bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

SCuPhr: A Probabilistic Framework for Cell Lineage Tree Reconstruction

Hazal Koptagel[1], Seong-Hwan Jun[1], and Jens Lagergren[1]

[1]Science for Life Laboratory, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

June 28, 2018

Abstract

Reconstruction of cell lineage trees from single-cell DNA sequencing data, has the potential to become a fundamental tool in study of development of disease, in particular cancer. For cells without copy number alterations that has not been exposed to specific marking techniques, that is normal cells, lineage tracing is naturally based on somatic point mutations. Current single cell sequencing techniques applicable to such cells require an amplification step, which introduces errors, and still often suffer from so-called allelic dropout. We present a detailed model of current technologies for the purpose of estimating the distance between cells without copy number changes, based on single-cell DNA sequencing data. The model is well suited for full Bayesian analysis by introducing prior probabilities for key parameters as well as *maximum a posteriori* estimation using expectation maximization algorithm. Our model outputs distance between two cells, simultaneously taking all the other cells into account. In particular, the model contains variables associated with pairs of loci, of which one is homozygous and the other heterozygous, and has the capacity to perform Bayesian probabilistic read phasing. By applying a fast distance based method, such as FNJ, to the estimated distance, a cell lineage tree can be obtained. In contrast to MCMC based methods, FNJ can easily handle data sets with tens of thousands of taxa. The high accuracy of the so obtained method, called SCuPhr, is shown in studies of several synthetic data set.

1 Introduction

That each individual originates from a single cell is, of course, common knowledge, but the fascinating implication that the vast number of subsequent cell divisions defines a cell lineage tree (CLT), with the extant cells as leaves, is far less recognized. Nevertheless the CLT provides a means to answer important questions concerning classical cell types, tissues, and the genetic diversity of our organs. At this point it may seem speculative, but it is conceivable that life style and environmental factors can impoverish an organ's genetic diversity. Such impoverishment may limit its ability to respond to future challenges and, thereby, constitute a risk factor for disease. Single-cell sequencing in conjunction with CLT reconstruction algorithms will allow us to initiate studies of this nature.

Tumor tree reconstruction from single-cell data is a closely related computational problem that has received substantially more attention (see for e.g., Navin et al. (2011); Xu et al. (2012); Zafar et al. (2017); Singer et al. (2018)). However, vast majority of these methods are based on algorithms originally developed for analysis of gene evolution and it is not clear whether the underlying assumptions behind these algorithms can be carried over to the analysis of single-cells. For example, the prevalence of copy-number variation (CNV) in tumor cells makes the CLT reconstruction problem differ substantially from the tumor tree reconstruction problem. In fact, the presence of CNVs make it difficult to reconstruct CLT based only on point mutations.

The main challenge, in particular when analyzing closely related cells, is identification of point mutations. In order to maximize the genetic materials (i.e., chromatids) from which fragments are sequenced, or equivalently, to minimize the probability of drop-out events, DNA amplification step is an essential part of single-cell DNA sequencing (scDNA-seq) of healthy cells (for a review of single cell sequencing methods, see for example, Gawad et al. (2016)). Allelic drop-outs (ADOs) refer to the phenomenon where fragments

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

from one of the chromatids do not get sequenced. The ADOs are particularly troublesome, partly because they can obscure point mutations and partly because they are hard to identify. Furthermore, amplification step itself may introduce errors, where the correct nucleotide is replaced by another nucleotide. This has an effect similar to a somatic mutation or a sequencing error. The complexity of scDNA-seq and various sources of error introduced during amplification and sequencing steps mean that it would be profitable to develop an integrated analysis framework that can capture them.

Our work is centered around analyzing spontaneous somatic mutations in cells in a healthy tissue to reconstruct the CLT. Somatic mutations can arise due to chemical instability, errors during DNA replications, and environmental factors (Lodish et al., 2000). The average somatic mutation rate in humans are about 10^{-9} mutations per locus per cell division (Lynch, 2010), which is equivalent to about three mutations per cell division on average. Our key contribution is in developing a novel probabilistic model that produces an estimate of the distance matrix between the single cells. The CLT is then, constructed from the estimated distance matrix. Our probabilistic model admits efficient estimation of the distance matrix via dynamic programming algorithm. One of the key features of our methodology is that it examines a pair of loci at close proximity to infer the cell similarities. A similar strategy has been used in Zhou et al. (2017), where pairs of mutation sites (two closely located SNVs) are considered to infer tumour subclone phylogeny. By examining pair of loci in close proximity, we are able to phase the reads, which is instrumental in identifying the allelic dropouts. We propose an efficient strategy to search for the pair of sites efficiently using only the standard tools from bioinformatics.

2 Background

In this section, we describe the problem of interest, related research, and introduce common assumptions as well as notation.

2.1 Problem Setting

We assume that we have access to scDNA-seq data as well as the bulk DNA-seq data from a healthy tissue. The generative process that leads to the scDNA-seq dataset is as follows. First, we start with a single cell at the root of CLT. This cell undergoes number of cell divisions, where each cell division may result in spontaneous somatic mutations acquired by the daughter cells. Figure 1 illustrates this process. In Figure 1 (a), the somatic mutations are represented by colored shapes and the extant cells are indicated by the black dots (appearing at the leaves of the tree). Figure 1 (b) shows mutation profile of each of the cells. Figure 1 (c) and (d) show the the sites harbouring mutations represented by colors green and yellow. On the left is the values taken by the ancestor cell before any mutation is acquired; it is homozygous with cytosine at both chromatids in Figure 1 (c). The mutation acquired replaces cytosine with thymine, as can be seen from cells 2, 3, 4, and 5 that inherit this mutation. Similarly, the ancestor cell is homozygous adenine at the locus harboring the mutation in Figure 1 (d), but mutated to cytosine for cells 2 and 3. In reality, the nucleotide bases of the ancestor cell is not observed. We impute this value using the bulk DNA-seq data. To see why this imputation is reasonable, recall that we are interested in reconstructing CLT of healthy cells and that bulk data is an average of large number of cells that have, presumably, not acquired this mutation. Therefore, we expect that large portion of the cells in the bulk sample would represent the value taken by the ancestor cell.

Accurate reconstruction of CLT is challenging due to various sources of noise associated with the single cell sequencing technology. Each of the existing whole genome amplification (WGA) methods has a different coverage level, allelic dropout rate, and false-positive error. Multiple displacement amplification (MDA) has high but uneven coverage. False positive error rate at about 10^{-5} per base (Esteban et al., 1993). Multiple annealing and looping based amplification cycles (MALBAC) has high coverage with low allelic dropout rate; however, it has high error rates because of the polymerase used in the process (Wei et al., 2017). Degenerate oligonucleotide primer polymerase chain reaction (DOP-PCR) has low coverage across the genome and has high error rates due to the polymerases (Gawad et al., 2016). In addition, sequencing errors can creep in; single nucleotide substitution errors range from 10^{-8} to 10^{-1} depending on the sequencing technology (Fox et al., 2014). These errors are introduced in a random fashion and hence, justifies development of a probabilistic framework.

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

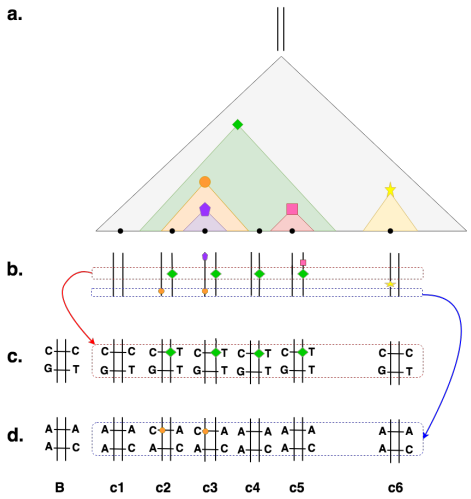


Figure 1: Illustration of cell lineage tree obtained through cell divisions and acquisition of somatic mutations.

2.2 Relevant Work

There are two recent papers featuring probabilistic analysis of single cells that are relevant to our work: Monovar (Zafar et al., 2017) and SCIΦ (Singer et al., 2018). Monovar serves a slightly different purpose of variant calling. SCIΦ has similar aim as our work in that they infer the CLT from sc-DNA seq data. A main difference is that our method estimates the distance matrix, from which CLT can be reconstructed efficiently using method such as (Saitou and Nei, 1987), whereas SCIΦ explores the space over the tree using MCMC move. Furthermore, SCIΦ is developed for inferring CLT from cancer cells and they attribute considerable effort into incorporating variation in zygosity to the likelihood model. Our focus is on inferring CLT from healthy cells where the number of mutations may be scarce. Hence, we attribute much efforts towards developing efficient strategy to hunt for candidate mutations. In particular, we developed *site pair model*, which allows to incorporate read phasing to improve sensitivity to allelic dropout. There are further differences in the finer details. For example, our method estimates the distance matrix by marginalizing out the effect of allelic dropout and amplification error whereas SCIΦ seeks to infer them. Our method and Monovar utilizes phred scores into the likelihood model. Because SCIΦ is focussed on cancer samples, their likelihood model is defined on the count of reference/alternate and hence, the phred scores do not enter into the equation. The three methods are similar in seeking to detect statistical signal by pooling information from all of the cells; however, the three methods have slightly different purposes and adopt vastly different strategies towards inference.

Phasing of reads from single cells is considered by Conbase (Hard et al., 2018). Conbase serves a purpose similar to Monovar in that their method is focussed primarily on making variant calls from the single cell DNA-seq data. Conbase differs from our method as well as Monovar and SCIΦ in that it is a voting based algorithm (i.e., it is not likelihood based). Therefore, it does not render itself to inference of model parameters via standard learning algorithms. Also, Conbase constructs a distance matrix from the variant calls as a post processing step, from which hierarchical clustering method is performed to construct the tree. On the contrary, we construct the distance matrix as part of all-encompassing probabilistic framework capturing random amplification, ADO, sequencing error, and the reads (i.e., evidence).

2.3 Assumptions

Throughout the paper, we will make the following assumptions.

Assumption 1. *(A1) We assume infinite sites model.*

Assumption 2. *(A2) The nucleotides at different sites evolve independently.*

Assumption 3. *(A3) The tissue has no account of copy number variations (i.e., assume diploidy).*

(A1) implies that no site is hit with mutation twice, i.e., the probability that a site is hit with two mutations is negligible. (A2) is a standard assumption in the analysis of DNA-seq data; see for example, Zafar et al.

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

(2017). (A3) ensures that somatic mutations provide sufficient evidence for constructing the CLT. We introduce additional notation as they are needed later on in the exposition.

2.4 Notation

We let $\Sigma = \{A, C, G, T\}$ denote the set of nucleotides. We denote the number of observed single cells by C. The reads for cell c at locus s is denoted $\boldsymbol{R}_c^s$ for $c = 1, ..., C$, and the associated quality scores denoted by $\boldsymbol{Q}_c^s$. We will assume that quality scores have been converted from Phred scores into probabilities. The number of reads for site s for cell c is denoted by $L_c^s = |\boldsymbol{R}_c^s|$. We index individual read by $R_{c,l}^s \in \Sigma$ for $l = 1, ..., L_c^s$. The bulk sequence at site s is denoted by $B^s = (B^{s,1}, B^{s,2}) \in \Sigma^2$ where the second superscript identifies the chromosome. We denote the true genotype of cell c at locus s by $X_c^s = (X_c^{s,1}, X_c^{s,2}) \in \Sigma^2$.

We may index the reads and quality scores at a pair of loci, for example, if $\boldsymbol{s} = (s, s')$, then $\boldsymbol{R}_c^{\boldsymbol{s}}$ denotes the reads at loci s and s' for cell c. For a pair of sites $\boldsymbol{s} = (s, s')$, $R_{c,l}^{\boldsymbol{s}} \in \Sigma^2$, and similarly, $B^{\boldsymbol{s},1}, B^{\boldsymbol{s},1} \in \Sigma^2$. We will adopt the standard practice from Statistics where the upper case letters are used to denote the random variables and the lower case letters are used to denote the realized values of the random variables. We denote the probability of amplification error by p_{ae}, probability of mutation by p_m, and probability of allelic dropout by p_{ado}. We will denote these probabilities by $\boldsymbol{p} = (p_{ae}, p_m, p_{ado})$.

3 Model

In this section, we describe the probabilistic framework based on *site pair model* (SPM). SPM can be viewed as efficient search strategy for detecting somatic mutation.

3.1 Site Pair Model

The site pair is identified as follows. First, we identify germline SNP by comparing the bulk data to the reference genome. This can be carried out using standard variant caller such as GATK (DePristo et al., 2011), FreeBayes (Garrison and Marth, 2012), VarScan 2 (Koboldt et al., 2012) and Platypus (Rimmer et al., 2013). Denote the number of sites identified by $S > 0$. For an average human, S can be in the order of hundreds of thousands to millions. For each site $s = 1, ..., S$, we check if it satisfies one of the following criteria:

1. B^s is heterozygous and $\exists s' : B^{s'}$ is homozygous and there exists a read that covers both s and s' or
2. B^s is homozygous and $\exists s' : B^{s'}$ is heterozygous and there exists a read that covers both s and s'.

Note that the key feature of this search strategy is in observing the read phasing, which is important in identification of allelic dropouts. For each pair of sites (s, s'), we examine the single cell dataset to further narrow down the candidate set to be considered. In particular, we amalgamate the entire scDNA-seq data and if we observe heterozygosity at either s or s', we add (s, s') to the candidate set $\mathcal{S}$. There are various strategies for implementing this in practice, for example, the method proposed in Zafar et al. (2017).

Figure 1 (c) is an example of such site pair. The bulk denoted by B shows one homozygous site and one heterozygous site. The homozygous site harbors the mutation and it is carried by cells 2, 3, 4 and 5 whereas cells 1 and 6 do not. The implication of the infinite sites assumption is that the mutation hits at most one site, either s or s'.

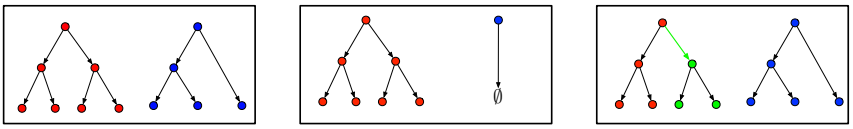
3.2 Amplification Tree

We model the amplification process using Pólya urn process (Mahmoud, 2008). For each candidate pair $\boldsymbol{s} = (s, s') \in \mathcal{S}$, we consider an urn initialized with $X^{\boldsymbol{s},1}, X^{\boldsymbol{s},2}$. The amplification process is viewed as drawing one of the chromatids and placing it back with additional copy of the chosen chromatid. But with some probability, p_{ae}, the copy generated is erroneous. In the traditional urn scheme, we can view $X^{\boldsymbol{s},1}, X^{\boldsymbol{s},2}$ as red and blue balls; with probability p_{ae}, we may put a ball of a different color along with the chosen ball and with probability $(1 - p_{ae})$, we put a ball of the same color as the chosen ball. Note that since $X^{\boldsymbol{s},1}, X^{\boldsymbol{s},2} \in \Sigma^2$, the total number of variants (i.e., colors) is limited to $|\Sigma^2| = 16$. We make the following assumption:

Assumption 4. *(A4) We assume that at most one amplification error occurs per site pair.*

The probability of having two amplification errors occur at a site is close to zero; therefore, such cases are negligible. As we will see in Section 4.2, this assumption enables us to make precise statement over the number of fragments from each genetic chromatid. Figure 2 provides numerous examples of amplification

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.



(a) (b) (c)

Figure 2: Illustration of amplification trees of a cell at a site pair. (a) $X^{\boldsymbol{s},1}$ is amplified 4 times, $X^{\boldsymbol{s},2}$ is amplified 3 times; no amplification error is introduced. (b) $X^{\boldsymbol{s},1}$ is amplified 4 times without amplification error whereas $X^{\boldsymbol{s},1}$ is dropped out (ADO) and is not amplified. (c) Amplification error takes place along an edge of the tree, introducing a third genotype $X^{\boldsymbol{s},3}$ represented by color green. Further, amplification of $X^{\boldsymbol{s},3}$ results in the increase of genetic material of this type.

tree. In this figure, we represent the true genotypes, $X^{\boldsymbol{s},1}, X^{\boldsymbol{s},2}$, by red and blue nodes at the roots. As can be seen from the figure, amplification process can be represented by a binary tree, one for each chromatid. Note that amplification error taking place early in the process increases the odds of false positive as large number of erroneous genetic material may be present at the sequencing step.

3.3 Graphical Model

The graphical model that combines various components introduced thus far is given in Figure 3. The graphical model is defined for each site pair, $\boldsymbol{s} \in \mathcal{S}$; the subscript $\boldsymbol{s}$ is omitted for brevity. The observed quantities are the bulk data B and the single cell reads, $\boldsymbol{R}_c$, along with the quality scores $\boldsymbol{Q}_c$ for each cell $c = 1, ..., C$. The observed quantities are shaded in gray in the graphical model. Recall that the bulk represents the values taken by the ancestor cell as described in Section 2.1. The remaining variables are unobserved.

Z denotes the mutation type taken at candidate pair $\boldsymbol{s}$, with $\boldsymbol{\alpha}$ denoting the parameter for the distribution over Z. We assume that Z follows Dirichlet-Categorical distribution with concentration parameter $\boldsymbol{\alpha}$. Note that we do not explicitly model the CLT, partly because the observed single cells are only a subset of the entire cell population. Therefore, we introduce a collection of indicator variables, G_c, where $G_c = 1 \Rightarrow X_c^{\boldsymbol{s}} = Z$ and $G_c = 0 \Rightarrow X_c^{\boldsymbol{s}} = B$ (i.e., the value taken by the ancestral cell). In Section 4.1, we will see that these indicator variables play an instrumental role in constructing the distance matrix.

The allelic dropout is modelled by a pair of indicator variables $D_{c,1}, D_{c,2}$ where $D_{c,1} = 1$ indicates that chromatid 1 has allelic dropout and similarly $D_{c,2}$ for chromatid 2. The amplification process produces the DNA fragments, denoted π_c, via Pólya urn process and this process is defined by conditioning on the variables $B, Z, G_c, D_{c,1}, D_{c,2}, L_c$. Recall that by (A4), the number of fragments is limited to be at most 3, the original two genotypes plus one introduced by an amplification error. Therefore, the fragments are represented by $\pi_c = ((F_{c,1}, \lambda_{c,1}), (F_{c,2}, \lambda_{c,2}), (F_{c,3}, \lambda_{c,3}))$, where $F_{c,j} \in \Sigma^2$, $\lambda_{c,j} \in \{0, ..., |R_c|\}$ for $j = 1, 2, 3$ with the constraint that $\sum_j \lambda_{c,j} = |\boldsymbol{R}_c|$.

Finally, the fragments are passed in through the sequence analyzer, and produce the reads as well as the quality scores, $(\boldsymbol{R}_c, \boldsymbol{Q}_c)$. We also consider the number of reads, $L_c = |\boldsymbol{R}_c|$ as observations produced by the sequence analyzer. Note that in a rare chance that allelic dropout hits both of the chromatids, it is possible for $\lambda_{c,j} = 0$ for $j = 1, 2, 3$. In this case, there would not be any observed reads, i.e., $|\boldsymbol{R}_c| = 0$.

4 Inference

We describe a procedure to efficiently marginalize over the latent variables. To focus the attention of the algorithm for marginalizing the latent variables, we keep the model parameters $\boldsymbol{\alpha}, \boldsymbol{p}$ fixed. The model parameters can be estimated using standard method such as expectation maximization algorithm (Dempster et al., 1977). See Appendix 7.3 for the extended version of the graphical model and the inference where we introduced prior probabilities for p_m and p_{ado}.

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

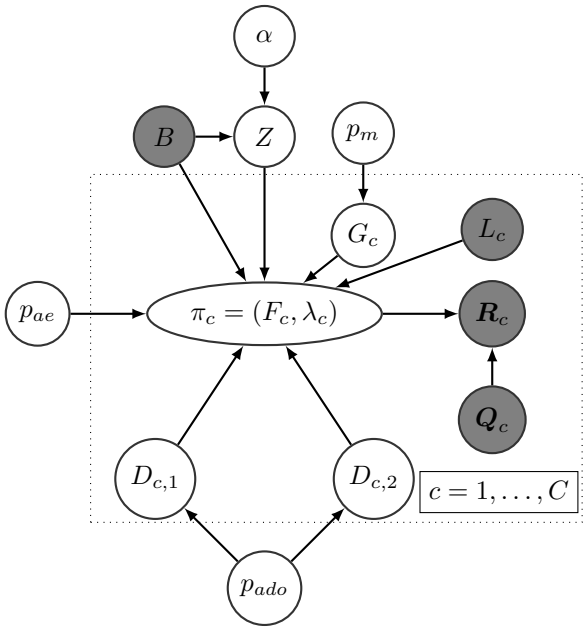


Figure 3: Graphical Model

4.1 Objective: Measuring Cell Similarity

For a pair of cells c, c', our objective is to compute the probability

$$P(G_c^{\boldsymbol{s}} = G_{c'}^{\boldsymbol{s}} | B^{\boldsymbol{s}}, \boldsymbol{R}^{\boldsymbol{s}}, \boldsymbol{Q}^{\boldsymbol{s}}) = P(G_c^{\boldsymbol{s}} = 1, G_{c'}^{\boldsymbol{s}} = 1 | B^{\boldsymbol{s}}, \boldsymbol{R}^{\boldsymbol{s}}, \boldsymbol{Q}^{\boldsymbol{s}}) + P(G_c^{\boldsymbol{s}} = 0, G_{c'}^{\boldsymbol{s}} = 0 | B^{\boldsymbol{s}}, \boldsymbol{R}^{\boldsymbol{s}}, \boldsymbol{Q}^{\boldsymbol{s}}), \tag{1}$$

where $\boldsymbol{R}^{\boldsymbol{s}}$ denotes the reads from all cells at site pair $\boldsymbol{s}$. The above and (A2) naturally lead to a measure of similarity between any pair of cells:

$$d(c, c') = \sum_{\boldsymbol{s}} P(G_c^{\boldsymbol{s}} = G_{c'}^{\boldsymbol{s}} | B^{\boldsymbol{s}}, \boldsymbol{R}^{\boldsymbol{s}}, \boldsymbol{Q}^{\boldsymbol{s}}). \tag{2}$$

Therefore, the key to efficiently computing the similarity between cells lies in Equation 1. We utilize the conditional independence statements extracted from the graphical model to derive an efficient marginalization machinery. First, we decompose the distribution of the latent variables given the observations as follows (where the superscript $\boldsymbol{s}$ is omitted for brevity):

$$P(z, \boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{g}, \boldsymbol{\pi} | b, \boldsymbol{r}, \boldsymbol{q}, \boldsymbol{l}, \boldsymbol{\alpha}, \boldsymbol{p}) \propto P(z | \boldsymbol{\alpha}, b) \prod_{c=1}^{C} [P(g_c | p_m) P(d_{c,1} | p_{ado}) P(d_{c,2} | p_{ado}) \times P(\pi_c | z, b, g_c, d_{c,1}, d_{c,2}, p_{ae}, l_c) P(\boldsymbol{r}_c | \pi_c, \boldsymbol{q}_c)],$$

where $\boldsymbol{g} = (g_1, ..., g_C), \boldsymbol{d}_1 = (d_{1,1}, ..., d_{C,1})$, and so on. Note that it suffices to derive a marginalization algorithm over all of the latent variables since, to compute the similarity between any two pair of cells c', c'', we just fix $G_{c'} = g', G_{c''} = g''$ and marginalize out all of the remaining latent variables.

The conditional independence of the cells given the global variables $B, Z, \boldsymbol{p}$ allows us to push the

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Algorithm 1 Dynamic Programming algorithm for precomputing the read probabilities

```
function retrieve(T, i, j, k):
  if i < 0 or j < 0 or k < 0 then
    return 0
  else
    return T[i, j, k]
  end if
function precompute(r, q, f_1, f_2, f_3):
  L ← |r|
  T ← allocate_array(L, L, L)
  T[0, 0, 0] ← 1
  for i = 0, ..., L do
    for j = 0, ..., L − i do
      for k = 1, ..., L − i − j do
        y_1 ← P(r_k|f_1, q_k) × retrieve(T, i − 1, j, k)
        y_2 ← P(r_k|f_2, q_k) × retrieve(T, i, j − 1, k)
        y_3 ← P(r_k|f_3, q_k) × retrieve(T, i, j, k − 1)
        T[i, j, k] ← y_1 + y_2 + y_3
      end for
    end for
  end for
  return T
```

summation inside the product:

$$
\begin{aligned}
P(\boldsymbol{r}|b,\boldsymbol{q},\boldsymbol{l},\boldsymbol{\alpha},\boldsymbol{p}) &= \sum_{z}\sum_{\boldsymbol{d}_1}\sum_{\boldsymbol{d}_2}\sum_{\boldsymbol{g}} P(z,\boldsymbol{d}_1,\boldsymbol{d}_2,\boldsymbol{g},\boldsymbol{\pi},\boldsymbol{r}|b,\boldsymbol{q},\boldsymbol{l},\boldsymbol{\alpha},\boldsymbol{p}) \\
&= \sum_{z} P(z|\boldsymbol{\alpha},b)\times \\
&\prod_{c=1}^{C}\Bigg[\sum_{g_c} P(g_c|p_m)\sum_{d_{c,1}} P(d_{c,1}|p_{ado})\sum_{d_{c,2}} P(d_{c,2}|p_{ado})\ \times \\
&\Bigg\{\sum_{\pi_c} P(\pi_c|z,b,g_c,d_{c,1},d_{c,2},p_{ae},l_c)P(r_c|\pi_c,q_c)\Bigg\}\Bigg]. \qquad (3)
\end{aligned}
$$

Since $g_c, d_{c,1}, d_{c,2} \in \{0,1\}$, summing over these variables is efficient. In the following section, we describe how to compute Line 3.

4.2 Marginalization of Fragments

The pre-requisite step is to first compute $P(r_c|\pi_c, q_c)$ for all possible values of π_c using dynamic programming. First, note that for a specific fragment $f_{c,j}$, we can compute a single read's probability by,

$$
P(r_{c,l}|f_{c,j},q_{c,l}) = \begin{cases} (1-q^1_{c,l})(1-q^2_{c,l}) & \text{if } r^1_{c,l}=f^1_{c,j} \text{ and } r^2_{c,l}=f^2_{c,j} \\ (1-q^1_{c,l})\frac{q^2_{c,l}}{3} & \text{if } r^1_{c,l}=f^1_{c,j} \text{ and } r^2_{c,l}\neq f^2_{c,j} \\ \frac{q^1_{c,l}}{3}(1-q^2_{c,l}) & \text{if } r^1_{c,l}\neq f^1_{c,j} \text{ and } r^2_{c,l}=f^2_{c,j} \\ \frac{q^1_{c,l}}{3}\frac{q^2_{c,l}}{3} & \text{if } r^1_{c,l}\neq f^1_{c,j} \text{ and } r^2_{c,l}\neq f^2_{c,j}. \end{cases} \qquad (4)
$$

We can compute $P(r_c|\pi_c, q_c)$ efficiently using dynamic programming algorithm in Algorithm 1. The time complexity of this algorithm is $O(L^3)$, where L is the number of reads. Note that this step can be carried out as a pre-computation step for each site pair in parallel. We consider all possible partitions of reads into fragments with respect to λ_c such that $\sum_j \lambda_{c,j} = L$.

With the read probabilities pre-computed, it remains to evaluate the probability of observing a fragment: $P(\pi_c|z, b, g_c, d_{c,1}, d_{c,2}, p_{ae}, l_c)$. We have constructed the look-up table for each possible configuration of the latent variables in Table 1 in the Appendix. Therefore, to compute Line 3, we only need to look up the

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

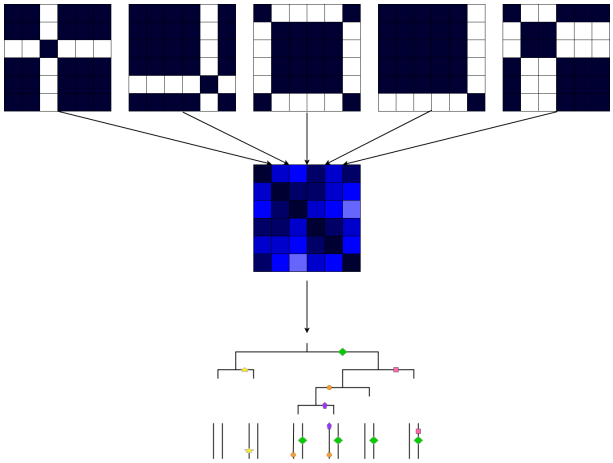


Figure 4: Similarity matrices at each site is combined to form global similarity matrix. We aim to infer the cell lineage tree from the similarity matrix.

value for $P(\pi_c = (f_c, \lambda_c)|z, b, g_c, d_{c,1}, d_{c,2}, p_{ae}, l_c)$ and look up the value corresponding to configuration λ_c in the dynamic programming table.

4.3 Cell Lineage Tree Reconstruction

Operating under the site independence assumption (A2), we generate a cell similarity matrix independently and combine these matrices to obtain a global similarity matrix. This matrix encodes the cell to cell similarities across all positions in the genome. We use off-the-shelf algorithm such as (Saitou and Nei, 1987; Elias and Lagergren, 2005) to generate the cell linage tree of the sequenced cells. Note that other distance based phylogenetic tree reconstruction methods can also be used.

Figure 4 illustrates this process. Pairwise similarities of cells form local site pair is combined via averaging to produce global similarity matrix. We only consider the positions where two cells co-exist (both of the cells have reads at the position pair).

5 Results

The synthetic data simulation consists of two parts; generating the cell lineage tree and simulating the reads of single cells. For the first part, we simulated the cell lineage trees with varying number of observed cells ($C = \{20, 50, 100\}$). In some of the cases, we generated trees with exactly C leaves and observed all the leaves. In other cases, we generated larger CLTs (where the number of leaves $> C$) and observed C randomly selected leaves. Second, we introduced mutations to the edges of the CLT. We used the assumption that every cell division introduces spontaneous mutations. In order to analyze the effect of the number of mutations, we used fixed number of mutations per cell division ($2, 3, 4$ and 5). We randomly partitioned mutations to the daughter cells, but we made sure that each daughter cell has at least one mutation.

For the second part, we generated reads of single cells based on the mutations in CLT. With two different allelic dropout probabilities ($p_{ado} = \{0.1, 0.3\}$), we first determined the allelic dropout statuses of cells. In half of the simulations, we fix the number of reads per allele to 10. For a pair of sites, each cell has $|r_c| = \{20, 10, 0\}$ reads if there is no allelic dropout, one allelic dropout or two allelic dropouts respectively. In the other half of the simulations, we varied the number of reads per allele (based on Poisson Distribution). We set the amplification error probability to $p_{ae} = 0.001$ in all of the simulations.

We measured the accuracy of the tree reconstruction by comparing real and inferred CLTs with Robinson-Foulds (RF) distance. We normalized the distance by dividing RF with the total number of edges in both of the trees. Figure 5 shows the results of synthetic data experiments based on normalized RF. In the plots, y-axis corresponds to normalized RF and x-axis corresponds to the number of spontaneous mutations per cell division. Blue and red bars represent different allelic dropout probabilities. From the figure, we make the following observations. (i) Increasing the number of mutations per cell division increases the number of informative pair of sites and decreases normalized RF. (ii) Our method performs better with lower allelic

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

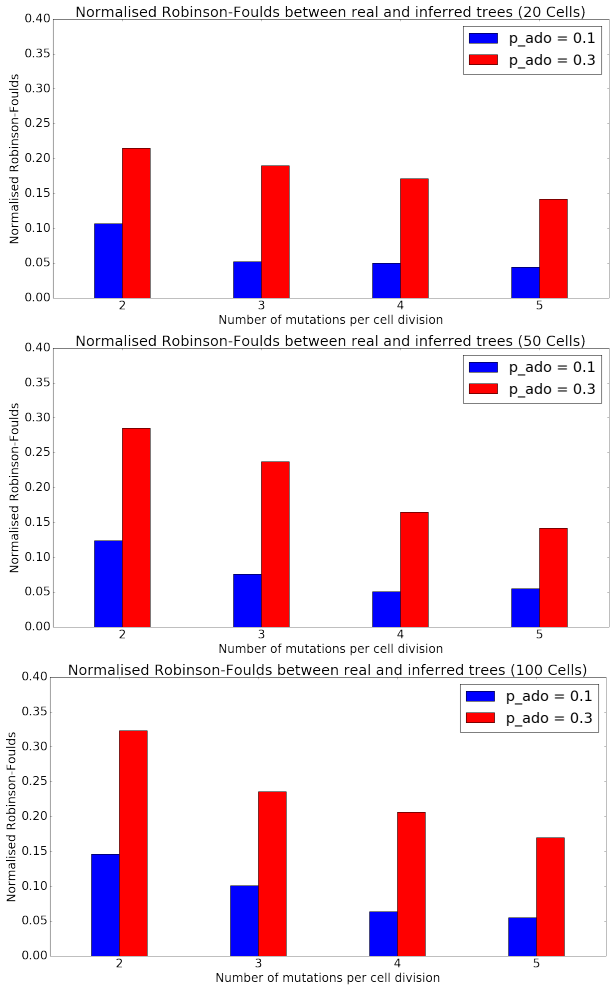


Figure 5: Results on simulated datasets. (a) $C = 20$. (b) $C = 50$. (c) $C = 100$.

dropout rate where the mutation information is not lost. However, if there are several mutations per cell division, even though some of the information is lost due to allelic dropouts, our method recovers the underlying structure. (iii) Increasing the number of observed cells increases the normalized RF, especially when there is high allelic dropout rate. The normalized RF decreases dramatically when more informative pair of sites are used.

We also show additional results of an experiment with 20 cells in Figure 6. Figure 6 (a) shows the real similarity matrix, inferred similarity matrix of cells and the absolute difference of similarity matrices. Matrices are constructed using 55 pair of sites. In Figure 6 (b), we show the change in the Frobenius norm of absolute difference matrix (absolute difference of estimated to the truth) when we combine the pair of sites one-by-one. As expected, introducing additional pair of sites helps our method to infer cell similarities and therefore, the underlying tree structure. We observed convergent behavior of the inferred similarity matrix to the real similarity matrix as the number of sites increased.

6 Conclusion and Discussion

We have introduced SCuPhr, a probabilistic framework to estimate cell distance matrix. We proposed a site pair model to efficiently search for mutations. Our model captures various sources of noise associated with sequencing of single cells. We have also developed an inference algorithm based on dynamic programming. From the distance matrix, we have reconstructed the CLT using off-the-shelf method Saitou and Nei (1987); Elias and Lagergren (2005). We have demonstrated its efficiency on simulated data.

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

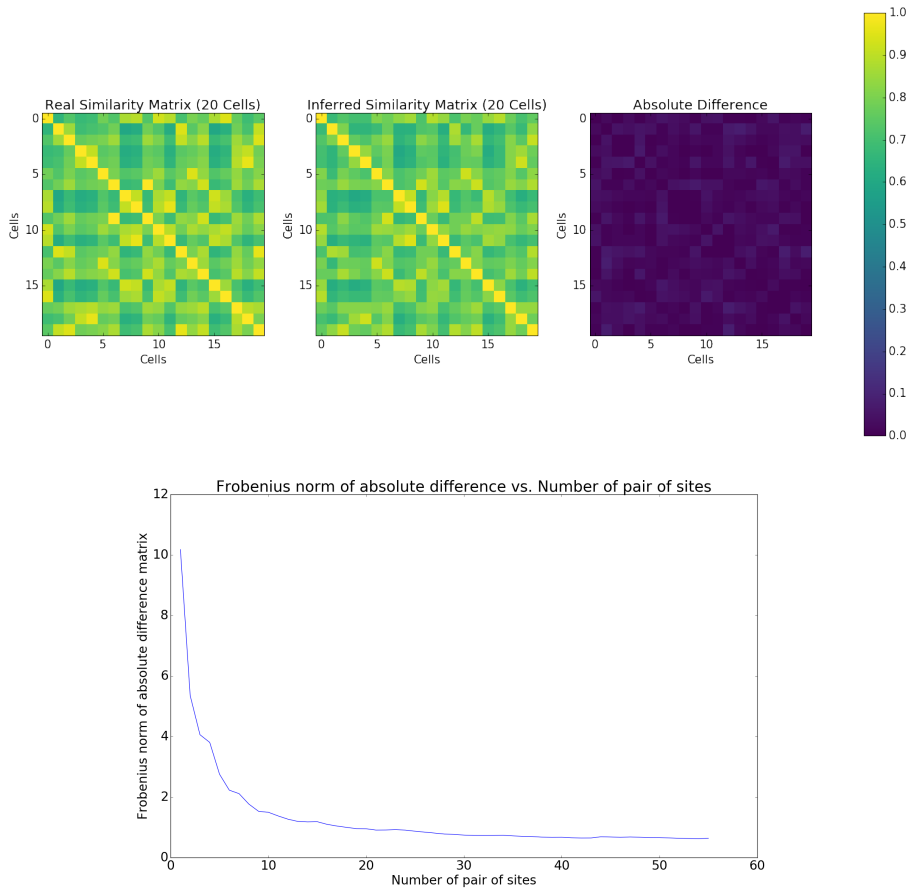


Figure 6: Results on a simulated dataset ($C = 20$). (a) From left to right; real similarity matrix of cells, inferred similarity matrix of cells, absolute difference of similarity matrices. (b) Frobenius norm of absolute difference matrix vs. the number of sites.

The future work remains to carry out real data analysis and comparison to similar, existing framework, namely, Monovar, Conbase, and SCIΦ (Zafar et al., 2017; Hard et al., 2018; Singer et al., 2018). There is potential for improvement regarding model parameter estimation. Due to potentially large number of site pairs, one may adopt stochastic gradient descent method where a subset of the site pair is selected at each iteration of EM algorithm (Bottou et al., 2016).

Acknowledgement

The authors would like to acknowledge support from Science for Life Laboratory. The computations were performed on resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project sens2018105.

References

Bottou, L., Curtis, F. E., and Nocedal, J. (2016). Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38.

DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A.,

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Del Angel, G., Rivas, M. A., Hanna, M., et al. (2011). A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491.

Elias, I. and Lagergren, J. (2005). Fast neighbor joining. In *International Colloquium on Automata, Languages, and Programming*, pages 1263–1274. Springer.

Esteban, J. A., Salas, M., and Blanco, L. (1993). Fidelity of phi 29 dna polymerase. comparison between protein-primed initiation and dna polymerization. *Journal of Biological Chemistry*, 268(4):2719–2726.

Fox, E. J., Reid-Bayliss, K. S., Emond, M. J., and Loeb, L. A. (2014). Accuracy of next generation sequencing platforms. *Next generation, sequencing & applications*, 1.

Garrison, E. and Marth, G. (2012). Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*.

Gawad, C., Koh, W., and Quake, S. R. (2016). Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics*, 17(3):175.

Hard, J., Al Hakim, E., Kindblom, M., Bjorklund, A., Demirci, I., Paterlini, M., Sennblad, B., Borgstrom, E., Stahl, P. L., Michaelsson, J., et al. (2018). Conbase: a software for discovery of clonal somatic mutations in single cells through read phasing. *bioRxiv*, page 259994.

Koboldt, D. C., Zhang, Q., Larson, D. E., Shen, D., McLellan, M. D., Lin, L., Miller, C. A., Mardis, E. R., Ding, L., and Wilson, R. K. (2012). Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research*, 22(3):568–576.

Lodish, H., Berk, A., Zipursky, S. L., Matsudaira, P., Baltimore, D., and Darnell, J. (2000). Mutations: types and causes. *Molecular Cell Biology*, 4.

Lynch, M. (2010). Evolution of the mutation rate. *TRENDS in Genetics*, 26(8):345–352.

Mahmoud, H. (2008). *Pólya urn models*. CRC press.

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

Rimmer, A., Phan, H., Mathieson, I., Lunter, G., and McVean, G. (2013). Platypus: A haplotype-based variant caller for next generation sequence data.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425.

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

Wei, Z., Shu, C., Zhang, C., Huang, J., and Cai, H. (2017). A short review of variants calling for single-cell-sequencing data with applications. *The international journal of biochemistry & cell biology*, 92:218–226.

Xu, X., Hou, Y., Yin, X., Bao, L., Tang, A., Song, L., Li, F., Tsang, S., Wu, K., Wu, H., et al. (2012). Single-cell exome sequencing reveals single-nucleotide mutation characteristics of a kidney tumor. *Cell*, 148(5):886–895.

Zafar, H., Tzen, A., Navin, N., Chen, K., and Nakhleh, L. (2017). Sifit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome biology*, 18(1):178.

Zhou, T., Sengupta, S., Mueller, P., and Ji, Y. (2017). TreeClone: Reconstruction of Tumor Subclone Phylogeny Based on Mutation Pairs using Next Generation Sequencing Data. *ArXiv e-prints*.

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

7 Appendix

7.1 Probability of Fragments $P(\pi_c|b, z, d_{c,1}, d_{c,2}, g_c, p_{ae}, |r_c|)$

Probability of fragments depends on the bulk genotype, mutation genotype, genotype indicator, allelic dropout indicators, amplification error probability and number of reads of the cell. We partitioned the probability into 8 cases based on the allelic dropout, amplification error and if there is an amplification error, which tree contains the amplification error. Columns of Table 1 shows the configurations of each case. The derivation of the tree-edge pair calculation is shown in the next section. Table 2 shows the fragment probabilities of each case. Note that in each case, the roots of amplification trees ($f_{c,1}$ and $f_{c,2}$) is determined by b, z, g_c; if $g_c = 0$ then the roots have bulk genotype, otherwise they have mutation genotype z. The genotype of alternating fragment ($f_{c,3}$) should be exactly one nucleotide different than its parent.

Table 1: Configuration

Cases	ADO	AE	$\lambda_c = (\lambda_{c,1}, \lambda_{c,2}, \lambda_{c,3})$	$pa(f_{c,3})$	Number of tree-edge pairs
1	1st	no	$(0, \|r_c\|, 0)$	-	-
2	1st	yes	$(0, \|r_c\| - \lambda_{c,3}, \lambda_{c,3})$	$f_{c,2}$	$\frac{2\|r_c\|!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
3	2nd	no	$(\|r_c\|, 0, 0)$	-	-
4	2nd	yes	$(\|r_c\| - \lambda_{c,3}, 0, \lambda_{c,3})$	$f_{c,1}$	$\frac{2\|r_c\|!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
5	no	no	$(\lambda_{c,1}, \lambda_{c,2}, 0)$	-	-
6	no	yes	$(\lambda_{c,1}, \lambda_{c,2}, \lambda_{c,3})$	$f_{c,1}$	$\frac{2(\lambda_{c,1}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
7	no	yes	$(\lambda_{c,1}, \lambda_{c,2}, \lambda_{c,3})$	$f_{c,2}$	$\frac{2(\lambda_{c,2}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
8	no	yes	$(\lambda_{c,1}, \lambda_{c,2}, \lambda_{c,3})$	$f_{c,2}$ or $f_{c,2}$	$\frac{2(\lambda_{c,1}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)} + \frac{2(\lambda_{c,2}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)}$

Table 2: Fragment Probability

Cases	$P(\pi_c\|b, z, d_{c,1}, d_{c,2}, g_c, p_{ae}, \|r_c\|)$
1	$(1-p_{ae})^{2(\|r_c\|-1)}$
2	$p_{ae}(1-p_{ae})^{2(\|r_c\|-1)-1}\frac{1}{6}\frac{2\|r_c\|!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
3	$(1-p_{ae})^{2(\|r_c\|-1)}$
4	$p_{ae}(1-p_{ae})^{2(\|r_c\|-1)-1}\frac{1}{6}\frac{2L_c!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
5	$\frac{1}{\|r_c\|-1}(1-p_{ae})^{2(L_c-2)}$
6	$\frac{1}{\|r_c\|-1}p_{ae}(1-p_{ae})^{2(\|r_c\|-2)-1}\frac{1}{6}\frac{2(\lambda_{c,1}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
7	$\frac{1}{\|r_c\|-1}p_{ae}(1-p_{ae})^{2(\|r_c\|-2)-1}\frac{1}{6}\frac{2(\lambda_{c,2}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)}$
8	$\frac{1}{\|r_c\|-1}p_{ae}(1-p_{ae})^{2(\|r_c\|-2)-1}\frac{1}{6}\frac{2(\lambda_{c,1}+\lambda_{c,3})!+2(\lambda_{c,2}+\lambda_{c,3})!}{\lambda_{c,3}(\lambda_{c,3}+1)}$

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

7.2 Counting Tree-Edge Pairs

We use the following tree structure to model amplification process:

- t-tree: A rooted tree with t leaves in which the inner vertices are labeled from $1, \ldots, t-1$ (which represents the amplification order).
- d-edge: An edge that has d number of leaves under it.
- t-tree d-edge pair: A tree with t leaves and d-edge.

Proposition 5. *The total number of t-trees is $C(t) = (t-1)!$.*

Proof. In order to find the total number of t-trees, we partition the inner vertices of the tree into left and right sub-trees and count the number of unique sub-trees.

$$\begin{aligned} C(t) &= \sum_{i=1}^{t-1} \binom{t-1-1}{i-1} C(i) \binom{t-i-1}{t-i-1} C(t-i) \\ &= \sum_{i=1}^{t-1} \binom{t-2}{i-1} C(i) C(t-i) \\ &= (t-1)! \end{aligned}$$

□

Proposition 6. *The total number of t-tree d-edge pairs is $C(t = d+k, d) = \frac{2(d+k)!}{d(d+1)}$ if $k \geq 0$.*

Proof. Let's start with the base case, where $k = 1$.

$$\begin{aligned} C(t = d+1, d) &= \binom{2}{1} \binom{t-1-1}{d-1} C(d) \\ &= 2(d-1)! \end{aligned}$$

Next step is:

$$\begin{aligned} C(t = d+k-1, d) &= 2(t-2)! \sum_{i=d}^{t-1} \frac{C(i,d)}{C(i)} \\ &= 2(d+k-3)! \sum_{i=d}^{d+k-2} \frac{C(i,d)}{C(i)} \end{aligned}$$

We change the position of terms and get the following equation:

$$\sum_{i=d}^{d+k-2} \frac{C(i,d)}{C(i)} = \frac{C(d+k-1,d)}{2(d+k-3)!}$$

Final step is:

$$\begin{aligned} C(t = d+k, d) &= 2(t-2)! \sum_{i=d}^{t-1} \frac{C(i,d)}{C(i)} \\ &= 2(d+k-2)! \left((\sum_{i=d}^{d+k-2} \frac{C(i,d)}{C(i)}) + \frac{C(d+k-1,d)}{C(d+k-1)} \right) \\ &= 2(d+k-2)! \left(\frac{C(d+k-1,d)}{2(d+k-3)!} + \frac{C(d+k-1,d)}{C(d+k-1)} \right) \\ &= (d+k) C(d+k-1, d) \\ &= (d+k)(d+k-1) \ldots (d+2) C(d+1, d) \\ &= \frac{2(d+k)!}{d(d+1)} \end{aligned}$$

□

bioRxiv preprint doi: https://doi.org/10.1101/357442; this version posted June 29, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

7.3 Model with Prior Probabilities

The graphical model with prior probabilities is shown below. We use Beta priors to mutation probability (p_m) and allelic dropout probability (p_{ado}).

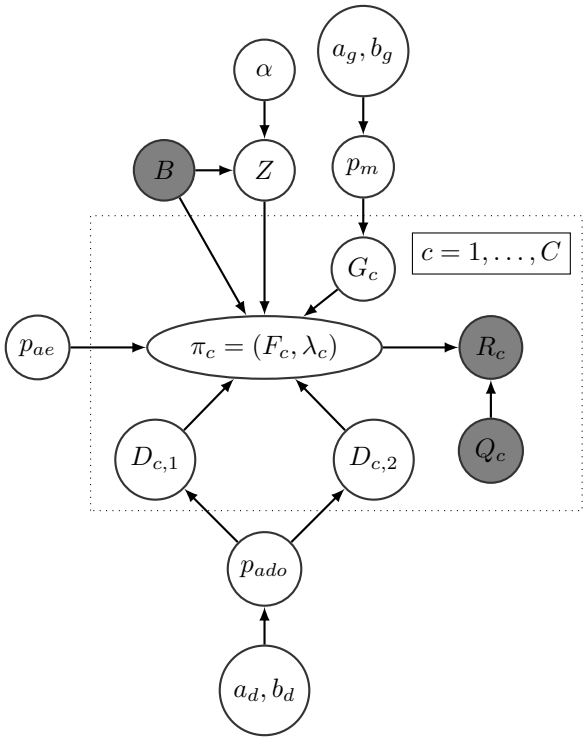


Figure 7: Graphical model with prior probabilities

Equation 5 shows the computation of distance between two pair of cells. The superscript $\boldsymbol{s}$ is omitted for brevity. B is the Beta function.

$$
\begin{aligned}
P(G_{c'} = g', G_{c''} = g'' | B, R, Q, \boldsymbol{\alpha}, a_g, b_g, a_d, b_d, p_{ae}) = \\
& \frac{1}{\mathcal{Z}} P(z|\boldsymbol{\alpha}, b) \\
& \sum_{m=0}^{C} \frac{B(a_g + m, b_g + C - m)}{B(a_g, b_g)} \sum_{\substack{\boldsymbol{g}: \\ \sum_{c=1}^{C} g_c = m \\ g_{c'} = g', g_{c''} = g''}} \prod_{c=1}^{C} \\
& \sum_{d_{c,1}} \frac{B(a_d + d_{c,1}, b_d + 1 - d_{c,1})}{B(a_d, b_d)} \\
& \sum_{d_{c,2}} \frac{B(a_d + d_{c,2}, b_d + 1 - d_{c,2})}{B(a_d, b_d)} \\
& \sum_{\pi_c} P(\pi_c | z, b, g_c, d_{c,1}, d_{c,2}, p_{ae}, |r_c|) P(r_c | \pi_c, q_c)
\end{aligned}
\tag{5}
$$