central feature of cancer and the most common cancers also exhibit genome instability and mutation, providing a basis for identifying cellular subpopulations. The combination of cell division and mutation naturally suggests a branching hierarchy which can be used to identify cells within a tumour related by descent from a unitary origin [Nowell, 1976, Holohan et al., 2013], as clones. Thus an important step toward characterization of intra-tumour genomic heterogeneity and measuring cell sub-population dynamics is the inference of a single cell phylogenetic tree, namely the branching structure of the ancestry of a collection of cells individually sequenced.

A feature of scWGS is variable density of genome coverage as there remains a tradeoff between the number of cells sequenced and the coverage of each genome. In DLP+ [Zahn et al., 2017] and related methods [Velazquez-Villarreal et al., 2019] entire tumour cell populations in the range 1000–10 000 or more cells are sequenced, each cell measured in a genome-wide but low coverage fashion ($\ll$1x). Low coverage limits the ability to infer phylogenetic tree based on point mutations, the information used by the majority of phylogenetic inference methods. Copy number alterations, a prevalent feature of many cancers where the number of copies of a locus takes value smaller or larger than that of normal cells [Zack et al., 2013], provide a promising alternative in this context, as it allows us to bin several loci to gain statistical strength.

In this work, we introduce novel computational methods for performing Bayesian phylogenetic tree inference from single cell data composed of a large number of cells with shallow coverage. Our method is based on lossy transformation of single cell copy number matrices retaining only presence or absence of changes in copy number profiles. This transformation turns a complex evolutionary process (integer-valued copy numbers, prone to a high degree of homoplasy and dense dependence structure across sites), into a simpler one which can be well approximated by a probabilistic version of a perfect phylogeny. Since we want to obtain not only a tree but also an assessment of the uncertainty over various parts of the tree, our approach is grounded in the Bayesian paradigm, as often done in the phylogenetic inference literature [Lemey et al., 2009]. We leverage the special structure created by the change point transformation to build a special purpose Markov chain Monte Carlo (MCMC) kernel which has better computational scalability per move compared to classical phylogenetic kernels. To make this statement more precise, let $|C|$ denote the number of cells, and $|L|$ the number of locations in the genome having at least one change point. We show that we can explore an exponential number of neighbours using a running time of $|C| + |L|$. In contrast, previous tree exploration methods cost $|C| \cdot |L|$ to explore a much smaller (polynomial) number of neighbours. To put things in perspective, for the typical regime we consider, $|C| = 1000s$

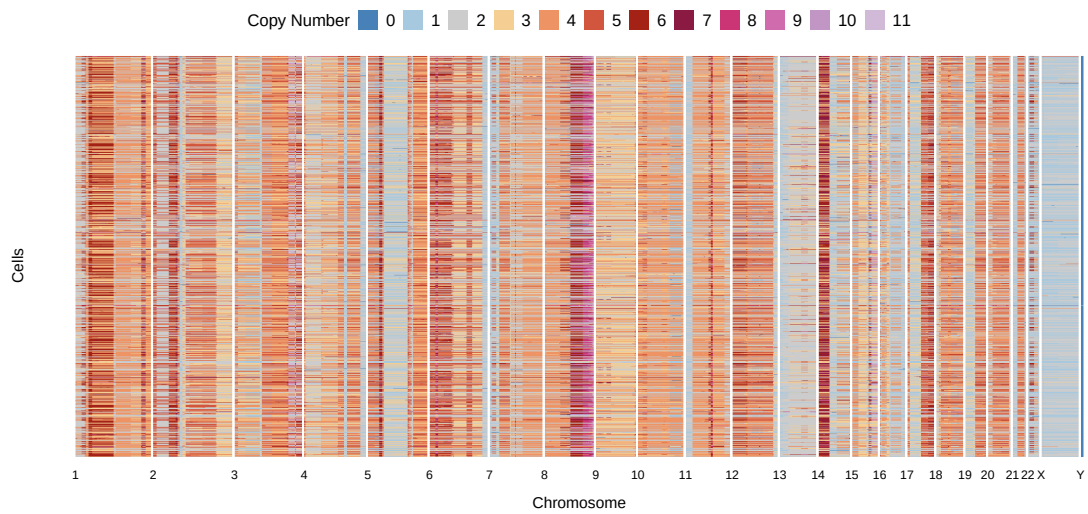

Figure 1: Example of input data used in the phylogenetic method described in this work: a copy number matrix for the HGS OV data described in Section 6. Rows index individually sequenced cells and columns index genomic loci. The colours show the integer copy number estimates taken as input in our phylogenetic method.

and $|L| = 100s$, so $|C| + |L| \approx 1000$ whereas $|C| \cdot |L| \approx 100\,000$.

The input data for our method can be visualized in a colour-coded matrix exemplified in Figure 1. Each row in the matrix corresponds to an individual cell that has been sequenced in a single-cell platform. Each column in the matrix is a locus, which in Figure 1 is a "bin," namely a contiguous set of genomic positions. We assume that the integer copy number of each bin has been called as a pre-processing step, e.g. using a hidden Markov model tool such as in Zahn et al. [2017]. In Figure 1 the copy number call is encoded by the colour of each entry in the matrix.

The output of our method includes a discrete tree where individual cells are identified by vertices at the leaves of the tree, and edges correspond to estimated placement of copy number change events. Such output is shown in Figure 2. We visualize the input and estimated tree simultaneously by sorting the cells (rows) in such a way as they line up with the position of the corresponding leaves of the tree.

The paper is organized as follows. In Section 2, we review related work on phylogenetic methods for cancer data, in Section 3, we introduce our probability model, in Section 4, the posterior inference method we developed. Then we describe in Section 5 an additional application in which copy number information is used to aid placement of highly sparse single nucleotide variation data. In Section 6 we present numerical examples on synthetic and real data, including triple negative breast patient derived xenograft samples and high grade serous ovarian primary and matched relapse samples.

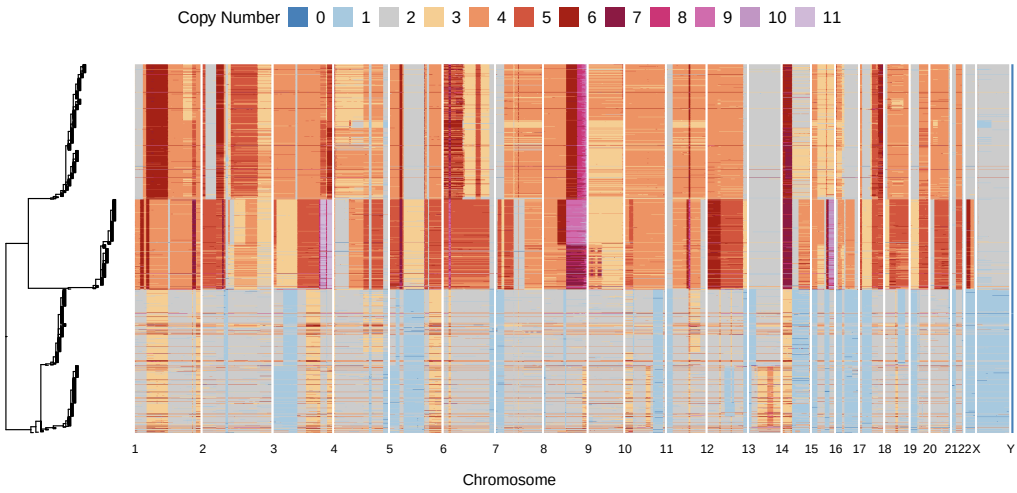

Figure 2: Same copy number matrix as Figure 1 but sorted by the reconstructed phylogenetic tree.

2 Literature review

Several methods, including Bayesian methods, have been proposed for reconstructing phylogenetic trees from single cell cancer data, see reviews in Schwartz and Schäffer [2017], Kuipers et al. [2017]. However the majority of existing single cell tree inference methods focus on point mutations. The new generation of single cell methods described in the previous section requires a shift of attention toward scalable methods based on copy number variation.

2.1 Single cell methods based on point mutations

Methods developed for the first generation of single cell data includes the work of Jahn et al. [2016], Singer et al. [2018], where single-cell Bayesian and maximum likelihood inference is considered, based on point mutation rather than copy number variation. A related method has also been proposed in Ross and Markowetz [2016], also for point mutations and this time based on the maximum likelihood framework only. These studies consider perfect phylogenetic models related to ours, however computation is performed via traditional phylogenetic moves, as a consequence each tree operation costs $O(|C||L|)$ and considers a polynomial number of neighbour trees. Since the datasets analyzed in these studies involve in the order of 50 cells, the traditional phylogenetic moves are reasonable. However for scWGS, the tree exploration innovations introduced here, which allow moves of cost $O(|C| + |L|)$ and exponential number of neighbours, are critical.

An alternative model was proposed in Zafar et al. [2016a] for building single cell trees, but again from point mutations. This method is based on maximum likelihood instead of Bayes (but using an unconventional MCMC-based search procedure). In order to model losses, the authors use a conventional continuous time Markov chain model with a parameterization tailored to the problem. Such an approach has the inconvenience of allowing unrealistic evolutionary paths such as

an arbitrary high number of independent gains at the same locus. Tree moves are also conventional and cost $O(|C||L|)$ per move.

Distance based methods such as neighbour joining have been used in numerous case studies for the purpose of building single-cell trees, see for example Xu et al. [2012]. Distance based methods are a useful heuristic, but they have been long known in the field of phylogenetics to be statistically sub-optimal relative to likelihood based methods [Williams and Moret, 2003]. Another method based on pairwise modelling but more closely tailored to single cell data is Kim and Simon [2014].

2.2 Bulk methods based on copy number variation

There is a large literature on the reconstruction of chromosome rearrangement histories. However, most methods performing detailed reconstruction from sequencing data assume monoclonal bulk data, for example Letouzé et al. [2010], McPherson et al. [2012], Greenman et al. [2012], Li et al. [2016]. More recent work relaxes the monoclonal assumption but still in a bulk data context [Mahmoody et al., 2012, McPherson et al., 2017]. Other methods [Schwarz et al., 2014] use several bulk samples from the same patient, but assume that each is monoclonal.

There are several difficulties involved in translating these methods into a phylogenetic model suitable to scWGS data. First, at any given breakpoint, pair-end reads, which many of the above methods use, are only expected to be found for a very sparse collection of cells because of the very low read depth per cell. Second, the combinatorial problem involved in detailed rearrangement tracking appears to be difficult to reconcile with efficient search of large single cell trees.

2.3 Single cell methods based on copy number variation

There is comparatively less work on single cell phylogenetics based on copy number alterations. One exception is the literature on phylogenetic tree construction based on FISH data, however the methods developed there do not scale beyond the very small number of evolutionary markers found in FISH data [Chowdhury et al., 2013, 2014].

In Zahn et al. [2017], a posterior distribution over single cell trees is inferred using MrBayes [Huelsenbeck and Ronquist, 2001] by treating inferred cell-specific copy numbers as discrete traits. The limitations of this way of building phylogenetic traits from the data are described in the next section.

3 Model construction

3.1 Phylogenetic traits for single cell genomics

Phylogenetic methods are based on defining certain traits or markers on observed operational taxonomic units, which are individual cells in our context. In this section, we discuss the important issue of construction of traits from copy number variation data.

We start by describing two different ways to encode copy number variation data, and then argue that the second one presents important computational advantages. The two encodings in question are as follows.

Integer copy numbers: where the traits are indexed by genomic bins and the traits are integer-valued (or possibly integer-pair-valued if phasing is incorporated). We avoid this encoding for reasons described in the Section 3.1.1.

Binary change points: where the traits are indexed by "spaces" between two contiguous bins, and traits are binary. The trait between bin i and $i+1$ is defined informally as whether there is a copy number change between bin i and $i+1$, which we call a *change point indicator variable*. We opt for this construction for reasons described in Section 3.1.2. We use "change point" rather than "breakpoint" to emphasize that we do not require measurements such as pair end reads.

We do not argue that using rearrangements as phylogenetic traits is novel: this observation has a long history in the cancer phylogenetics, briefly reviewed in Section 2.2. Instead, what is novel is to use *only* the change point indicator variables while discarding copy numbers. From a statistical point of view, this is suboptimal: the theoretical foundation for "full Bayesian inference," namely the Bayes estimator, prescribes conditioning on the full data y, and this makes Bayes estimators *efficient* in the formal *statistical* sense of the term [C. Robert, 2007]. However, the Bayes estimator can be *computationally* very inefficient. In particular, full Bayesian phylogenetic reconstruction based on a detailed account of chromosomal rearrangements seems highly challenging with current computational and mathematical tools.

This tension between computational and statistical efficiency suggests that there is fertile grounds for trade-offs between the two, and indeed these trade-offs have been instrumental to many recent advances in other areas of computational statistics [Chandrasekaran and Jordan, 2013]. Here we approach this trade-off by computing Bayes estimators conditioning on a non-sufficient statistic of the data, $S(y)$. This is related to Approximate Bayesian Computation (ABC) [Tavare et al., 1997], however in ABC this idea is compound with other approximations in the posterior inference step which requires the non-sufficient statistics to be low-dimensional (called "summary statistics" in the ABC literature, typically containing in the order of 10s or less components). In contrast, here we design $S(y)$ so that posterior simulation methods can be used which support data with large numbers of components.

3.1.1 Avoiding integer encodings

Given an unlimited computational budget, again the integer encoding is the full data we would like to model. However, the evolution of copy numbers has a very dense and complex dependence structure. Most neighbour bins are nearly identical replicates. Even after "chunking" into partitions of constant copy, overlapping events create further dependencies. For example, ploidity change events induce dependencies across *all* traits, and hence formidable computational inference challenges.

For a detailed explanation of the drawbacks of this encoding, see Schwartz and Schäffer [2017]. The issue is also identified in Beerenwinkel et al. [2015].

We do not use integer copy number traits in this work but we hope the tree built using our method may form the basis for inference of more sophisticated models which may be able to use the full integer-value copy number data.

3.1.2 Proposed alternative: binary change points

Let us fix some notation to make the binary change point traits more precise. Assume the genome has been segmented into contiguous regions called bins, and that in any given bin, the copy number is constant in all cells and all bins (violation of this assumption is expected; next section describe how to increase robustness to such violation).

With a slight abuse of terminology, we define a locus l as an inter-nucleotide position between two contiguous bins. Let $n^-_{c,l}, n^+_{c,l}$ denote the integer-valued copy number in the bins right before and after locus l in cell c.

Let C and L denote the disjoint sets of cells and loci respectively. We define the trait $x_{c,l}$ for cell $c \in C$ at locus $l \in L$ as follows:

$$x_{c,l} = \mathbf{1}[n^-_{c,l} \neq n^+_{c,l}] = \begin{cases} 1, & \text{if } n^-_{c,l} \neq n^+_{c,l}, \\ 0, & \text{otherwise.} \end{cases}$$

Using this construction, we obtain that in the majority of the distinct evolutionary histories formed by overlapping pairs of events, at most two sets of traits can be dependent (the two end points of the region amplification/loss). This observation is described in more detail in Section 3.3.3. This simpler form of dependence is easier to mitigate, compared to the dense dependencies of the integer encoding.

3.2 Observation model

The binary change point traits as defined in the previous section are not directly observed as they depend on the unobserved copy number variables $n^-_{c,l}, n^+_{c,l}$.

Therefore, we posit an observation probability model $p(y|x, \theta)$, where x is the matrix $x = (x_{c,l})$, y is the data or a data summary, and θ are model parameters.

There is considerable flexibility on how to construct the observation model. For concreteness, we assume the following form to start with: assume that as a preprocessing step, we have performed copy number inference on each individual cell. Let $\hat{n}^-_{c,l}, \hat{n}^+_{c,l}$ denote the copy number calls for the bins right before and after locus l in cell c. Now define $y_{c,l} = \mathbf{1}[\hat{n}^-_{c,l} \neq \hat{n}^+_{c,l}]$ which can be viewed as a noisy version of $x_{c,l}$. We show an example of y for a real dataset encoded this way in Figure 3.

To model errors in copy number calls, we introduce false positive and negative rate parameters

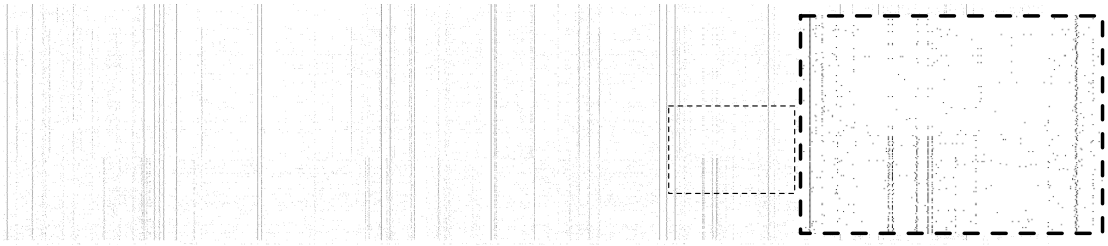

Figure 3: An example of a matrix $y_{c,l}$ (SA 535) obtained by performing HMM-based calls on DLP data.

$r^{\mathrm{FP}} \in (0,1)$ and $r^{\mathrm{FN}} \in (0,1)$ respectively, and an error matrix

$$e^{r^{\mathrm{FP}},r^{\mathrm{FN}}} = \begin{bmatrix} 1-r^{\mathrm{FP}} & r^{\mathrm{FP}} \\ r^{\mathrm{FN}} & 1-r^{\mathrm{FN}} \end{bmatrix},$$
$$p\left(y_{c,l}|x_{c,l}, r^{\mathrm{FP}}, r^{\mathrm{FN}}\right) = e^{r^{\mathrm{FP}},r^{\mathrm{FN}}}_{x_{c,l},y_{c,l}}.$$

from which we set:

$$p(y|x,\theta) = \prod_{l\in L}\prod_{c\in C} p\left(y_{c,l}|x_{c,l}, r^{\mathrm{FP}}_{c,l}(\theta), r^{\mathrm{FN}}_{c,l}(\theta)\right).$$

The functions $r^{\mathrm{FP}}_{c,l}(\theta)$ and $r^{\mathrm{FN}}_{c,l}(\theta)$ denote potentially cell and locus specific error rates. The simplest choice is to make these parameters constant across cell and loci, $\theta_{\mathrm{global}} = (r^{\mathrm{FN}}_{\mathrm{global}}, r^{\mathrm{FN}}_{\mathrm{global}})$, $r^{\mathrm{FP}}_{c,l}(\theta_{\mathrm{global}}) = r^{\mathrm{FP}}_{\mathrm{global}}$ and $r^{\mathrm{FN}}_{c,l}(\theta_{\mathrm{global}}) = r^{\mathrm{FN}}_{\mathrm{global}}$, which we call the global error parameterization, but we discuss a more sophisticated version shortly.

3.3 Probability model for the latent variables $x_{c,l}$

3.3.1 Foundations: perfect phylogenies

A perfect phylogeny is a venerable model which assume that each trait emerges at exactly one point in a rooted phylogenetic tree, and that all the descendants of that point in the tree have the trait (i.e. there is no loss).

In and of itself, the perfect phylogeny model is not suitable for analyzing real data, as it is brittle: a single erroneous character or violation of the assumption can have high impact on the inferred tree.

However, coupled with a carefully constructed observation model, we can make the perfect phylogeny construction more robust. A key modelling choice is to assume the perfect phylogeny assumption not on the observed traits $y_{c,l}$, but rather on their latent counterparts $x_{c,l}$.

3.3.2 Priors on trees

We build a simple probabilistic model with support on the set of all perfect phylogenetic trees. This is done using the following two-step generative process.

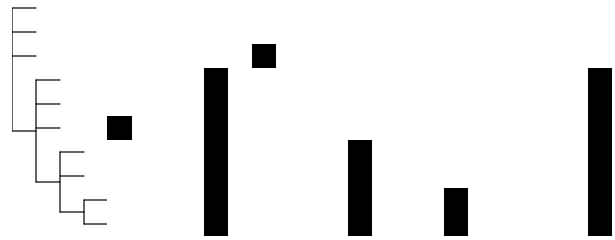

Figure 4: Visualization of a small tree $t \in \mathcal{T}$. The right side shows the matrix of binary latent values $x_{c,l}$ where rows are cells and columns are loci. The left side shows a summary of the perfect phylogeny. To remove clutter, we remove from the tree (but not the matrix) all loci where $x_{c,l} = 0$ for all cells. We also collapse chains (list of edges with exactly one descendant) into a single edge. Note that all clade containing more than one descendant cells is necessarily supported by one or more traits l having value $x_{c,l} = 1$ if and only if c is a descendant of the clade. Clades at the leaves, which have size one may or may not be supported.

Sampling a mutation tree: let $\mathcal{V}^{\mathrm{m}} = L \cup \{v^*\}$ denote a vertex set composed of one vertex for each of the $|L|$ loci plus one artificial root node v^*. The artificial root node induces an implicit notion of direction on the edges, viewing them as pointing away from v^*. Let $\mathcal{T}^{\mathrm{m}}$ denote the set of trees t^{m} spanning $\mathcal{V}^{\mathrm{m}}$. The interpretation of t^{m} is as follows: there is a directed path from vertex/locus l to l' in t^{m} if and only if the trait indexed by l is hypothesized to have emerged in a cell which is ancestral to the cell in which l' emerged.

Sampling cell assignments: assign each cell to a vertex in t^{m}. The interpretation of assigning cell c to locus l is that among the traits under study, c is hypothesized to possess only the traits visited by the shorted path from v^* to l in t_m. If a cell c is assigned to v^*, the interpretation is that c is hypothesized to possess none of the traits under study.

Both steps can be viewed as graphs (for the second step, it is a bipartite graph with one component being the set of loci, and the other, the set of cells). It is convenient to summarize both processes at the union of the two graphs, which is also a tree, this time on $|L| + |C| + 1$ vertices, $\mathcal{V} = L \cup C \cup \{v^*\}$. Let us denote the set of trees obtained by this two step process by $\mathcal{T}$. Again, the vertex v^* induces an implicit direction to edges in $t \in \mathcal{T}$.

Given $t \in \mathcal{T}$, the matrix x is a deterministic function obtained by setting $x_{c,l} = 1$ if vertex c is a descendant of vertex l in t, and zero otherwise. We denote this deterministic function by $x(t)$. We show a small example in Figure 4.

The tree structure encoded by $\mathcal{T}$ is unidentifiable: for example, if a tree contains a chain of traits, then permutations of the traits yield the same matrix $x(t)$. Since we are taking a Bayesian approach this is not problematic. As we describe later, tree summaries we build from the posterior distribution collapse the non-identifiable parts. Moreover, the specific choice we make for constructing the set $\mathcal{T}$ help us obtain simple and efficient sampling algorithms.

To complete the construction of the prior on trees, we need to assign probabilities to elements

of $\mathcal{T}$. For simplicity, we take the uniform distribution on this finite set. We have:

$$\begin{aligned}|\mathcal{T}| &= |\mathcal{T}^{\mathrm{m}}||\{f : C \to L \cup \{v^*\}\}| \\ &= (|L|+1)^{(|L|+1)-2}(|L|+1)^{|C|} \\ &= (|L|+1)^{|L|+|C|-1},\end{aligned}$$

where we used Cayley's formula to compute $|\mathcal{T}^{\mathrm{m}}|$, hence we set:

$$p(t) = \frac{\mathbf{1}[t \in \mathcal{T}]}{(|L|+1)^{|L|+|C|-1}}.$$

This simple prior has a useful property: if a collection of say two splits are supported by m_1 and m_2 traits, then the prior probability for an additional trait to support the first versus second split is proportional to (m_1+1, m_2+1). Therefore, there is a "rich gets richer" behaviour built-in into the prior, which is viewed as useful in many Bayesian non-parametric models. Of course, more complicated priors over $\mathcal{T}$ could be easily incorporated as the complexity of inference typically comes from the likelihood rather than the prior.

Simulation from the prior can be performed using Wilson's algorithm [Wilson, 1996], which yields exact uniform samples from $\mathcal{T}^{\mathrm{m}}$, followed by independent categorical sampling to simulate the cell assignments.

3.3.3 Robustness to violations of the perfect phylogenetic assumption

It is informative to look at the different types of copy number alteration evolutionary paths and their implication on the perfect phylogeny assumption. When two events affect disjoint genomic regions, then clearly the perfect phylogeny assumption holds on the latent variables $x_{c,l}$ (although not necessarily so on the noisy observations $y_{c,l}$, which the global error parameterization of Section 3.2 can handle). This means that by selecting at most one locus by chromosome, we can be guaranteed that there will be no violation of the perfect phylogenetic assumption on the $x_{c,l}$. However we would like to use the full data.

We show in Figure 5 the interesting cases where two events are overlapping and their effect on the perfect phylogeny assumption on $x_{c,l}$. Only when there are gains followed by an overlapping loss, or loss followed by overlapping loss can there be violations. Even then, the probability of the violation is given by the inverse of the copy number before the second event. So it is reasonable to assume these violations will represent a minority of the traits.

Moreover, notice that for all cases except for those violating the perfect phylogeny assumption, at most two loci are dependent, namely the two end points of the event.

Heuristics could be constructed to detect potential violation as a pre-processing step, for example cases such as Figure 5(D) leave an easy-to-detect Loss of Heterozygosity (LOH) signature. However, it is simpler and more principled to include all traits, and instead ensure that the inferred tree will be robust to perfect phylogeny violations.

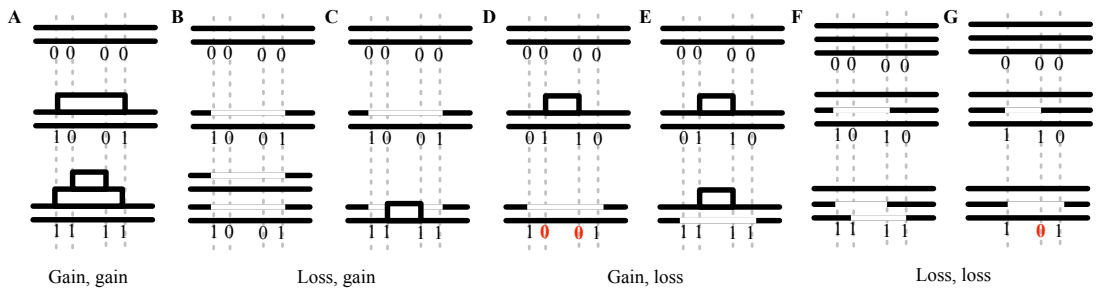

Figure 5: Overlapping copy number alteration events and their effect on the perfect phylogeny assumption. We only show overlapping scenarios as non-overlapping ones systematically respect the perfect phylogeny assumption. Each panel shows an evolutionary path from an ancestral cell (top) to intermediate (middle) to a child cell (bottom). Panel **A** shows overlapping copy number gains. By the standard infinite site argument (the number of potential end-points is large relative to the number of events), the end-points of the events have high probability to be all distinct. In all such cases, the perfect phylogeny assumption is respected as the trait variables $z_{c,l} = 0$ transition to $z_{c,l} = 1$, but not vice versa. **B, C**: a loss followed by a gain follows the same argument. For example, note in **B** that a ploidity change in fact leaves the change-point characters invariant. **D, E**: for gains followed by a loss, whether the perfect phylogeny assumption holds depends on whether the loss erases the amplification (**D**) or not (**E**). **F, G**: similarly, for a loss followed by another loss whether the perfect phylogeny assumption holds depends on whether the second loss erases an end point of the first loss.

To do so, we set the error rates to be locus-dependent: $\theta = (r_1^{\text{FP}}, r_2^{\text{FP}}, \ldots, r_{|L|}^{\text{FP}}, r_1^{\text{FN}}, r_2^{\text{FN}}, \ldots, r_{|L|}^{\text{FN}})$, $r_{c,l}^{\text{FP}}(\theta) = r_l^{\text{FP}}$ and $r_{c,l}^{\text{FN}}(\theta_{\text{global}}) = r_l^{\text{FN}}$. With this extra flexibility, the model can discount the effect of a trait which violates the tree structure imposed by a majority, by setting high error rates for the trait's locus.

The number of parameters under the locus-dependent parameterization, $2|L|$ can be relatively large, however this does not cause difficulties given we are taking a Bayesian approach to parameter estimation. Since there are many cells, many of these parameters concentrate in practise, but not all, e.g. those for traits placed very high or very low in the tree, hence providing justification for a Bayesian approach.

It would be easy to add batch or cell-specific error parameters, however we leave this to future work.

3.4 Priors on the parameters θ

Care must be taken when placing prior distributions on θ, denoted $p(\theta)$. We observed that the default choice of putting a uniform on all error rates leads to hard to interpret posterior distributions. The reason is that allowing error rates to be close or equal to one can lead to undesirable results. Consider for example a perfect phylogenetic tree with a topology given by a full, balanced tree with four leaves. Suppose there is one event observed in a noise-less fashion identifying the left clade, and one noiseless event identifying the right clade. Then the reconstruction we want would

set the error rates to zero. However, there is another reconstruction, which is undesirable yet has the same probability: set error rates to one, and permutes the two trait assignments.

To avoid such pathology, we use priors of the form

$$r_l^{\mathrm{FP}} \sim \mathrm{Uniform}\left(0, \overline{r^{\mathrm{FP}}}\right),$$
$$r_l^{\mathrm{FN}} \sim \mathrm{Uniform}\left(0, \overline{r^{\mathrm{FN}}}\right),$$

where we recommend to take the upper bounds lower or equal to 1/2. Moreover, since there are more mechanisms that can lead to false negative compared to false positive, we recommend to take $\overline{r^{\mathrm{FP}}} < \overline{r^{\mathrm{FN}}}$: indeed, both types of errors can be caused by incorrect HMM calls, but false negatives can additionally be used to model cases in Figure 5D,G breaking the perfect phylogeny assumption. We use $\overline{r^{\mathrm{FP}}} = 1/10$ and $\overline{r^{\mathrm{FN}}} = 1/2$ as default in our experiments.

4 Inference

Both Bayesian and maximum likelihood methods require exploring the space of trees efficiently. To do so, we develop in Section 4.1 a move that explores an exponentially large number of neighbours in time $O(|C| + |L|)$. After establishing this result, we describe how this move can be used to construct a full Bayesian inference pipeline, both for posterior sampling and minimization of the Bayes risk objective.

4.1 Efficient tree exploration

Suppose we have a current hypothesized tree $t \in \mathcal{T}$, or equivalently, an imputed latent matrix x. The move introduced in this section explores a set of neighbour trees $t' \in N^l(t) \subset \mathcal{T}$ (equivalently, $x' \in N^l(x)$). The set of neighbours is indexed by $l \in L$. At a high level, the trees t' in the neighbourhood can be described as follows. First, pick a locus l, and remove the corresponding column from x. Second, consider all possible ways to reintroduce the removed column while re-assigning binary values to its entries in a way allowed by the other columns (i.e., re-assignments x' such that there exists a perfect phylogeny t' such that $t' = t(x')$).

We now describe how to sample or maximize with respect to the probability of all the trees in $N^l(t)$. First, it is useful to describe the operations of erasing one column and adding it back in terms of tree operations. We show the interpretation in Figure 6 in terms of edge contraction for deleting a column, and edge insertion for adding it back.

From this figure, we see that the removal of a column l yields a tree $t_{\setminus l}$. Suppose from now on that the locus l being resampled is fixed. Adding the column l back while choosing an allowed reassignment can be done in three steps:

1. Select any node v in $t_{\setminus l}$ which is not a cell (i.e., either the root or one of the remaining loci, a set of candidates denoted $R = \{v^*\} \cup L \setminus \{l\}$).

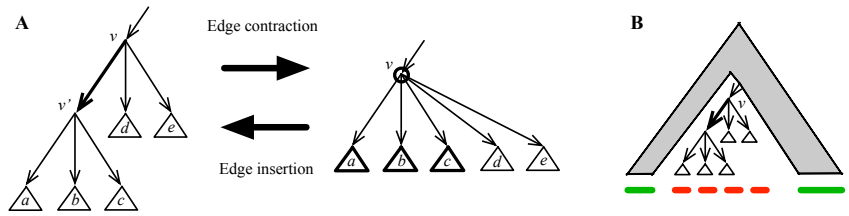

Figure 6: **A**: Reading from left to right: the interpretation of removing a column in the matrix x is to perform contraction of an edge corresponding to a locus shown in bold. Reading from right to left: the interpretation of inserting back a column while assigning new binary values is an edge insertion. The circled node v refers to Step 1. The subtrees in bold refer to those selected in Step 2. The edge in bold, the one introduced in Step 3. **B**: Decomposition used for the recursion of Section 4.1.

2. Pick any subset (possibly empty) from the children subtrees of v in $t_{\setminus l}$.
3. Add a new vertex v' corresponding to the locus being reintroduced, and an edge from v to v'. Move the subset of children subtrees selected in Step 2 to hang from v' instead of v.

We will follow this three step process when sampling or maximizing. We partition $N^l(t_{\setminus l})$ into block corresponding to the choice of node v made in Step 1, $N^l(t_{\setminus l}) = \cup_v N^l_v(t_{\setminus l})$. Our first objective is to compute efficiently the following probabilities

$$\bar{\rho}_v = \frac{\rho_v}{\sum_{\tilde{v} \in R} \rho_{\tilde{v}}},$$

where:

$$\rho_v = \sum_{t \in N^l_v(t_{\setminus l})} p(t) p(y|x(t), \theta). \tag{1}$$

We start by computing the following recursion for all vertex v in $t_{\setminus l}$: first, for all vertices c corresponding to a cell and $b \in \{0, 1\}$, define

$$p^b_c = p\left(y_{c,l} | b, r^{\mathrm{FP}}_{c,l}(\theta), r^{\mathrm{FN}}_{c,l}(\theta)\right),$$

which can be interpreted as "local likelihood" terms. Alternatively, the same algorithm can be used initialized instead with

$$\tilde{p}^b_c = \frac{p^b_c}{p^0_c + p^1_c}, \tag{2}$$

which can be interpreted as a "local" posterior distributions for $x_{c,l}$ based on a flat prior and the information of $y_{c,l}$ only. Using either initialization $\tilde{p}^b_c$ or initialization p^b_c yields exactly the same distribution because the normalization in Equation 2 is constant as a function of b. We use $\tilde{p}^b_c$ in implementation because it cuts storage by a factor two (we store only $\tilde{p}^1_c$ and compute $\tilde{p}^0_c = 1 - \tilde{p}^1_c$

on the fly). However later when resampling θ the normalization needs to be taken into account. For simplicity in the remaining derivations we stick with p_c^b.

Next, we lift this to local posterior distributions for all subtrees of $t_{\setminus l}$. This can be done efficiently via a bottom-up recursion on the tree $t_{\setminus l}$: for all $v \in R$, $b \in \{0, 1\}$,

$$p_v^b = \prod_{v'' \in \text{children}(v)} p_{v''}^b,$$

where children(v) denotes the list of children of vertex v.

We can now return to the problem of computing $\bar{\rho}_v$. First, observe that the sum in Equation (1) can be re-indexed by a bit vector $b = (b_1, b_2, \ldots, b_k)$, $b_{v''} \in \{0, 1\}$ of length equal to $k = |\text{children}(v)|$. Each bit $b_{v''}$ is equal to one if children v'' is to be moved into a child of v' (refer to Figure 6), and zero if it to stay as a child of v. For each possible assignment, we obtain at tree $t \in N_v^l(t_{\setminus l})$, and its probability can be decomposed into factors corresponding to cells that are descendant of v (denoted C_v, solid red thick line under the tree of Figure 6B) and those that are not (denoted $C_{\setminus v}$, dashed green thick line under the tree of Figure 6B).

The product of the likelihood contributions for those that are not descendant does not depend on the choice of bit vector. It can be obtained by taking the product of all exclusion local posteriors and removing those that are descendant of v:

$$\prod_{c \in C_{\setminus v}} p_c^0 = \frac{p_{v^*}^0}{p_v^0}.$$

Note that this assumes $p_v^0 > 0$. As a workaround to cases where there are structural zeros, we recommend injecting small numerical values if $p_v^0 = 0$ (we used 10^{-6} in our implementation).

For the cells under v, we now have to take into account whether they are selected under the newly introduced locus or not. More precisely, for each of the children $v_1, v_2, \ldots, v_k$, we have to take into account the value of the bit vector $b = (b_1, b_2, \ldots, b_k)$. The sum over possible assignments written naively has a number of terms which is exponential in k, but can be rewritten into a k factors:

$$\begin{aligned}\sum_{t \in N_v^l(t_{\setminus l})} \prod_{c \in C_v} p_c^{x_{c,l}(t)} &= \sum_{b_1=0}^{1} \cdots \sum_{b_k=0}^{1} \prod_{i=1}^{k} p_{v_i}^{b_i} \\ &= \prod_{i=1}^{k} (p_{v_i}^0 + p_{v_i}^1).\end{aligned}$$

Putting these results together, we obtain, for some constants K_i that do not depend on v:

$$
\begin{aligned}
\rho_v &= K_1 \sum_{t \in N_v^l(t_{\setminus l})} p(y|x(t),\theta) \\
&= K_1 \sum_{t \in N_v^l(t_{\setminus l})} \prod_{l' \in L} \prod_{c \in C} p\left(y_{c,l'}|x_{c,l'}(t), r_{c,l'}^{\mathrm{FP}}(\theta), r_{c,l'}^{\mathrm{FN}}(\theta)\right) \\
&= K_1 \left(\prod_{l' \in L, l' \neq l} \prod_{c \in C} p\left(y_{c,l'}|x_{c,l'}(t), r_{c,l'}^{\mathrm{FP}}(\theta), r_{c,l'}^{\mathrm{FN}}(\theta)\right) \right) \sum_{t \in N_v^l(t_{\setminus l})} \prod_{c \in C} p\left(y_{c,l}|x_{c,l}(t), r_{c,l}^{\mathrm{FP}}(\theta), r_{c,l}^{\mathrm{FN}}(\theta)\right) \\
&= K_1 K_2 \sum_{t \in N_v^l(t_{\setminus l})} \prod_{c \in C} p\left(y_{c,l}|x_{c,l}(t), r_{c,l}^{\mathrm{FP}}(\theta), r_{c,l}^{\mathrm{FN}}(\theta)\right) \\
&= K_1 K_2 \sum_{t \in N_v^l(t_{\setminus l})} \prod_{c \in C} p_c^{x_{c,l}(t)} \\
&= K_1 K_2 \sum_{t \in N_v^l(t_{\setminus l})} \left(\prod_{c \in C_v} p_c^{x_{c,l}(t)} \right) \left(\prod_{c \in C_{\setminus v}} p_c^{x_{c,l}(t)} \right) \\
&= K_1 K_2 \left(\prod_{c \in C_{\setminus v}} p_c^{x_{c,l}(t)} \right) \sum_{t \in N_v^l(t_{\setminus l})} \prod_{c \in C_v} p_c^{x_{c,l}(t)} \\
&= K_1 K_2 \left(\frac{p_{v^*}^0}{p_v^0} \right) \sum_{t \in N_v^l(t_{\setminus l})} \prod_{c \in C_v} p_c^{x_{c,l}(t)} \\
&= K_1 K_2 \left(\frac{p_{v^*}^0}{p_v^0} \right) \prod_{i=1}^{k} (p_{v_i}^0 + p_{v_i}^1) \\
&= K_1 K_2 K_3 \frac{\prod_{i=1}^{k} (p_{v_i}^0 + p_{v_i}^1)}{p_v^0}.
\end{aligned}
$$

We can therefore compute the probabilities $\bar{\rho}_v$ in Step 1 via:

$$
\bar{\rho}_v = \frac{\rho_v}{\sum_{\tilde{v} \in R} \rho_{\tilde{v}}} \tag{3}
$$

$$
= \frac{\left(\frac{\prod_{v_i \in \mathrm{children}(v)} \left(p_{v_i}^0 + p_{v_i}^1\right)}{p_v^0} \right)}{\sum_{\tilde{v} \in R} \left(\frac{\prod_{v_i' \in \mathrm{children}(\tilde{v})} \left(p_{v_i'}^0 + p_{v_i'}^1\right)}{p_{\tilde{v}}^0} \right)}. \tag{4}
$$

Once an attachment node v is selected, sampling or maximizing over Step 2, i.e. picking the children to move into children of v', is straightforward. This can be done by considering independent Bernoulli variables, one for each current child v_i of v with success probability

$$
\frac{p_{v_i}^1}{p_{v_i}^0 + p_{v_i}^1},
$$

where a success corresponds to moving the child v_i of v into a child of the newly re-introduced vertex/locus v'. The realization of these Bernoulli variables can be trivially maximized or sampled from.

4.2 Parameter estimation

An effective approach to resample the parameters θ is to do so while conditioning on x, i.e. in a Metropolis-within-Gibbs fashion. Conditioning on x allows us to compute and cache sufficient statistics on x; once this cache is computed, updates on components of θ can be performed in time complexity constant in L and C.

The structure of the cache depends on the parameterization of the likelihood $p(y|x,\theta)$. For the global parameterization $\theta_{\text{global}} = (r^{\text{FN}}_{\text{global}}, r^{\text{FN}}_{\text{global}})$ introduced in Section 3.2, we summarize the matrix x with two integers, the number of false positive instances, n^{FP}, and the number of false negative instances, n^{FN},

$$
\begin{aligned}
n^{\text{FP}} = n^{\text{FP}}(x) &= \sum_{c\in C}\sum_{l\in L} \mathbf{1}[x_{c,l} = 0, y_{c,l} = 1] \\
n^{\text{FN}} = n^{\text{FN}}(x) &= \sum_{c\in C}\sum_{l\in L} \mathbf{1}[x_{c,l} = 1, y_{c,l} = 0].
\end{aligned}
$$

Based on these cached statistics, we obtain:

$$
p(y|x,\theta_{\text{global}}) = \left(r^{\text{FP}}\right)^{n^{\text{FP}}} \left(r^{\text{FN}}\right)^{n^{\text{FN}}} \left(1 - r^{\text{FP}}\right)^{n^{\text{N}} - n^{\text{FN}}} \left(1 - r^{\text{FN}}\right)^{n^{\text{P}} - n^{\text{FP}}}, \tag{5}
$$

where the the number of positive n^{P} and negative n^{N} instances in the data can be pre-computed,

$$
\begin{aligned}
n^{\text{P}} &= \sum_{c\in C}\sum_{l\in L} \mathbf{1}[y_{c,l} = 1] \\
n^{\text{N}} &= |C||L| - n^{\text{P}}.
\end{aligned}
$$

For the locus-specific parameterization of Section 3.3.3, we simply need to compute the statistics for each locus l:

$$
\begin{aligned}
n^{\text{FP}}_l = n^{\text{FP}}_l(x) &= \sum_{c\in C} \mathbf{1}[x_{c,l} = 0, y_{c,l} = 1] \\
n^{\text{FN}}_l = n^{\text{FN}}_l(x) &= \sum_{c\in C} \mathbf{1}[x_{c,l} = 1, y_{c,l} = 0] \\
n^{\text{P}}_l &= \sum_{c\in C} \mathbf{1}[y_{c,l} = 1] \\
n^{\text{N}}_l &= |C| - n^{\text{P}}_l \\
p(y|x,\theta) &= \prod_l \left(r^{\text{FP}}_l\right)^{n^{\text{FP}}_l} \left(r^{\text{FN}}_l\right)^{n^{\text{FN}}_l} \left(1 - r^{\text{FP}}_l\right)^{n^{\text{N}}_l - n^{\text{FN}}_l} \left(1 - r^{\text{FN}}_l\right)^{n^{\text{P}}_l - n^{\text{FP}}_l}.
\end{aligned}
$$

Once this caching mechanism is setup, the details of the mechanism used to update θ are not critical. For uniform or beta priors, conjugate updates are available. For simplicity we use a slice sampling algorithm which does not make any assumption on the priors, namely the "doubling procedure" of Neal [2003].

4.3 Probabilistic programming implementation

The model is implemented in the Blang probabilistic programming language Bouchard-Côté et al. [2019]. Three main variants of the model are implemented. *FixedMatrixModel*,which assumes a fixed matrix $\tilde{p}_c^b$ of local posterior distribution is pre-computed and fixed, as well as *NoisyBinaryModel*, which comes in two variant, namely the global parameterization of Section 3.2 and the locus-specific parameterization of Section 3.3.3.

4.4 Greedy inference

Before moving to more sophisticated full Bayesian methods, we first consider a simple baseline algorithm which approximately maximizes $p(y|x,\theta)$ for fixed θ. It will be useful to describe this baseline, greedy algorithm in a slightly more general context: suppose we are given distributions of the form $q_{c,l}(x_{c,l})$ for each cell c and locus l. The algorithm approximate

$$\begin{aligned} M(q) &= \operatorname{argmax}\{q(t) : t \in \mathcal{T}\}, \\ q(t) &= q(x(t)) \\ q(x) &= \prod_{c \in C} \prod_{l \in L(x)} q_{c,l}(x_{c,l}). \end{aligned}$$

To recover the case of maximum likelihood for fixed θ, set $q_{c,l}(x_{c,l}) = p(y_{c,l}|x_{c,l},\theta)$. Notice we use a slight generalization of the set of loci, $L(x)$ instead L as we will consider in this section matrices built with a subset of the columns, then $L(x)$ is the set of columns/loci in x.

The greedy algorithms starts with the star tree rooted at r^* with leaves C, and greedily adds locus one at the time. The order at which loci are inserted has an effect on the result of this procedure. For example, if a locus is highly uncertain and added early in the process, the sub-optimal choice made might prevent later in the algorithm correct identification of a clade clearly supported in the data. Conversely, by delaying uncertain decision, there might be more valid constraints imposed by traits correctly placed in the tree early in the algorithm.

In order to attempt to visit the loci in a desirable order, we maintain a queue sorted by priorities given by:

$$\text{priority}(l) = \max_{t' \in N^l(t)} \frac{q(t')}{\sum_{t'' \in N^l(t)} q(t'')}.$$

The maximum and the sum can be computed using the algorithm of Section 4.1 in its maximization flavour. In the case where $q_{c,l}(x_{c,l}) = p(y_{c,l}|x_{c,l},\theta)$, the quantity being maximized is the posterior probability of one column x_l given the columns inserted so far $\tilde{x}$, i.e. $p(x_l|t(\tilde{x}),\theta)$.

The priorities change after each locus insertion, however to avoid a cost of $|L|^2$, we only recompute and resort the queue at a pre-specified period.

Once a locus l is specified the new tree t' is obtained from the previous one t via

$$t' = \text{argmax}_{t'' \in N^l(t)} \, q(t'').$$

Again, this maximization problem is approach using the algorithm of Section 4.1.

4.5 Approximation of the posterior distribution

We seek to approximate the posterior distribution

$$\pi(t, \theta) \propto p(t)p(\theta)p(y|x(t), \theta).$$

We consider three methods to approximate the posterior distributionn:

1. A direct Markov chain Monte Carlo (MCMC) scheme.
2. A non-reversible Parallel Tempering (PT) algorithm [Syed et al., 2019].
3. The sequential change of measure (SCM) scheme of Del Moral et al. [2006] with the adaptive scheme of Zhou et al. [2013].

Schemes 2 and 3 require a sequence of annealed distributions. We use:

$$\pi_\beta(t, \theta) \propto p(t)p(\theta) \left(p(y|x(t), \theta)\right)^\beta,$$

for $\beta \in [0, 1]$, called the inverse temperature parameter. Note that we recover the posterior when $\beta = 1$ and the prior when $\beta = 0$. We can extract exact independent samples from the prior. For $\beta > 0$, we explore the space with π_β invariant kernels, obtained by composing kernels $K_l^{\text{tree},\beta}$ modifying the tree and kernels $K^{\text{param},\beta}$ modifying parameters.

The kernels modifying trees are indexed by $l \in L$ and are defined as

$$K_l^{\text{tree},\beta}((t, \theta), (t', \, \mathrm{d}\theta')) \propto \mathbf{1}[t' \in N_l(t)]\delta_\theta(\, \mathrm{d}\theta') \left(p(y|x(t), \theta)\right)^\beta.$$

Sampling from $K_l^{\text{tree},\beta}$ can be performed using the algorithm of Section 4.1 by replacing p_c^b by $(p_c^b)^\beta$ in the initialization step and using the same recursions otherwise.

The kernels modifying parameters are indexed by $d \in \{1, \ldots, \dim(\theta)\}$ and are slice sampling kernels modifying each of the $\dim(\theta)$ component of the random vector θ. The method of Section 4.2 can be modified to accommodate the annealing parameter β by replacing Equation (5) by

$$(p(y|x, \theta_{\text{global}}))^\beta = \left(r^{\text{FP}}\right)^{\beta n^{\text{FP}}} \left(r^{\text{FN}}\right)^{\beta n^{\text{FN}}} \left(1 - r^{\text{FP}}\right)^{\beta(n^{\text{N}} - n^{\text{FN}})} \left(1 - r^{\text{FN}}\right)^{\beta(n^{\text{P}} - n^{\text{FP}})}.$$

The $|L| + \dim(\theta)$ kernels are combined via deterministic alternation followed by random reshuffling once the alteration completes a cycle. Since the kernels $K^{\text{tree},\beta}$ cost $O(|L| + |C|)$ to compute

while the kernels $K^{\text{param},\beta}$ cost $O(1)$, we do a full cycle of all $K^{\text{param},\beta}$ kernels for $d \in \{1, \ldots, \dim(\theta)\}$ in between each call of one the tree kernels $K^{\text{tree},\beta}$.

We consider two initialization: (i) the star tree, and (ii) a tree sampled uniformly from the prior. Both initializations can be used for the MCMC inference method. For the PT and SCM methods, initialization (ii) has to be used in order to fit into the sequence of distributions we selected.

4.6 Summary via minimum Bayes risk estimator

In order to summarize the posterior distribution, we use the minimum Bayes risk framework. Based on a loss function $L(t, t')$, Bayesian analysis prescribes to use the estimator:

$$\text{argmin}_{t \in \mathcal{T}} \sum_{t' \in \mathcal{T}} \int L(t, t') \pi(t, \, \mathrm{d}\theta). \tag{6}$$

A natural and computationally agreeable choice for the loss is the L1 metric on the matrices of induced indicators $x(t)$:

$$L(t, t') = \sum_{l \in L} \sum_{c \in C} |x_{c,l}(t) - x_{c,l}(t')|.$$

With this choice, we can use linearity to rewrite the objective function of Equation (6) into:

$$\begin{aligned}
\sum_{t' \in \mathcal{T}} \int L(t, t') \pi(t, \, \mathrm{d}\theta) &= \sum_{t' \in \mathcal{T}} \int \sum_{l \in L} \sum_{c \in C} |x_{c,l}(t) - x_{c,l}(t')| \pi(t, \, \mathrm{d}\theta) \\
&= \sum_{l \in L} \sum_{c \in C} \sum_{t' \in \mathcal{T}} \int |x_{c,l}(t) - x_{c,l}(t')| \pi(t, \, \mathrm{d}\theta) \\
&= \sum_{l \in L} \sum_{c \in C} \{ m_{c,l}(1 - x_{c,l}(t)) + (1 - m_{c,l}) x_{c,l}(t) \} \\
&= \sum_{l \in L} \sum_{c \in C} \{ x_{c,l}(t) - 2 m_{c,l} x_{c,l}(t) \} + \text{constant},
\end{aligned} \tag{7}$$

where the marginal indicators $m_{c,l}$ are defined by:

$$m_{c,l} = \sum_{t \in \mathcal{T}} \int \mathbf{1}[x_{c,l}(t) = 1] \pi(t, \, \mathrm{d}\theta),$$

and can therefore be approximated from a Monte Carlo approximation $t^1, t^2, \ldots, t^N$ by averaging the indicators:

$$\frac{1}{N} \sum_{i=1}^{N} x_{c,l}(t^i) \to m_{c,l} \text{ a.s.}$$

Once the averages have been computed from the Monte Carlo samples, we can cast the problem of minimizing Equation (6) into the problem solved by the greedy procedure of Section 4.4, which

equivalently can be written as

$$\begin{aligned} M(q) &= \operatorname{argmax}\left\{\log q(t) : t \in \mathcal{T}\right\}, \\ \log q(t) &= \log q(x(t)) \\ \log q(x) &= \sum_{c \in C} \sum_{l \in L(x)} \log q_{c,l}(x_{c,l}). \end{aligned} \tag{8}$$

To get the require minimization program in Equation (7) to coincide with the objective targeted by the greedy reformulation of Equation (8), we therefore need

$$\log q_{c,l}(x_{c,l}) = -\left\{x_{c,l} - 2m_{c,l}x_{c,l}\right\},$$

hence we decode using $M(q^{\text{mbr}})$ where:

$$q^{\text{mbr}}_{c,l}(x_{c,l}) = \exp(2m_{c,l}x_{c,l} - x_{c,l}). \tag{9}$$

4.7 Visualization of consensus trees

To visualize the consensus tree, we collapse the chains (sequence of loci having only one children) as well as remove the subtrees containing no cells. We align the leaves of the tree which correspond to cells after collapsing to the rows of a cell-locus matrices (see Figure 12, left, for an example described in more detail in the next section). The columns of the cell-locus matrix are loci, grouped by blocks of columns. Each column in a block belongs to a series of matrices; depending on whether real or synthetic data is used the nature of these matrices depends and is explained in the next section. In both cases however, the first column in each group always shows (in black and white) the traits supporting the consensus tree t^*, namely the matrix $x(t^*)$; and the last column, the posterior probability of each entry of $x_{c,l}$. The middle column is used either to display the reference tree in synthetic experiments, or the input data, in real data experiments.

5 Application: single nucleotide variants calling

A phylogenetic tree over single cells provides a principled way to pull statistical strength across groups of cells sequenced at very low coverage. To exemplify this, we show how our framework can be used for single nucleotide variants(SNVs) calling from low coverage genome wide single cell data such as those generated by platforms such as DLP or 10x genomics.

There is a large literature on single nucleotide variants calling, see e.g. Zafar et al. [2016b], Dong et al. [2017], Hård et al. [2019], Bohrson et al. [2019]. The closest related work is Singer et al. [2018], which also uses a phylogenetic tree to improve variant calling. However Singer et al. [2018] operates on different types of data being purely based on single nucleotide data, whereas the approach we propose in this section uses both copy number (CNV) information and single nucleotide variants. For the low coverage data we consider, the low coverage per cell implies that copy number

information is critical for phylogenetic inference, even if the end goal is single nucleotide variant calling.

We consider two use cases: first, joint tree inference based on both SNV and CNV, and second, post-hoc addition of SNVs to an existing CNV-based tree. We find that the first use case, joint inference, would require a coverage higher than what is currently achieved by the datasets at hand, but that the second use case performs well in the same datasets.

5.1 Incorporating single nucleotide data

In this section, we posit an observation probability model for single nucleotide variant data.

We denote the observed single nucleotide data by $y_{c,l}^{SNV} = (\nu_{c,l}, d_{c,l}, c_{c,l})$ where $d_{c,l}$ denotes the total number of reads covering the locus l in cell c, $\nu_{c,l}$, the number of reads with a variant allele (compared to the reference genome), and $c_{c,l}$, the corresponding copy number.

We use $x_{c,l}^{SNV}$ to denote an indicator variable taking the value one if and only if an ancestor of cell c harboured a single nucleotide alteration event at locus l. This variable is unobserved and the focus of inference in this section. As before, we assume a perfect phylogeny structure on these indicators, but add to this an error model to relate $x_{c,l}^{SNV}$ to the observed data while allowing violations of the perfect phylogeny assumption as well as measurement errors. In the context of single nucleotide data, this is similar to Jahn et al. [2016]. The parameters of the error model are denoted $\theta^{SNV} = (\epsilon_{FP}, \epsilon_{FN})$, where ϵ_{FP} and ϵ_{FN} are false positive rate and false negative rates, respectively. Therefore

$$q_{c,l}^{b} = p(y_{c,l}^{SNV} | x_{c,l}^{SNV}, \theta^{SNV}) = p(\nu_{c,l} | d_{c,l}, c_{c,l}, x_{c,l}^{SNV} = b, \theta^{SNV}), \tag{10}$$

where $d_{c,l}$ and $c_{c,l}$ are given. The likelihood probability of cell node c is denoted by $q_{c,l}^{b}$, where $b \in \{0, 1\}$. For $b = 1$, $q_{c,l}^{b}$ reflects the likelihood of cell c being mutated at locus l; and for $b = 0$, $q_{c,l}^{b}$ reflects the likelihood of cell c not being mutated at locus l.

The probability $q_{c,l}^{b}$ follows a mixture of binomial distributions depending on all possible genotype states of locus l at cell c. Given the copy number $c_{c,l}$, the possible genotype states are $\boldsymbol{\mathcal{G}} = \{A \ldots A, AA \ldots B, A \ldots BB, \ldots, B \ldots B\}$, where each element has a length equal to $c_{c,l}$. For example, the genotype AAB refers to a genotype with one variant allele B and 2 reference allele A. For each genotype state g_i, where i indexes the elements of $\boldsymbol{\mathcal{G}}$, the mean parameter of the corresponding binomial distribution is denoted by $\xi_{c,l}^{i}$:

$$\xi_{c,l}^{i} = \begin{cases} \frac{\mathcal{B}(g_i)}{c_{c,l}}, & 1 \leq \mathcal{B}(g_i) < c_{c,l}, \\ 1 - \epsilon_{FP}, & \mathcal{B}(g_i) = c_{c,l}, \\ \epsilon_{FP}, & \text{otherwise}, \end{cases} \tag{11}$$

where $\mathcal{B}(g_i)$ represents the number of variant allele of genotype g_i. Therefore, for $b = 1$,

$$q^1_{c,l} = p(\nu_{c,l}|d_{c,l}, c_{c,l}, x^{SNV}_{c,l} = 1, \theta^{SNV}) \tag{12}$$

$$= \sum_{i=1}^{c_{c,l}} p(g_i)[\xi^{\nu_{c,l}}_{c,l}(1-\xi_{c,l})^{d_{c,l}-\nu_{c,l}}] \tag{13}$$

$$+ \epsilon_{FN}[\epsilon^{\nu_{c,l}}_{FP}(1-\epsilon_{FP})^{d_{c,l}-\nu_{c,l}}].$$

The value of $p(g_i)$ equals $\frac{1-\epsilon_{FN}}{c_{c,l}}$, and ϵ_{FN} represents the error due to mutation loss or tree errors.

If the mutation status of cell c at locus l is a wildtype (i.e mutation is not present), then the possible genotype states should not have any variant allele. The only possible genotype state is $\{A \dots A\}$. The mean parameter of the binomial distribution equals ϵ_{FP} (false positive rate). Therefore,

$$q^0_{c,l} = p(\nu_{c,l}|d_{c,l}, c_{c,l}, x^{SNV}_{c,l} = 0, \epsilon_{FP}). \tag{14}$$

Algorithm 5.2 summarizes the assumed generative model where $g_{c,l}$ is the true genotype of locus l of cell c.

With the proposed probability model for SNVs, we can incorporate both SNV data and CNV data to infer the underlying tree phylogeny in the Corrupt model. Therefore,

$$p(y|x,\theta) = \prod_{c\in M}\prod_{l\in N_{CNV}} p(y^{CNV}_{c,l}|x^{CNV}_{c,l}, \theta^{CNV}) \prod_{l\in N_{SNV}} p(y^{SNV}_{c,l}|x^{SNV}_{c,l}, \theta^{SNV}), \tag{15}$$

where M and L is the disjoint set of cells and loci, respectively. L includes both CNV and SNV traits, and $L = L_{CNV} + L_{SNV}$. Therefore, Equation (4) (when locus l is removed from the tree) can be rewritten in the following form:

$$\bar{\rho}_v = \frac{\left(\frac{\prod_{v_i\in \text{children}(v)} (\gamma^0_{v_i}+\gamma^1_{v_i})}{\gamma^0_v}\right)}{\sum_{\bar{v}\in R}\left(\frac{\prod_{\bar{v}_i\in \text{children}(\bar{v})} (\gamma^0_{\bar{v}_i}+\gamma^1_{\bar{v}_i})}{\gamma^0_{\bar{v}}}\right)}, \tag{16}$$

where γ^b_v, for $b \in \{0,1\}$ is:

$$\gamma^b_v = \begin{cases} p^b_v, & \text{If } l \text{ represents a CNV loci,} \\ q^b_v, & \text{If } l \text{ represents a SNV loci.} \end{cases}$$

For $v \in R = \{v^*\} \bigcup L \backslash \{l\}$, and $b \in \{0,1\}$, the value of q^b_v is

$$q^b_v = \prod_{v''\in \text{children}(v)} q^b_{v''}. \tag{17}$$

For the cell nodes that are the leaves of the tree $q^b_v = q^b_{c,l}$.

Algorithm 1 Generative model underlying the SNV model

Input: $d_{c,l}, c_{c,l}, \theta^{SNV}, b$
Output: $\nu_{c,l}$
if $b = 1$ **then**
  draw $l \sim \text{Bernoulli}(\epsilon_{FN})$ **if** $l = 0$ **then**
    draw $\mathcal{B}(g_{c,l}) \sim \text{Uniform}(\{1, \ldots, c_{c,l}\})$
    if $\mathcal{B}(g_{c,l}) = c_{c,l}$ **then**
      draw $\nu_{c,l} \sim \text{Binomial}(d_{c,l}, 1 - \epsilon_{FP})$
    else
      $\xi_{c,l} = \frac{g_{c,l}}{c_{c,l}}$
      draw $\nu_{c,l} \sim \text{Binomial}(d_{c,l}, \xi_{c,l})$
    end
  else
    draw $\nu_{c,l} \sim \text{Binomial}(d_{c,l}, \epsilon_{FP})$
  end
else
  draw $\nu_{c,l} \sim \text{Binomial}(d_{c,l}, \epsilon_{FP})$
end
return $\nu_{c,l}$

5.2 Detection of SNVs for individual cells

Given the underlying tree (denoted by t) and the read counts data (y^{SNV} denoted by y for simplicity), here the goal is to calculate the posterior probability of $x_{c,l}^{SNV}$ (denoted by $x_{c,l}$ for simplicity), where $x_{c,l}$ denotes the mutation status of locus l at cell c. According to the underlying tree, cells are attached to the trait nodes, and they are leaves of the tree (See Figure 7 (b)). Figure 7(a) shows a schematic view of a CNV tree in which the black nodes represent cells, and white nodes represent CNV traits. Figure 7(b) shows the tree after placement of SNV loci (The grey nodes represent SNV traits).

The joint probability distribution of $x_{c,l}$, y and t can be written as:

$$p(x_{c,l}, y, t) = \sum_{v \in R} \sum_{t' \in \mathcal{N}_v^l(t \backslash l)} p(x_{c,l}, t', y) \tag{18}$$

$$= \sum_{v \in R} \sum_{t' \in \mathcal{N}_v^l(t \backslash l)} p(x_{c,l}|t') p(y|t') p(t'), \tag{19}$$

where R is the set of all loci nodes in the tree (including the root) excluding locus l. The joint probability distribution is calculated as

$$p(x_{c,l} = 1, y, t) = \sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \backslash l)} p(y|t') p(t'), \tag{20}$$

where $x_{c,l}$ is a deterministic function of t' which belongs to $\mathcal{N}_v^l(t \backslash l)$. The value of $x_{c,l}$ is 1 if locus

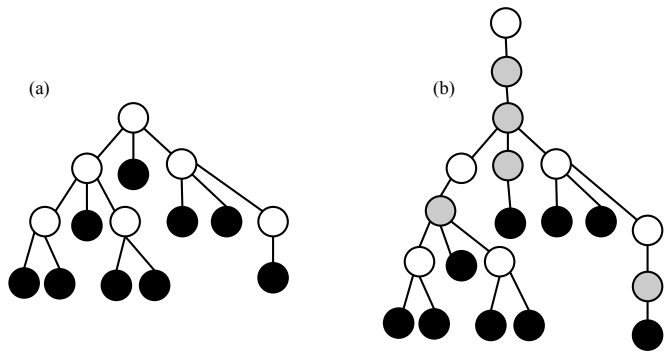

Figure 7: (a) A schematic view of CNV tree where the black nodes represent cells, and the blank nodes represent CNV traits. (b) A schematic view of the tree after adding SNV loci. The grey nodes represent SNV traits.

l is an ancestor of cell c, otherwise it is 0. The set $\mathcal{P}(c,t)$ denotes all the nodes on the path from cell c to the root of the tree (including the root and excluding the cell c node). An example of the path on an imaginary tree is depicted in Figure 8. The nodes coloured in green belong to $\mathcal{P}(c,t)$. Therefore, the posterior probability distribution of $x_{c,l} = 1$ yields

$$p(x_{c,l} = 1|y,t) = \frac{p(x_{c,l} = 1, y, t)}{p(y,t)} = \frac{\sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} p(y|t')p(t')}{p(y,t)}. \tag{21}$$

Rewriting Equation (21) assuming uniform probability distribution for $p(t')$ yields:

$$\begin{aligned}
p(x_{c,l} = 1|y,t) &\propto \sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} p(y|t'), \\
&= \sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} \prod_{l' \in N} \prod_{c' \in C} p(y_{c',l'}|t'), \\
&= \sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} \prod_{\substack{l' \in N \\ l' \neq l}} \prod_{c' \in C} p(y_{c',l'}|t') \prod_{m' \in C} p(y_{c',l}|t'), \\
&= K_1 \sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} \prod_{c' \in M} p(y_{c',l}|t'), \\
&= K_1 \sum_{v \in \mathcal{P}(c,t)} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} \prod_{c' \in M_{\setminus v}} p(y_{c',l}|t') \prod_{c' \in M_v} p(y_{c',l}|t'),
\end{aligned}$$

where N denotes the set of all trait nodes, M denotes the set of all cell nodes, M_v denotes the cells that are a descendant of node v, and $M_{\setminus v}$ denotes the cells that are a not descendant of node v. The product of the likelihood contributions for non-descendant nodes can be calculated by taking

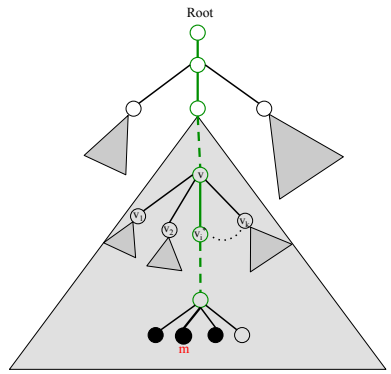

Figure 8: A schematic view of the underlying tree inferred from CNV and SNV loci across multiple cells. Black and white nodes represent cells and loci, respectively. The grey triangle represents a subtree rooted at a node. It includes all of the nodes and edges in the subtree.

the product of q_c^0 for all cells, divided by the ones that are descendant of v:

$$\prod_{c' \in M_{\setminus v}} q_{c'}^0 = \frac{q_{v^*}^0}{q_v^0}.$$

Therefore:

$$p(x_{c,l} = 1|y,t) \propto K_1 \sum_{v \in \mathcal{P}(c,t)} \frac{q_{v^*}^0}{q_v^0} \sum_{t' \in \mathcal{N}_v^l(t \setminus l)} \prod_{c' \in M_v} p(y_{c',l}|t'). \tag{22}$$

The likelihood contribution of descendant cells can be re-indexed by a binary vector of $\mathbf{b} = (b_1, b_2, \ldots, b_k)$, where $b_i \in \{0,1\}$, and $b_i = 1$ if the child v is to be moved into a child of the node l. The value of k denotes the number of children of v. The i^*th child of v which is on the path from node v to cell c is called v_i^*. This implies $b_{i^*} = 1$ (See Figure 8). Therefore:

$$\sum_{t' \in \mathcal{N}_v^l(t \setminus l)} \prod_{c' \in M_v} p(y_{c',l}|t') = q_{v_c^*}^1 \sum_{b_1=0}^{1} \sum_{b_2=0}^{1} \cdots \sum_{b_{i-1}=0}^{1} \sum_{b_{i+1}=0}^{1} \cdots \sum_{b_k=0}^{1} \prod_{\substack{i=1 \\ i \neq i^*}}^{k} q_{v_i}^{b_i}. \tag{23}$$

Rewriting Equation (22) using Equation (23) yields:

$$
\begin{aligned}
p(x_{c,l}=1|y,t) &\propto K_1 \sum_{v\in\mathcal{P}(c,t)} \frac{q^0_{v^*}}{q^0_v} q^1_{v^*_c} \sum_{b_1=0}^{1}\sum_{b_2=0}^{1}\cdots\sum_{b_{i-1}=0}^{1}\sum_{b_{i+1}=0}^{1}\cdots\sum_{b_k=0}^{1}\prod_{\substack{i=1\\ i\neq i^*}}^{k} q^{b_i}_{v_i}, \\
&= K_1 \sum_{v\in\mathcal{P}(c,t)} \frac{q^0_{v^*}}{q^0_v} q^1_{v^*_c} \prod_{\substack{i=1\\ i\neq i^*}}^{k} (q^0_{v_i}+q^1_{v_i}), \\
&= K_1 \sum_{v\in\mathcal{P}(c,t)} \frac{q^0_{v^*}}{q^0_v} \frac{\prod_{i=1}^{k}(q^0_{v_i}+q^1_{v_i})}{(q^0_{v_{i^*}}+q^1_{v_{i^*}})} q^1_{v_{i^*}}, \\
&= K_1 q^0_{v^*} \sum_{v\in\mathcal{P}(c,t)} \frac{q^1_{v_{i^*}}}{q^0_v(q^0_{v_{i^*}}+q^1_{v_{i^*}})} \prod_{i=1}^{k}(q^0_{v_i}+q^1_{v_i}).
\end{aligned} \tag{24}
$$

With this approach, the posterior probability of $x_{c,l}$ is calculated based on placing locus l at different places on the tree. This considers different subset of cells (sub-trees of the underlying tree) to calculate the posterior probability of $x_{c,l}$.

5.3 Computational complexity of the SNV calling algorithm

The computational complexity of Equation (24) is $O(|C|\cdot|L|)$ with $|C|$ is the number of cells and $|L|$, the number of loci. In order to reduce the complexity of calculating $p(x_{c,l}=1|y,t)$ for each locus and cell, $\mathcal{P}'(c,t)$ is defined to denote the nodes sitting on the path from root to cell c, excluding the root node and including the cell c node. Then,

$$
{q^*}_v = \prod_{i=1}^{k}(q^0_{v_i}+q^1_{v_i}). \tag{25}
$$

Therefore,

$$
K_1 q^0_{v^*} \sum_{v\in\mathcal{P}(c,t)} \frac{q^1_{v_{i^*}}}{q^0_v(q^0_{v_{i^*}}+q^1_{v_{i^*}})} \prod_{i=1}^{k}(q^0_{v_i}+q^1_{v_i}) = K_1 q^0_{v^*} \sum_{v\in\mathcal{P}'(c,t)} \frac{q^1_v}{(q^0_v+q^1_v)} \frac{{q^*}_{\text{parent}(v)}}{q^0_{\text{parent}(v)}}.
$$

Calculating $p(x_{c,l}=1|y,t)$ with a recursive approach reduces the complexity from $O(|C|N)$ to $O(|C|+N)$, where $N = l_{SNV}+l_{CNV}$.

6 Numerical results

We first present results on phylogenetic tree reconstruction, and then complement them with SNV variant call experiments based on the method of Section 5.

6.1 Phylogenetic tree experimental results

6.1.1 Synthetic data experiments

In this section we generated data based on a tree generated uniformly over topologies and observations according to $y_{c,l}|x_{c,l} \sim \text{Normal}(x_{c,l}, \sigma^2)$, varying σ^2 to assess robustness to noise and to benchmark different inference methods. The parameter σ^2 allows us to control the amount of noise in the observation. We generate matrices of size $|C| = 1000$ and $|L| = 50$ for $\sigma^2 \in \{1/10, 2/10, \ldots, 7/10\}$. In these experiments we provide the well-specified observation matrix to our inference methods.

We run the three inference methods described in Section 4.5 (MCMC, non-reversible parallel tempering, and adaptive sequential Monte Carlo) as well a maximum likelihood estimator approximated using the greedy scheme of Section 4.4, with $q_{c,l}(x_{c,l}) = p(y_{c,l}|x_{c,l}, \theta)$. Different initializations and number of iterations are also explored.

To assess the quality of tree reconstructions, we use the L1 loss function of Section 4.6. For the Bayesian methods, we use the minimum Bayes risk estimator also derived in Section 4.6 under the same loss. For the maximum likelihood method, we use the point estimate.

The results are shown in Figure 9. The Bayesian methods outperform the greedy maximum likelihood heuristic by a large margin. The two versions of the greedy method indicate whether the deterministic order described in Section 4.4 is used for locus inclusion or a randomized order. Surprisingly, a random order performs slightly better. In another experiment based on smaller scale data, and when the noise level was low, the informed heuristic performed better than the random order (Figure 10). These results support that to perform inference based on our model, the Bayesian method should be preferred to the greedy maximum likelihood method.

Among the Bayesian methods, the largest improvement comes from increasing the number of iterations from 1,000 to 10,000. Only marginal improvements are obtained by going from 10,000 to 100,000 iterations, similar to the gain obtained by using instead parallel tempering with 100 chains and 10,000 iterations. However one limitation of this result is that the non-reversible parallel tempering method uses an equally spaced annealing ladder, in the future it would be interesting to see if an adaption scheme, e.g. Syed et al. [2019], would improve its performance.

To provide some qualitative interpretation of the quality corresponding to the L1 reconstruction errors shown, we show some representative reconstructions at the end of the document, Figures 20–22. To visualize the consensus trees, we use the method outlined in Section 4.7. Recall that the columns of the cell-locus matrix are loci, in the case of real data, grouped by blocks of three columns. Recall also that the first column in each group shows (in black and white) the traits supporting the consensus tree, namely the matrix $x(t^*)$. The second column in each group (in white to red) shows the matrix $x(t^{\text{truth}})$, where t^{truth} is the tree used to generate the data. The third and last column in each group (in shades of grey), shows the posterior support, in other words the probability that $x_{c,t}$ takes the value one given the data.

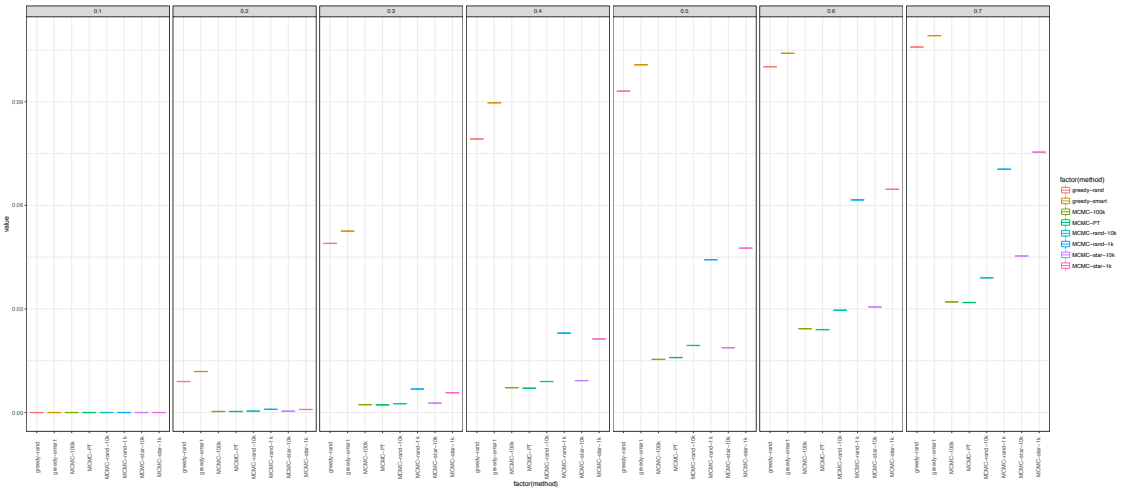

Figure 9: Synthetic data results. Boxes from left to right show different values for σ^2. The y axis measures the L1 loss, normalized by $|C||L|$. Method name suffixes encode the initialization and number of iterations.

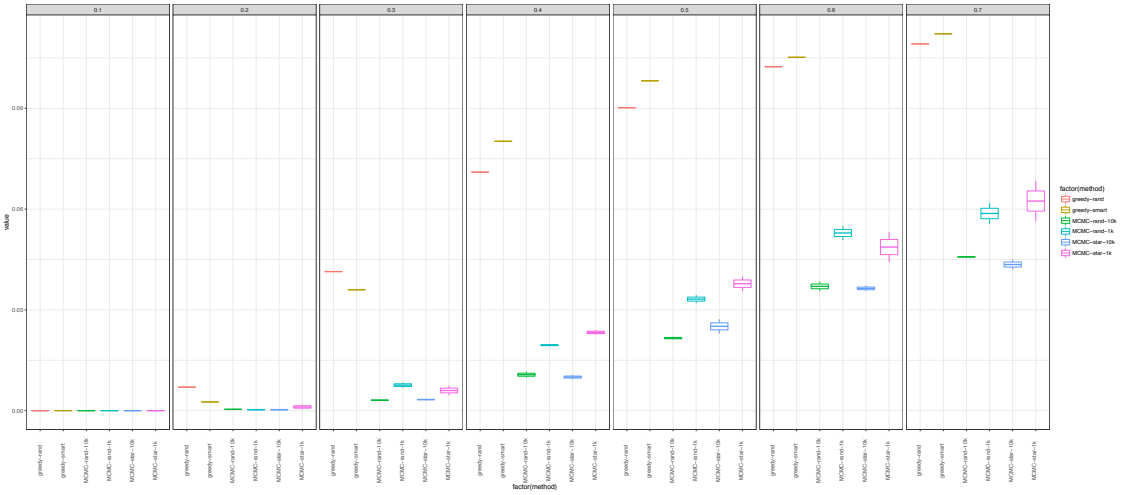

Figure 10: Experiments similar to Figure 9 but with smaller scale data (100 cells, 50 loci), showing a regime where the informed greedy outperforms greedy with randomized order. The two methods are still dominated by the Bayesian inference methods.

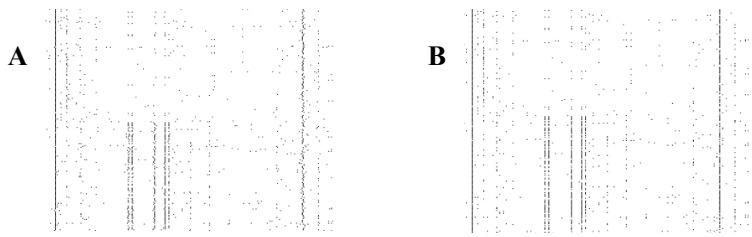

Figure 11: An example of the jitter correction algorithm on real data. **A**: original copy number call in the zoomed box of Figure 3. **B**: the same box after the jitter correction.

6.1.2 Real data

In this section we look at the inferred trees and parameters on two dataset. The first dataset, SA 535 from Laks et al. [2019], contains 679 cells from three passages of a triple negative breast cancer (TNBC) patient derived xenograft sample. Passages X1, X5, and X8 had 62, 369, and 231 cells post quality filtering respectively. We also included 17 mostly diploid control cells. These cells were combined to generate the input to the analysis pipeline. The second dataset consists in cells from three clonally related high grade serous (HGS) ovarian cancer samples, sourced from one primary tumour ($OV2295$) and two relapse specimens after therapy ($TOV2295$ and $OV2295(R)$); $OV2295$ and $OV2295(R)$ are from ascites, and $TOV2295$ is from the solid tumour. For simplicity, these three samples together are called $OV2295$.

The fact that the copy number calls are done independently for each cell implies that there is a slight jitter in the end points of the copy number change events (see Figure 3). To alleviate this, we use the following heuristic: visiting columns of the matrix y in decreasing order of density, perform a logical OR with the k columns in the right and left, if they have not been touched yet. We use $k = 2$ based on visual inspection. An example of the result of the jitter correction heuristic is shown in Figure 11. We removed columns l in y with relative density $\sum_{c \in C} y_{c,l}/|C|$ less than a threshold, set to 5%. This yields 104 loci.

Inferred phylogenies

We inferred trees using the locus-specific embedded into the following computational pipeline.

First, we approximate the posterior distribution using MCMC initialized at the star tree. We set the computational budget so that the inference takes about one day and samples, in the order of 100MB: 151,500,000 iterations, composed of 3000 scans each ended with recording a sample, each scan composed of 500 passes, where a pass goes through each locus parameter ($2|L|$) plus one tree move (this ratio is motivated in Section 4.5). Second, we compute the average over the matrices $x(t^1), x(t^2), \ldots$, denoted $\bar{x}$. Apply the transformation shown in Equation (9) to each entry of the average matrix. Denote the matrix obtained in this fashion q. Finally, we compute the consensus tree $t^* = M(q)$ as described in Section 4.4.

To visualize the consensus tree, we use the method outlined in Section 4.7. Recall that the

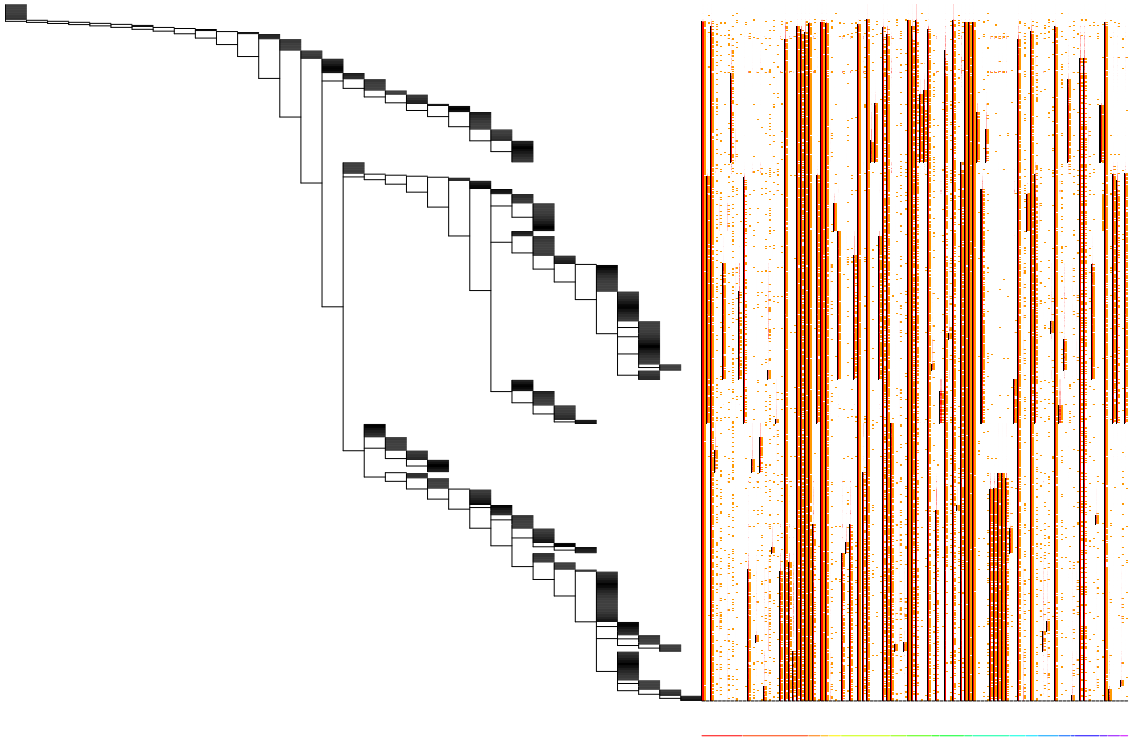

Figure 12: Full consensus tree from SA 535.

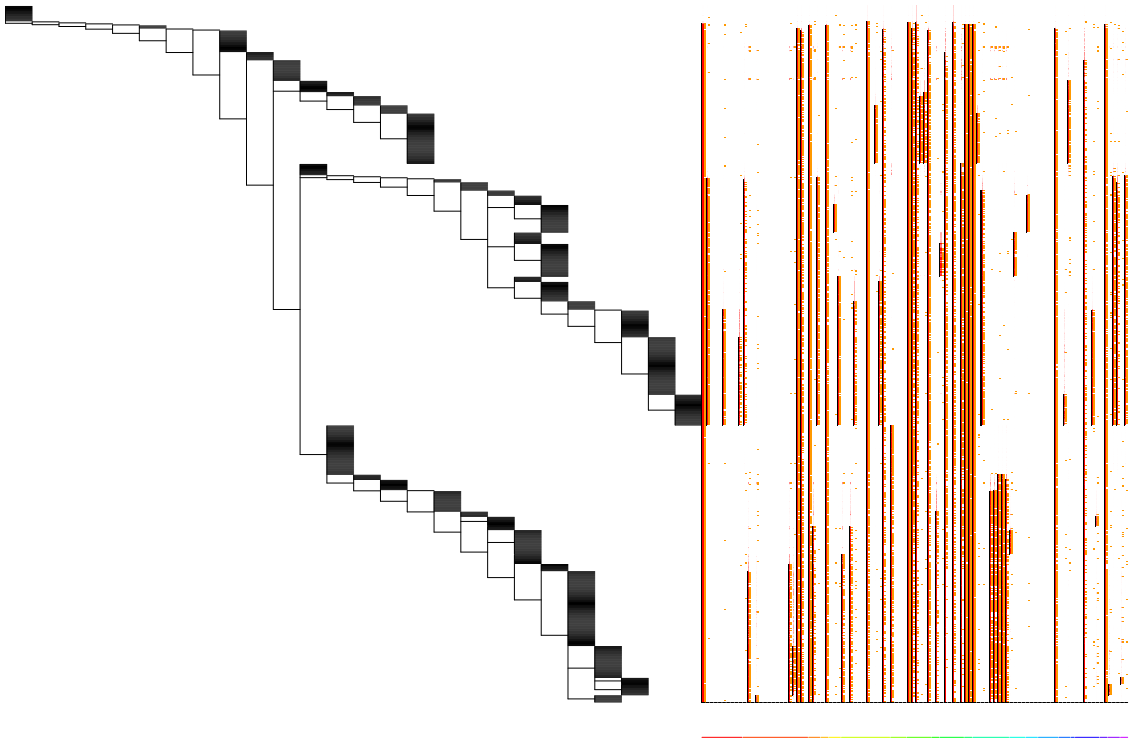

Figure 13: Consensus clades based on resolving the clades induced by the 60 loci with lowest inferred false positive rates.

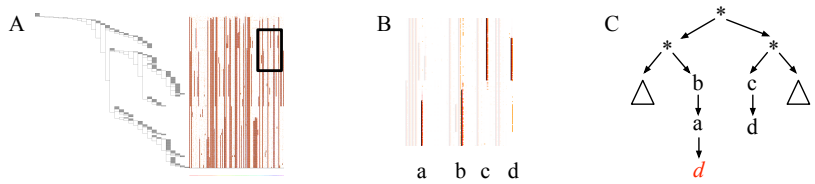

Figure 14: A locus in SA 535 that violates the perfect phylogeny assumption. The method is robust to this violation and correctly allocates the other relations. **A**: the part of the tree where the violation occurs, magnified in **B**, where the short band of orange not matched with a black band is indicative of the violation. **C**: the corresponding tree showing the two points of the tree where the two events collocated in the same bin occurred in the tree. The sister clades of b and c suggest that this is a case of two insertion events rather than a loss.

columns of the cell-locus matrix are loci, in the case of real data, grouped by blocks of three columns. As before the first column in each group shows (in black and white) the traits supporting the consensus tree, namely the matrix $x(t^*)$. The second column in each group (in colours from white to red) shows the matrix $\bar{x}$ which can be interpreted as a confidence matrix. The third and last column in each group (in white and orange), shows the jitter-corrected data. Each group corresponds to a locus ordered from chromosome 1, 2, ... (chromosome identity indicated by the rainbow at the bottom of the plot.

We focus here on the description of the SA 535 dataset. The tree identifies several clades with strong support, and also reveal a locus that clearly violate the perfect phylogeny assumption, hence supporting the "majority hypothesis" made in Section 3.3.3 and also that the model is robust to such violation.

We start by highlighting the clades that are most clearly supported by the data. To do so, we first rank the loci by the false positive rates based on t^* and the data y. Since we expect the false positive rate to be relatively low (see discussion in Section 3.4), we expect those with high inferred false positive rates to correspond to loci involved in perfect phylogeny violations. We show in Figure 13 the tree obtained by resolving the splits created by the first 60 out of 104 loci. The clades are in close agreement with the calls y in orange, with the exception of < 10 cells on the top. Notice that many events are clearly paired, as one would expect; note that prior information about pairings is not provided to the inference algorithm but rather recovered naturally from the data.

Next, we show in Figure 14 a locus which violates the perfect phylogeny assumptions. The locus shown at column (d) is likely a bin which contains two distinct copy number events, one under the event at (a), and one under the event at (c). Some cases like this could be avoided by looking for discordant reads, and excluding bins where sets of discordant reads in two different loci within the bin are found. However since our method is resilient to violations we do not pursue this direction.

A second potential violation is shown in Figure 15, where in this case the evidence points

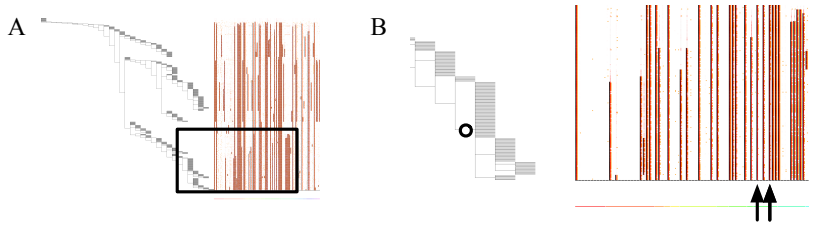

Figure 15: A second locus in SA 535 that potentially violates the perfect phylogeny assumption. **A**: the part of the tree where the violation occurs. **B**: the two loci above the arrows have a region where the density of the observed positive trait is lower yet the region is still reconstructed as having the trait (black band). These two loci might have been affected by a loss at the location of the tree indicated with a circle.

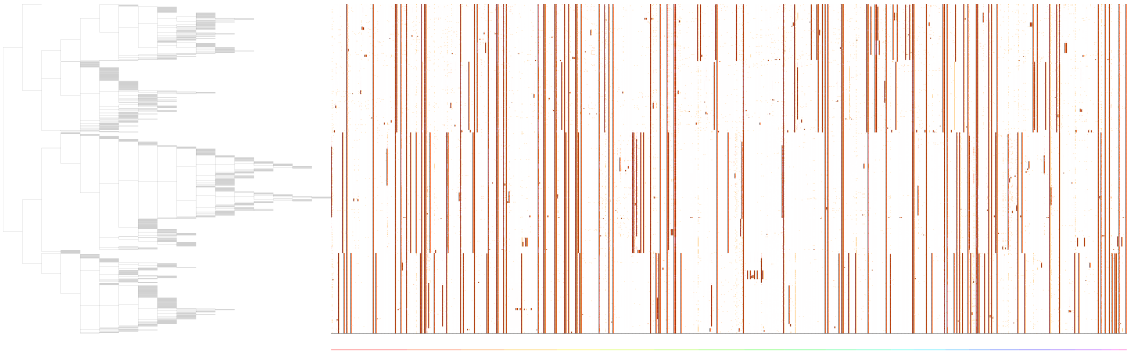

Figure 16: Consensus tree from HGS OV data.

towards a loss. The evidence is not as clear in this second case.

Fortunately, the method is robust to these two violations and correctly allocates the other relations between loci a–c, and assign d to one of the two possible choices. We found no other locus among the 104 loci which exhibit violations.

Figure 16 shows the consensus tree reconstructed from 947 cells pooled from the three samples in the HGS dataset. Preprocessing the data yields 574 loci. Each specimen comprises one of the major clades in the tree, with the two post treatment samples grouped together.

Inferred parameters

We show in Figure 17 the inferred posterior distributions on the $2|L|$ locus-specific false positive and false negative parameters for the SA 535 dataset.

Encouragingly, several posterior distributions put most of the mass to very low error rates. The parameters with large uncertainty on the false positive and low uncertainty on the false negative are likely due to clades that are high in the tree; and vice-versa: large uncertainty on the false negative and low uncertainty on the false positive are likely due to clades that are low in the tree.

We also show the trace plots in Figure 18 to support that the sampler mixes well. The trace

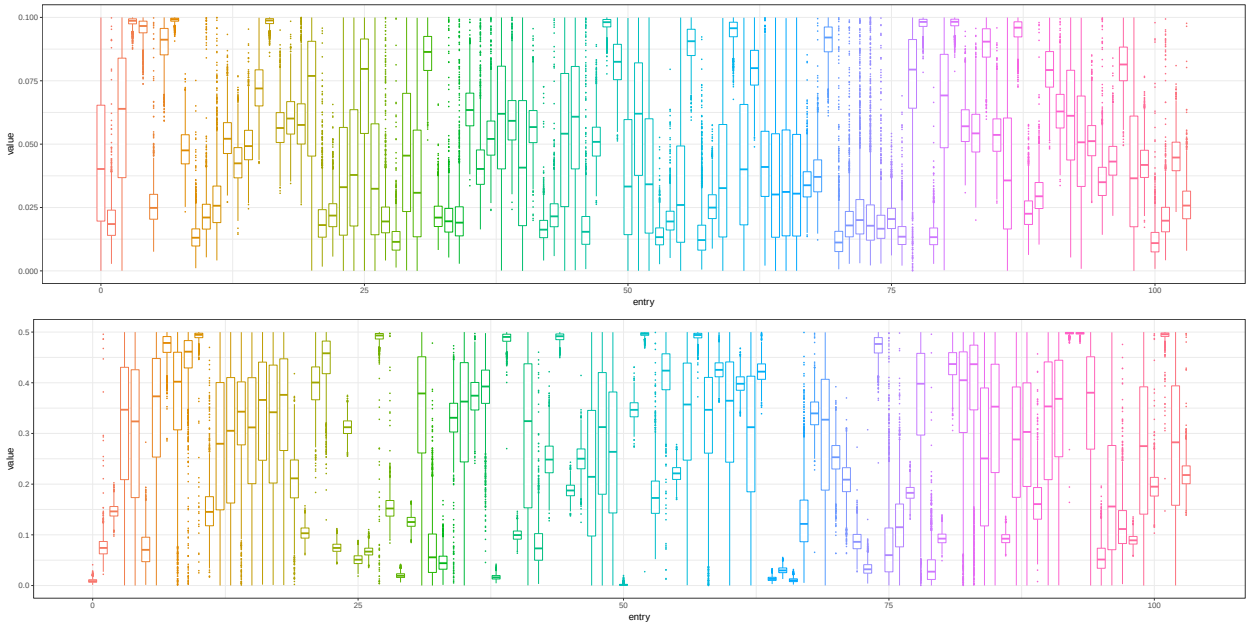

Figure 17: Posterior distributions over the false positive rate parameters (top) and false negative rate parameters (bottom).

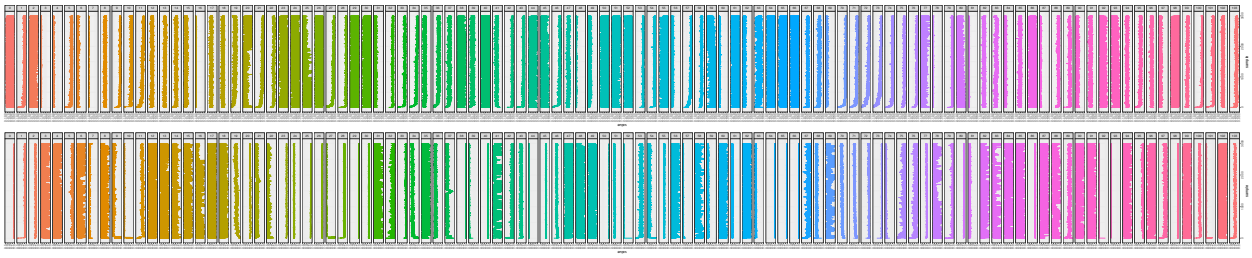

Figure 18: Trace plots for the false positive rate parameters (top) and false negative rate parameters (bottom).

plot for the full log density is also shown in Figure 19.

6.2 Mutation calling experimental results

6.2.1 Synthetic data generation

To generate the synthetic data, a tree is uniformly sampled from the set of undirected trees with $N = N_{SNV} + N_{CNV}$ nodes. Then cells are assigned to the trait nodes such that the cell nodes are the leaves of the the tree. The internal nodes are randomly assigned to either a SNV locus or to a CNV locus. For each SNV locus, the copy number and the number of reads covering the locus (depth) are independently generated. The copy number is generated from a discrete uniform distribution from 0 to $c_{\max}$ ($c_{\max}$ equals 5 in the analysis); the depth is generated from a Poisson distribution given the coverage. Given the depth and the copy number, a random genotype is uniformly sampled from all possible genotype states ($\boldsymbol{\mathcal{G}}$). The number of variant reads is sampled

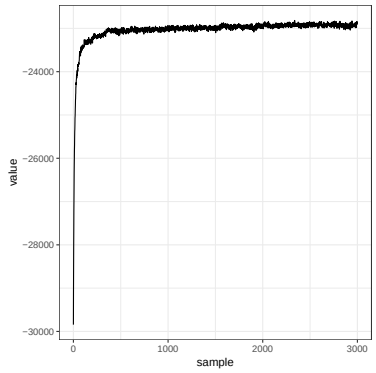

Figure 19: Trace plot of the log density across MCMC iterations.

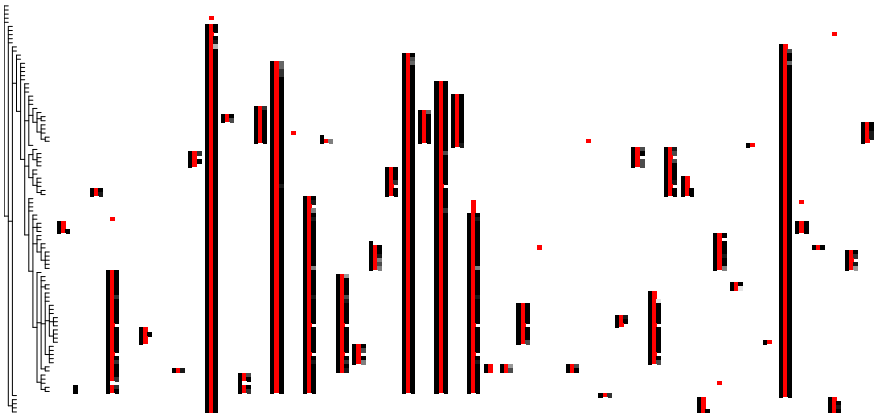

Figure 20: Example of consensus tree on synthetic data, in the regime of medium-low noise ($\sigma^2 = 0.3$), based on MCMC with a star initialization and 10,000 iterations. The corresponding L1 error is less than 0.01.

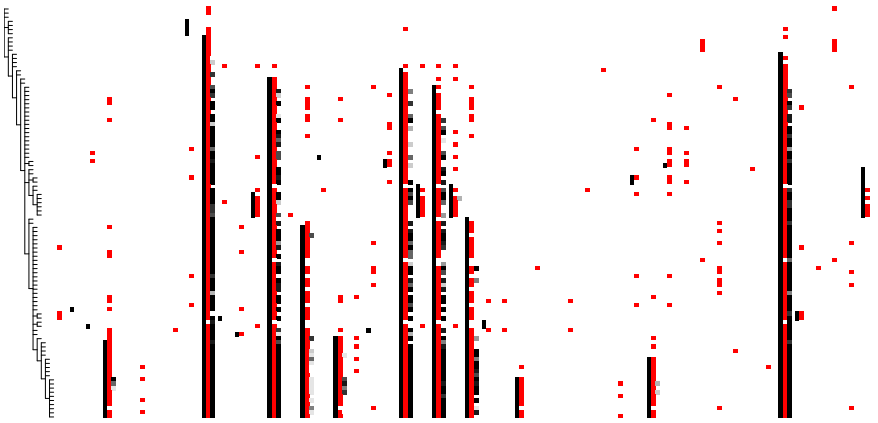

Figure 21: Example of consensus tree on synthetic data, in the regime of high noise ($\sigma^2 = 0.7$), based on MCMC with a star initialization and 10,000 iterations. The corresponding L1 error is 0.03.

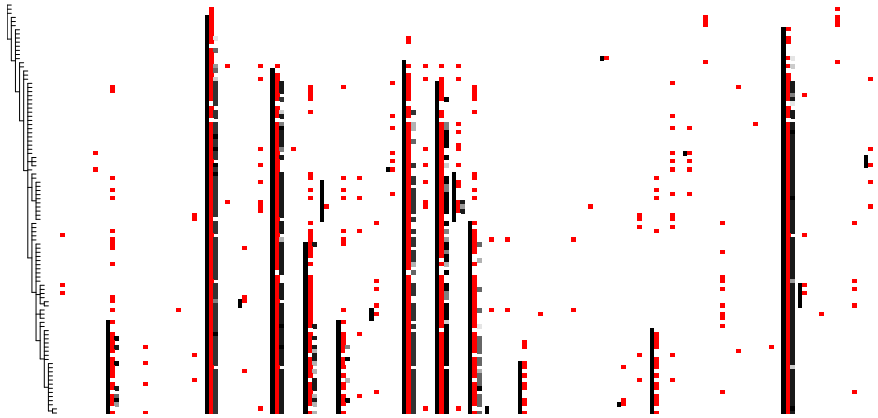

Figure 22: Example of consensus tree on synthetic data, in the regime of high noise ($\sigma^2 = 0.7$), based on MCMC ran with not enough iterations, here 1,000, with a star initialization. The corresponding L1 error is 0.075.

from a binomial distribution given depth, copy number, false positive and false negative error rate parameters (See Algorithm 1).

6.2.2 Synthetic data evaluation

In this section we evaluate both the topology of the inferred tree, and the accuracy of SNV calls. We define a distance D to evaluate the tree topology performance. Consider X to be a matrix derived from the phylogeny, where $x(i, j) = 1$ if locus i is an ancestor of cell j. If X_1 is derived based on a reference tree and X_2 is derived from an inferred tree, then the distance between X_1 and X_2 is estimated by

$$D = \sum_i \sum_j |(x_1(i, j) - x_2(i, j))|/z, \tag{26}$$

where z is the total number of entries of matrix X_1 or X_2 (both are the same size).

The performance of SNV calls are estimated by sensitivity, specificity and Youden's index. They are estimated given both the binary SNV calls, and the ground truth calls (derived from the reference tree). The SNV calls are binarized by setting a threshold equal to 0.5 (probability greater than 0.5 is 1 and smaller than 0.5 is 0). The topology of the reference tree provides the ground truth of SNV calls in which locus i is mutated in the entire descendant cells of a sub-tree rooted at i.

The performance of SNV calls are estimated by sensitivity (true positive rate), specificity (true negative rate) and Youden's index (sensitivity + specificity - 1). They are estimated given both the binary SNV calls, and the ground truth calls (derived from the reference tree). The SNV calls are binarized by setting a threshold equal to 0.5 (probability greater than 0.5 is 1 and smaller than 0.5 is 0). The topology of the reference tree provides the ground truth of SNV calls in which locus i is mutated in the entire descendant cells of a sub-tree rooted at i.

In Figure 23 and 24, we explore how the performance changes with different proportion of

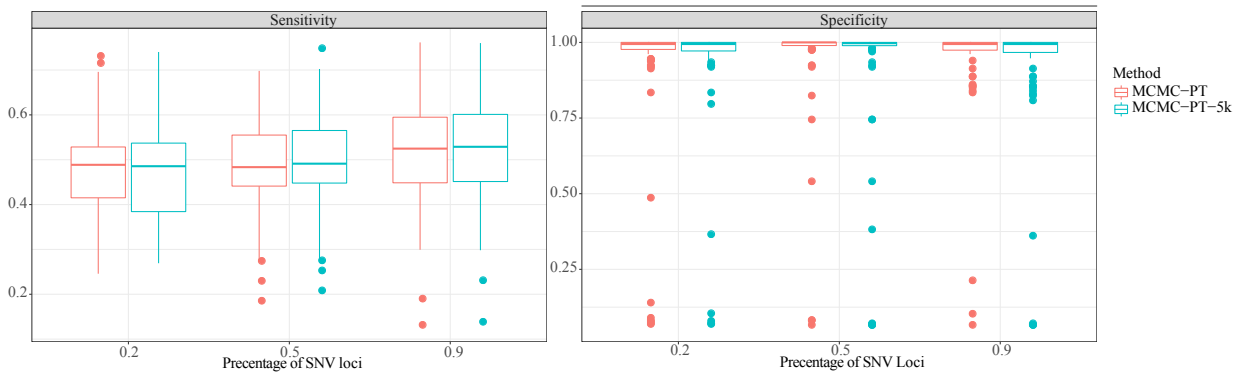

Figure 23: The performance (Sensitivity and Specificity) of SNV calls if various percentage of SNV loci are used for the inference, and the remaining are added afterwards. The data is generated for 200 SNV loci, 100 CNV loci across 1000 cells for 10 different topology. Copy number is uniformly sampled from $\{1 \ldots 5\}$ and the coverage mean is 0.07. MCMC-PT and MCMC-PT-5k denote the inference algorithms, and use PT with 1000 and 5000 iterations, respectively.

SNVs used for tree inference. We assign a copy number to each SNV locus. The copy number is uniformly sampled from $\{1 \ldots 5\}$. The genotype is also uniformly sampled from all possible genotype states given the copy number. The number of reads and the number of variant alleles are generated following the binomial probability distribution defined in 14 and 12 given the genotype and different coverage means of 0.04, 0.07, 0.1, 0.5 or 2. The inference is done for 10 different seeds, and non-reversible Parallel Tempering (PT) algorithm [Syed et al., 2019] with 1000 and 5000 number of iterations with one chain. The values of ϵ_{FP} and ϵ_{FN} are equal to 0.01. The inference parameters are constant through all the following analysis. The performance of detecting SNV is higher when 90% percent of SNV loci are used jointly with CNVs at tree inference. However, the difference is not significant. Figure 25, suggests that incorporating SNVs improves resolving the phylogeny. However, for larger number of the nodes, we may need to increase the number of iterations to reach a higher accuracy.

In Figure 26 and 27, we demonstrate the performance of single nucleotide variant calls for data with different coverage. The data is generated for 200 single nucleotide variant loci, 100 copy-number loci across 1000 cells. We randomly picked 90% of SNV loci and utilized them with the CNV loci for inferring the tree. The remaining 10% SNV loci are later added to the inferred tree. The nodes are assigned to 10 tree generated uniformly over topologies. As expected, the higher coverage the better performance. As depicted in Figure 26 and 27, the performance is not very high with 0.07x coverage. However, the performance reaches almost 1 with 2x coverage. Note that the results are evaluated per cell, and these coverage are all lower than the coverage of the data used in all other existing methods. Figure 28 depicts the performance of the model (in terms of tree topology) improves with more coverage.

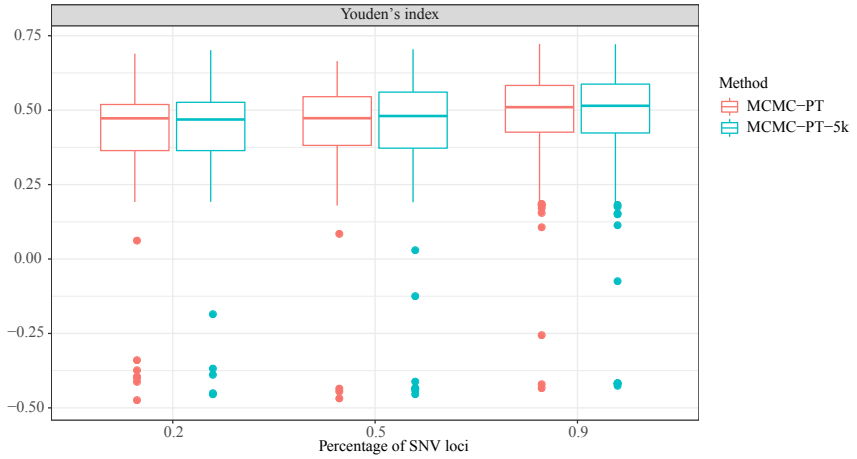

Figure 24: The performance (Youden's index) of SNV calls if various percentage of SNV loci are used for the inference, and the remaining are added afterwards. The data is generated for 200 SNV loci, 100 CNV loci across 1000 cells for 10 different topology. Copy number is uniformly sampled from $\{1 \ldots 5\}$ and the coverage mean is 0.07. MCMC-PT and MCMC-PT-5k denote the inference algorithm, and use PT with 1000 and 5000 iterations, respectively.

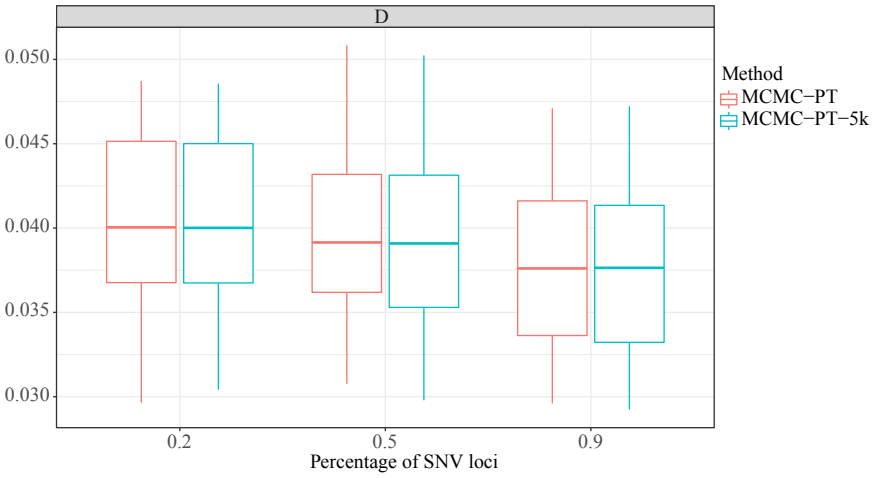

Figure 25: The performance (D) of tree topology if various percentage of SNV loci are used for the inference, and the remaining are added afterwards. The data is generated for 200 SNV loci, 100 CNV loci across 1000 cells for 10 different topology. Copy number is uniformly sampled from $\{1 \ldots 5\}$ and the coverage mean is 0.07. MCMC-PT and MCMC-PT-5k denote the inference algorithms, and use PT with 1000 and 5000 iterations, respectively.

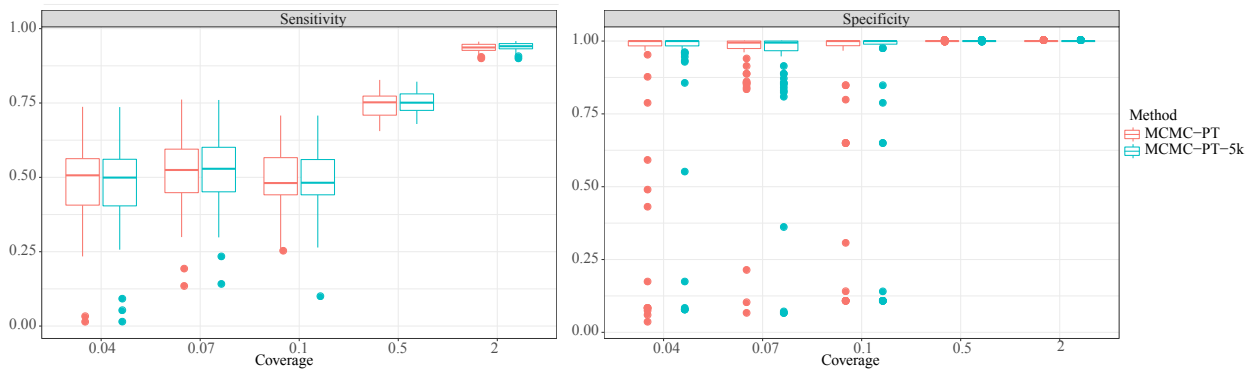

Figure 26: The performance (Sensitivity and Specificity) of SNV calls across data coverages of $0.04, 0.07, 0.1, 0.5$, and 2. The data is generated for 200 SNV loci, 100 CNV loci across 1000 cells for 10 different topologies. Only 90% of SNV loci are used for the inference, and the remaining loci are added to the tree afterwards. Copy number is uniformly sampled from $\{1 \ldots 5\}$. MCMC-PT and MCMC-PT-5k denote the inference algorithm using PT with 1000 and 5000 iterations, respectively.

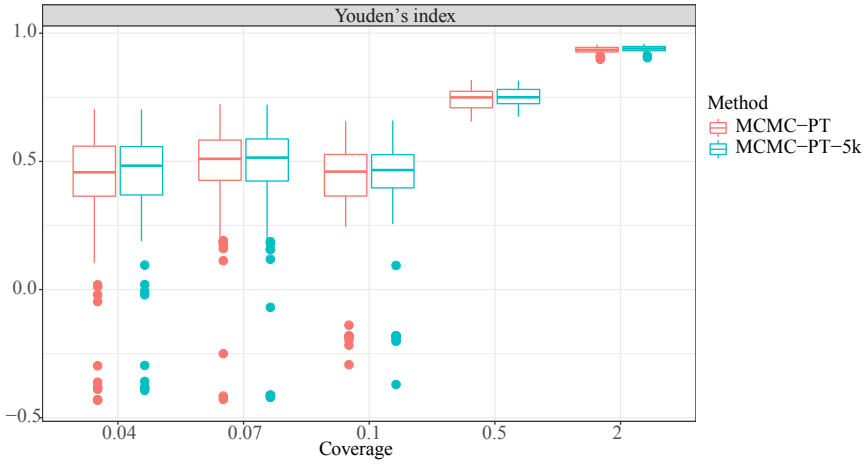

Figure 27: The performance (Youden's index) of SNV calls across data coverages of $0.04, 0.07, 0.1, 0.5$, and 2. The data is generated for 200 SNV loci, 100 CNV loci across 1000 cells for 10 different topology. Only 90% of SNV loci are used for the inference, and the remaining are added to the tree afterwards. Copy number is uniformly sampled from $\{1 \ldots 5\}$. MCMC-PT and MCMC-PT-5k denote the inference algorithm using PT with 1000 and 5000 iterations, respectively.

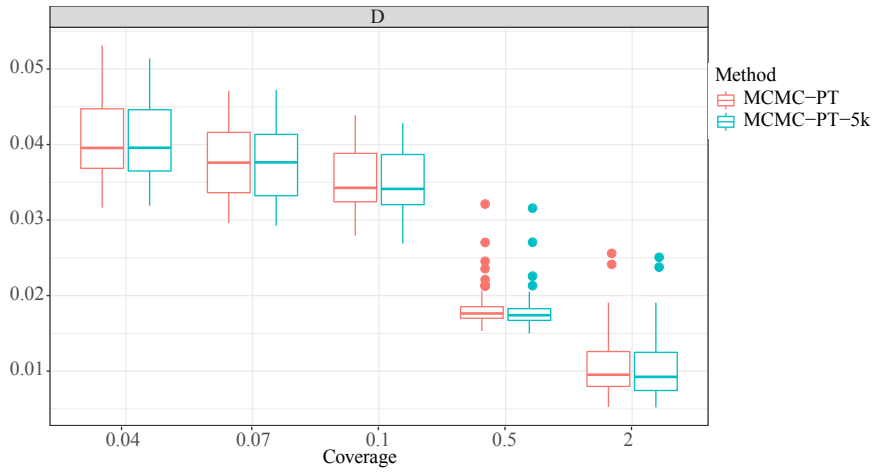

Figure 28: The performance (D) of tree topology across data coverages of $0.04, 0.07, 0.1, 0.5$, and 2. The data is generated for 200 SNV loci, 100 CNV loci across 1000 cells for 10 different topology. Only 90% of SNV loci are used for the inference, and the remaining are added to the tree afterwards. Copy number is uniformly sampled from $\{1 \ldots 5\}$. MCMC-PT and MCMC-PT-5k denote the inference algorithm using PT with 1000 and 5000 iterations, respectively.

6.2.3 Real data

We focus on the high-grade serous ovarian cancer described in Section 6.1.2. SNV loci candidates are selected from the aggregated data in which all cells were collapsed into a 'pseudo-bulk' genome. The SNV loci candidates are identified as described in Salehi et al. [2020] and this number is 23005 for $OV2295$.

The loci are sorted using R's 'hclust' function with 'average' metric and 'manhattan' distance. Figure 30 depicts the number of variant reads in each cell across the same set of loci. The loci in SNV call probability plot (Figure 29) and the plot that depicts the number of variant reads (Figure 30) follow a exactly similar order. Figure 29 depicts SNVs present in all or only a subset of cells that are phylogenetically (based on the tree) near to each other. Considering both Figures 30 and 29, the number of variant reads in the low probability area is small (black area in Figure 30, blue area in Figure 29).

References

N. Beerenwinkel, R. F. Schwarz, M. Gerstung, and F. Markowetz. Cancer Evolution: Mathematical Models and Computational Inference. *Systematic Biology*, 64(1):e1–e25, January 2015. ISSN 1063-5157.

C. L. Bohrson, A. R Barton, M. A. Lodato, R. E. Rodin, L. J. Luquette, V. V. Viswanadham, D. C. Gulhan, I. Cortés-Ciriano, M. A. Sherman, M. Kwon, M. E. Coulter, A. Galor, C. A. Walsh, and

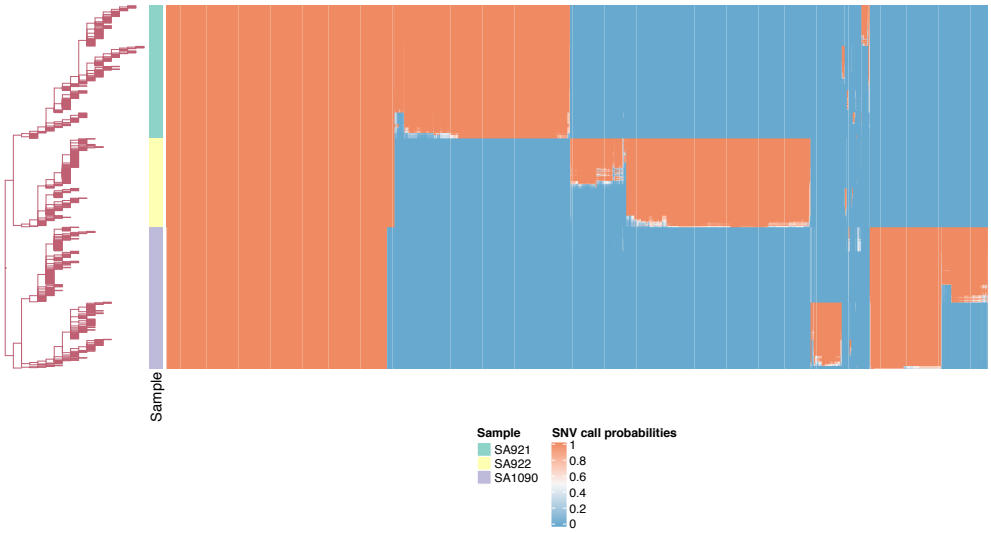

Figure 29: The heatmap depicts $OV2295$ SNV call probability of 200 random loci across the cells. The blue colour depicts the low probabilities near 0, the white colour depicts the probabilities around 0.5, and the orange colour depicts the probabilities around 1. The inferred underlying phylogeny tree based on CNV data is on the left side of the heatmap. The samples colour coded. The cells are from three samples $OV2295$ ($SA1090$), $OV2295(R)$ ($SA921$) and $TOV2295(R)$ ($SA922$).

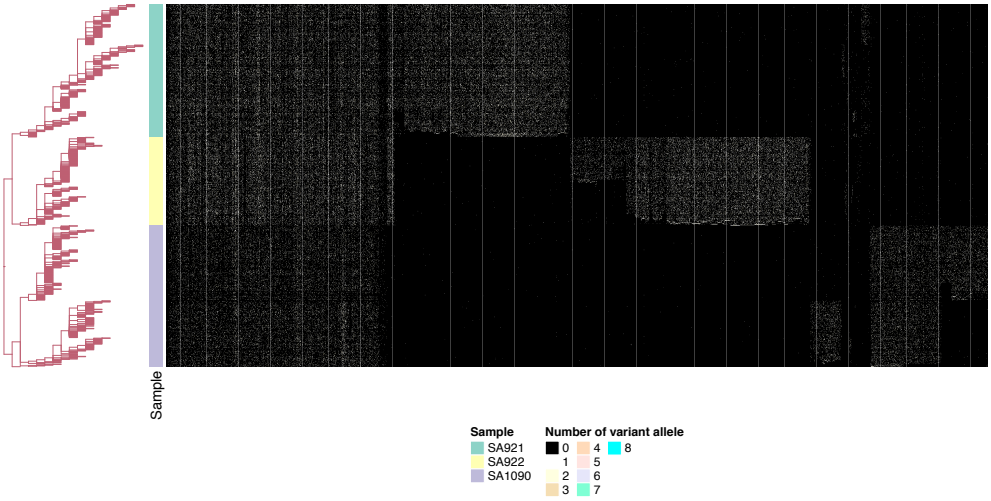

Figure 30: The heatmap depicts the number of variant reads observed in each cell of $OV2295$ for 200 random loci. No variant read is depicted in black, and reads with variant alleles are depicted in bright colours.The inferred underlying phylogeny tree based on CNV data is on the left side of the heatmap. The samples colour coded. The cells are from three samples $OV2295$ ($SA1090$), $OV2295(R)$ ($SA921$) and $TOV2295(R)$ ($SA922$).

P. J. Park. Linked-read analysis identifies mutations in single-cell DNA-sequencing data. *Nature Genetics*, 51(4):749–754, 2019.

A. Bouchard-Côté, K. Chern, D. Cubranic, S. Hosseini, J. Hume, M. Lepur, Z. Ouyang, and G. Sgarbi. Blang: Bayesian declarative modelling of arbitrary data structures. *arXiv:1912.10396 [stat]*, 2019.

C. Robert. *The Bayesian Choice - From Decision-Theoretic Foundations | Christian Robert | Springer*. 2007.

V. Chandrasekaran and M. I. Jordan. Computational and Statistical Tradeoffs via Convex Relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181–E1190, March 2013. ISSN 0027-8424, 1091-6490. arXiv: 1211.1073.

S. A. Chowdhury, S. E. Shackney, K. Heselmeyer-Haddad, T. Ried, A. A. Schäffer, and R. Schwartz. Phylogenetic analysis of multiprobe fluorescence in situ hybridization data from tumor cell populations. *Bioinformatics*, 29(13):i189–i198, 2013. ISSN 1367-4803.

S. A. Chowdhury, S. E. Shackney, K. Heselmeyer-Haddad, T. Ried, A. A. Schäffer, and R. Schwartz. Algorithms to Model Single Gene, Single Chromosome, and Whole Genome Copy Number Changes Jointly in Tumor Phylogenetics. *PLOS Computational Biology*, 10(7):e1003740, 2014. ISSN 1553-7358.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(3):411–436, 2006. ISSN 1369-7412.

X. Dong, L. Zhang, B. Milholland, M. Lee, A. Y Maslov, T. Wang, and J. Vijg. Accurate identification of single-nucleotide variants in whole-genome-amplified single cells. *Nature methods*, 14 (5):491—493, May 2017.

C. D. Greenman, E. D. Pleasance, S. Newman, F. Yang, B. Fu, S. Nik-Zainal, D. Jones, K. W. Lau, N. Carter, P. A. W. Edwards, P. Andrew Futreal, M. R. Stratton, and P. J. Campbell. Estimation of rearrangement phylogeny for cancer genomes. *Genome Research*, 22(2):346–361, February 2012. ISSN 1088-9051, 1549-5469.

J. Hård, E. Al Hakim, M. Kindblom, Å. K. Björklund, B. Sennblad, I. Demirci, M. Paterlini, P. Reu, E. Borgström, P. L. Ståhl, J. Michaelsson, J. E. Mold, and J. Frisén. Conbase: a software for unsupervised discovery of clonal somatic mutations in single cells through read phasing. 20(1): 68, 2019.

C. Holohan, S. Van Schaeybroeck, D. B Longley, and P. G Johnston. Cancer drug resistance: an evolving paradigm. *Nature Reviews Cancer*, 13(10):714, 2013.

J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001. ISSN 1367-4803.

K. Jahn, J. Kuipers, and N. Beerenwinkel. Tree inference for single-cell data. *Genome Biology*, 17: 86, 2016. ISSN 1474-760X.

K. I. Kim and R. Simon. Using single cell sequencing data to model the evolutionary history of a tumor. *BMC Bioinformatics*, 15(1):27, January 2014. ISSN 1471-2105.

J. Kuipers, K. Jahn, and N. Beerenwinkel. Advances in understanding tumour evolution through single-cell sequencing. *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 2017. ISSN 0304-419X.

E. Laks, A. McPherson, H. Zahn, D. Lai, A. Steif, J. Brimhall, J. Biele, B. Wang, T. Masud, J. Ting, et al. Clonal decomposition and DNA replication states defined by scaled single-cell genome sequencing. *Cell*, 179(5):1207–1221, 2019.

P. Lemey, M. Salemi, and A.-M. Vandamme. *The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing*. Cambridge University Press, 2009.

E. Letouzé, Y. Allory, M. A. Bollet, F. Radvanyi, and F. Guyon. Analysis of the copy number profiles of several tumor samples from the same patient reveals the successive steps in tumorigenesis. *Genome Biology*, 11(7):R76, 2010. ISSN 1474-760X.

Y. Li, S. Zhou, D. C. Schwartz, and J. Ma. Allele-Specific Quantification of Structural Variations in Cancer Genomes. *Cell Systems*, 3(1):21–34, 2016. ISSN 2405-4712.

A. Mahmoody, C. L. Kahn, and B. J. Raphael. Reconstructing genome mixtures from partial adjacencies. *BMC Bioinformatics*, 13(Suppl 19):S9, 2012. ISSN 1471-2105.

A. McPherson, C. Wu, A. W. Wyatt, S. Shah, C. Collins, and S. C. Sahinalp. nFuse: Discovery of complex genomic rearrangements in cancer using high-throughput sequencing. *Genome Research*, 2012. ISSN 1088-9051, 1549-5469.

A. W. McPherson, A. Roth, G. Ha, C. Chauve, A. Steif, C. P. E. de Souza, P. Eirew, A. Bouchard-Côté, S. Aparicio, S. C. Sahinalp, and S. P. Shah. ReMixT: clone-specific genomic structure estimation in cancer. *Genome Biology*, 18(1):140, 2017. ISSN 1474-760X.

R.M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003. ISSN 0090-5364, 2168-8966.

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

E. M. Ross and F. Markowetz. OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome Biology*, 17:69, 2016. ISSN 1474-760X.

S. Salehi, F. Kabir, N. Ceglia, M. Andronescu, M. Williams, K. R. Campbell, T. Masud, B. Wang, J. Biele, J. Brimhal, J. Ting, A. W. Zhang, C. O'Flanagan, F. Dorri, N. Rusk, E. Laks, H. Lee, T. Algara, S. Lee, B. Y. C. Cheng, P. Eirew, T. Kono, J. Pham, D. Grewel, D. Lai, R. Moore,

A. J. Mungall, M. A Marra, IMAXT Consertium, A. McPherson, A. Bouchard-Côté, S. Aparicio, and S. P. Shah. Single cell fitness landscapes induced by genetic and pharmacologic perturbations in cancer. *To be published*, 2020.

R. Schwartz and A. A. Schäffer. The evolution of tumour phylogenetics: principles and practice. *Nature Reviews Genetics*, 18(4):213–229, 2017. ISSN 1471-0064.

R. F. Schwarz, A. Trinh, B. Sipos, J. D. Brenton, N. Goldman, and F. Markowetz. Phylogenetic Quantification of Intra-tumour Heterogeneity. *PLOS Computational Biology*, 10(4):e1003535, 2014. ISSN 1553-7358.

J. Singer, J. Kuipers, K. Jahn, and N. Beerenwinkel. SCIΦ: Single-cell mutation identification via phylogenetic inference. *bioRxiv*, page 290908, March 2018.

S. Syed, A. Bouchard-Côté, G. Deligiannidis, and A. Doucet. Non-Reversible Parallel Tempering: an Embarassingly Parallel MCMC Scheme. *arXiv:1905.02939 [stat]*, May 2019. arXiv: 1905.02939.

S. Tavare, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring Coalescence Times from DNA Sequence Data. *Genetics*, 145(2):505–518, February 1997. ISSN 0016-6731.

E. I. Velazquez-Villarreal, S. Maheshwari, J. Sorenson, I. T. Fiddes, V. Kumar, Y. Yin, M. Webb, C. Catalanotti, M. Grigorova, P. A. Edwards, J. D. Carpten, and D. W. Craig. Resolving subclonal heterogeneity within cell-line growths by single cell sequencing genomic DNA. *bioRxiv*, 2019.

T. L. Williams and B. M. E. Moret. An investigation of phylogenetic likelihood methods. In *Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings.*, pages 79–86, March 2003.

D.B. Wilson. Generating Random Spanning Trees More Quickly Than the Cover Time. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 296–303, New York, NY, USA, 1996. ACM. ISBN 978-0-89791-785-8.

X. Xu, Y. Hou, X. Yin, L. Bao, A. Tang, L. Song, F. Li, S. Tsang, K. Wu, H. Wu, W. He, L. Zeng, M. Xing, R. Wu, H. Jiang, X. Liu, D. Cao, G. Guo, X. Hu, Y. Gui, Z. Li, W. Xie, X. Sun, M. Shi, Z. Cai, B. Wang, M. Zhong, J. Li, Z. Lu, N. Gu, X. Zhang, L. Goodman, L. Bolund, J. Wang, H. Yang, K. Kristiansen, M. Dean, Y. Li, and J. Wang. Single-Cell Exome Sequencing Reveals Single-Nucleotide Mutation Characteristics of a Kidney Tumor. *Cell*, 148(5):886–895, March 2012. ISSN 0092-8674.

T. I. Zack, S. E. Schumacher, S. L. Carter, A. D. Cherniack, G. Saksena, B. Tabak, M. S. Lawrence, C.-Z. Zhang, J. Wala, C. H. Mermel, C. Sougnez, S. B. Gabriel, B. Hernandez, H. Shen, P. W. Laird, G. Getz, M. Meyerson, and R. Beroukhim. Pan-cancer patterns of somatic copy-number alteration. *Nature genetics*, 45(10):1134–1140, 2013. ISSN 1061-4036.

H. Zafar, A. Tzen, N. Navin, K. Chen, and L. Nakhleh. SiFit: A Method for Inferring Tumor Trees from Single-Cell Sequencing Data under Finite-site Models. *bioRxiv*, page 091595, 2016a.

H. Zafar, Y. Wang, L. Nakhleh, N. Navin, and K. Chen. Monovar: single-nucleotide variant detection in single cells. *Nature Methods*, 13:505, April 2016b.

H. Zahn, A. Steif, E. Laks, P. Eirew, M. VanInsberghe, S.P. Shah, S. Aparicio, and C.L. Hansen. Scalable whole-genome single-cell library preparation without preamplification. *Nature Methods*, 14(2):167–173, 2017. ISSN 1548-7105.

Y. Zhou, A.M. Johansen, and J. A. D. Aston. Towards Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach. *arXiv:1303.3123 [stat]*, March 2013. arXiv: 1303.3123.