Tumor Phylogeny Simulator to Build Standardized Data Sets for Deconvolution Tools

Jesse Eaton, Murtaza Saif, and Shefali Umrania

December 5, 2017

Abstract

The ubiquity of cancer demands useful and widely distributed tools for characterizing the severity of a tumor. A majority of tumors are sequenced in bulk, and not on a single cell resolution. To analyze bulk tumor data, there exist many "deconvolution" tools that seek to infer the genetic profiles of populations of cells residing within a tumor along with their phylogenetic history. Unfortunately, each of these tools are highly specialized and require unique input formats and data types to run. In this project, we use a continuous time Markov model to simulate biologically relevant data sets for testing deconvolution tools. We show our simulator is capable of producing a variety of phylogenetic topologies using different input parameters and argue for their biological importance. We also provide a standard data set for which deconvolution tools can be tested. All code developed for this project is provided here https://github.com/sumrania/tum_sim and the standard data set is provided here https://cmu.box.com/s/thq5ba255f24av575ax5xhk9kvp2syve.

1 Introduction

Cancer can be defined as an umbrella term for the uncontrolled division of abnormal cells which can spread to various parts of the body. In most cases, the malignant growth of these cells can lead to the formation of heterogenous tumors which are comprised of several cell types, each with their own genetic profiles. Therefore, when patients undergo cancer treatment, the causal tumor sample is isolated and is then subsequently sequenced. The most accurate way of studying the genomic content of these cells would be to perform single-cell sequencing for each of the cell types. However, majority of the clinics rely on bulk sequencing, because single-cell sequencing is relatively more expensive, time consuming, and produces noisy data sets. In bulk sequencing, the genomic content is sequenced from a group of cells together. As a result, the sequenced reads could potentially align to any of the possible cell types which are present in the original tumor thus creating ambiguity in identifying the exact causal cell type. There are several tools [6][5][1][2][9][3][4] available which can "unmix" or "deconvolve" this bulk sequenced data to predict the different cell types present in the tumor cell. One of the subsets of these problems is known as the Copy Number Tree Mixture Deconvolution (CNTMD) problem [10] where the objective is to identify the phylogenetic tree that explains the copy number data from various samples of the tumor.

In this project, we are essentially trying to simulate a bulk sequenced tumor sample and represent it as a phylogenetic tree in order to build standardized data sets that can be used as input to deconvolution tools for tumor phylogeny research. Some of these tools use sequences which contain single nucleotide polymorphisms (SNPs) [4], some contain copy number variations (CNVs) [9], while others contain both [6][5][1][3]. One thing to keep in mind is that each of these tools take their own sequence read inputs, generate custom outputs and then simulate their own data which makes it very difficult to compare their efficiency. Previous attempts have been made in the past to create generalized phylogeny simulators [7][8] but have not covered all possible mutation events and can thus cannot be used to generate a standardized test data set. One simulator, PyVolve, uses a continuous time Markov model to build a tumor phylogeny with SNP mutations but fails to infer CNVs [8]. Another called Pomegranate infers SNPs and CNVs but does not maintain a genome so each CNV acts independently [7]. This project presents the foundation of a simulator tool that focuses on the difficult task of segmental copy number changes: duplications and deletions as well as segmental inversions while leaving point mutations for future work. These data sets can then be tested with the above mentioned softwares to solve the Copy Number Tree Mixture Deconvolution (CNTMD) problem in the future.

2 Methods

In order to create standardized data sets for the deconvolution tools, we propose creating a phylogenetic simulator that will use a continuous-time Markov Model (CTMM) to determine when and which mutations occur at each timestep. This not only helps us generate a matrix of copy number profiles for each cell type but it allows us to assign values to the percent of the total tumor that is made by each cell type. Formally, we sample a tree T with n nodes (representing distinct cell types) and $m = n - 1$ directed edges representing ancestral linkage between cell types from the set of all possible trees with n nodes by starting with a single root node and iteratively mutating leaf nodes, restricting transitions back to smaller trees. Our state set is the set of all possible trees with n nodes with all possible segmental mutations as edges. At each state q_i at time i, we define the set of possible transitions from tree T_i to be to any tree T_{i+1} of size $i + 1$ with T_i as a subgraph. The transition rates for any one node to gain a child are defined by the percent of the total tumor made by that node. We take advantage of the memoryless property for exponentials by assuming the waiting time between each mutation is exponentially distributed.

More simply put, the algorithm for simulating a tree T with $m = n - 1$ mutation edges and n nodes is as follows. Each node contains a genetic profile including the current copy number of each segment on each homologous chromosome across all chromosomes along with its transition history. Start with a single root node which is subsequently mutated. For any mutation, the left child is assumed to contain a single additional mutation than the parent while the right child will be identical to the parent. This ensures the set of leaves will contain all unique genetic profiles and mutating only leaves will cause the tree to be binary. At each mutation, the percent of cell types of the node being mutated is distributed to its children according to a Beta distribution (more on this at the end of the section). Leaf nodes are chosen to mutate based on how many individual cells match that cell type. Specifically, an exponential is drawn for each leaf with rate equal to the percent of total tumor made up by that leaf, and the event with minimum time takes place. The algorithm terminates once the tree has been mutated a user specified m number of times. All nodes with matching genetic profiles are subsequently collapsed so there is a final tree contains $n = m + 1$ nodes and m edges. Note this tree need not be binary.

Each node in the phylogenetic tree keeps track of the exact genetic profile of a cell type and the total number of cells with that genetic profile. The set of mutations we create are 1. CNV insertion, 2. CNV amplification and 3. CNV deletion. To keep a better track of mutations, we have constructed the phylogenetic tree in such a way that we restrict the mutations only on the left side of the tree and corresponding subtrees. Let us suppose that the first mutation happens in the left child of the root node, then the right child would essentially contain the exact genetic profile as the parent root node. As a result, over several simulated mutations, the leaves of the trees would be the populations of the individual cell types that comprise the original tumor sample. We then artificially "mix" these cells to produce mixed copy number values for CNVs. Our final task is to use this simulator to create a standardized data set with many simulated samples binned into simulated "patients" to be used by the tools above.

Theoretically speaking, the Copy Number Tree Mixture Deconvolution (CNTMD) problem can be broken down into the construction and analysis of three matrices. For the sake of convenience, we will refer to them as the **F**, **U** and **C** matrices which are related according to this equation:

$$\mathbf{F} = \mathbf{U} \cdot \mathbf{C}$$

U is a m x n matrix where m represents the number of samples and n represents the different cell types. Each row in the U matrix would therefore depict the different cell types which are present in that particular tumor sample. The core logic behind construction of the Usage matrix is to randomly sample cell types based on a Uniform distribution [0,1] but also associate an auxiliary probability of selecting that cell type. In our case, we sample the cell types using a uniform random generator [0,1] and the population of each cell type as obtained from the leaves of the phylogenetic tree. The Copy Number Matrix **C** is an n x s matrix, where n represents the number of cell types and s represents the number of segments. In order to collect copy number profiles for **C**, we iterate through all the leaves in our simulated phylogenetic tree. We then partition each of the chromosomes in such a way that the number of segments for a chromosome is equal across all leaves. The overall chromosomes are then concatenated so each row of the C matrix represents the copy numbers of all segments for a single cell type. Hence, each row contains a unique cell type that directly corresponds to each of the leaves in the tree. The above two matrices can be multiplied to obtain **F** which is a m x s matrix where m is the number of tumor samples and s is the number of segments. The deconvolution tools described earlier use **F** as input to generate phylogenetic trees.

The algorithms that have been used for coding the simulator, collecting copy number profiles, and percent of each cell type are search algorithms using tree traversal methods such as breadth first search and depth first search.

The time complexity for one iteration of our algorithm is $O(m^2)$. There are two rate limiting steps that produce this quadratic runtime. For the i^{th} step of mutating the tree, we look at all i leaves and pick the first mutation. All other operations are constant time. We mutate the tree m times so $\sum_{i=1}^{m} i = m \cdot (m+1)/2 = O(m^2)$. Additionally, upon creating the copy number matrix, we partition chromosomes into s total segments. Our algorithm iterates through each set of segments for each leaf. Given there are $m+1$ leaves and each leaf can have no more than $2 \cdot m + d$ segments where d is the number of chromosomes, each mutation creates 2 additional segments. We can see the total time to combine copy number segments is $(m+1) \cdot (2 \cdot m + d) = O(m^2)$ given $m \leq d$ which is usually the case.

The simulator has been designed in such a way as to allow the user to input a wide array of parameters including the number of samples s, number of allowed mutations m, mean length of a single mutation ℓ, options to choose for variance in probability for generating the mixed sample v, and options to choose for α and β values for selecting the beta distribution that determines the probability a mutated node will have a larger cellular population. The variance in probability for generating a mixed sample uses the multinomial distribution to create non perfect sample mixtures. Given the user input v, we calculate the appropriate number of trials to draw from a multinomial distribution where the categories are distinct cell types and the associated probabilities is the percent of cells carrying that cell type (usage). We determine the optimal number of trials t using the variance of the multinomial distribution to be $t = \frac{m}{v \cdot (m+1)^2}$. We additionally define a Beta distribution for weighting the probability that either the mutated or normal child of any mutated node will contain a larger portion of the percent of cell types. Providing large $\frac{\alpha}{\beta}$ increases the number of cells with a mutation while decreasing this increases the number of children not containing the mutation. $\alpha = \beta = 1$ creates a $Unif(0, 1)$ and adds no bias to either child.

3 Results

We simulated 288 data sets generated using a variety of parameters influencing the complexity and progression of a tumor. All data can be found on the CMU Box repo: https://cmu.box.com/s/thq5ba255f24av575ax5xhk9kvp2syve. Table 1 defines each parameter used along with the values for each parameter. Along with these parameters, we also created a small data set labeled "toy" and larger data set labeled "real". The toy data set contains five chromosomes of lengths $10000, 15000, 20000, 25000, 30000$ while the real data set contains the 22 human chromosomes excluding X and Y chromosomes.

Symbol	Description	Range
ℓ	mean length of a single mutation event	$\ell_{toy} \in \{100, 1000\}$, $\ell_{real} \in \{50000, 100000\}$
m	number of mutations	$m \in \{10, 50, 300\}$
s	number of samples	$s \in \{1, 3, 5, 10\}$
v	variance in probability of generating a perfectly mixed sample	$v \in \{0.002, 0.01, 0.05\}$
α	parameter to Beta distribution	$(\alpha, \beta) \in \{(1, 1), (5, 1)\}$
β	parameter to Beta distribution	

Table 1: Parameters for large generated data set. Draw from beta distribution determines the percent of cells that will mutate at each mutation. Smaller α (and larger β increases population of normal children while smaller β (and larger α) increase the population of mutated children.

To analyze the effects these parameters have on our simulated data, we visualize tumor progression through phylogenetic tree diagrams. Figure 1 shows a tree generated using our continuous time Markov model on a single chromosome of length 10,000 with parameter set $\ell = 1000$, $m = 30$, $\alpha = \beta = 1$. In this figure, all right children share an identical genetic profile to their parents while left children contain a single additional mutation. Information on the percent cell types is not given in these figures however this information can be inferred by observing which cells are mutating more often (see Figure 2). For Figure 1 through 4, each node is labeled with the L1 distance of its copy number profile to the root node's copy number profile (assumed to be normal $[2, 2 \ldots, 2]$). The color of nodes represents the time when the node appears and by extension, when the mutation occurs. Yellow indicates early mutations while more red coloring indicates later mutations.

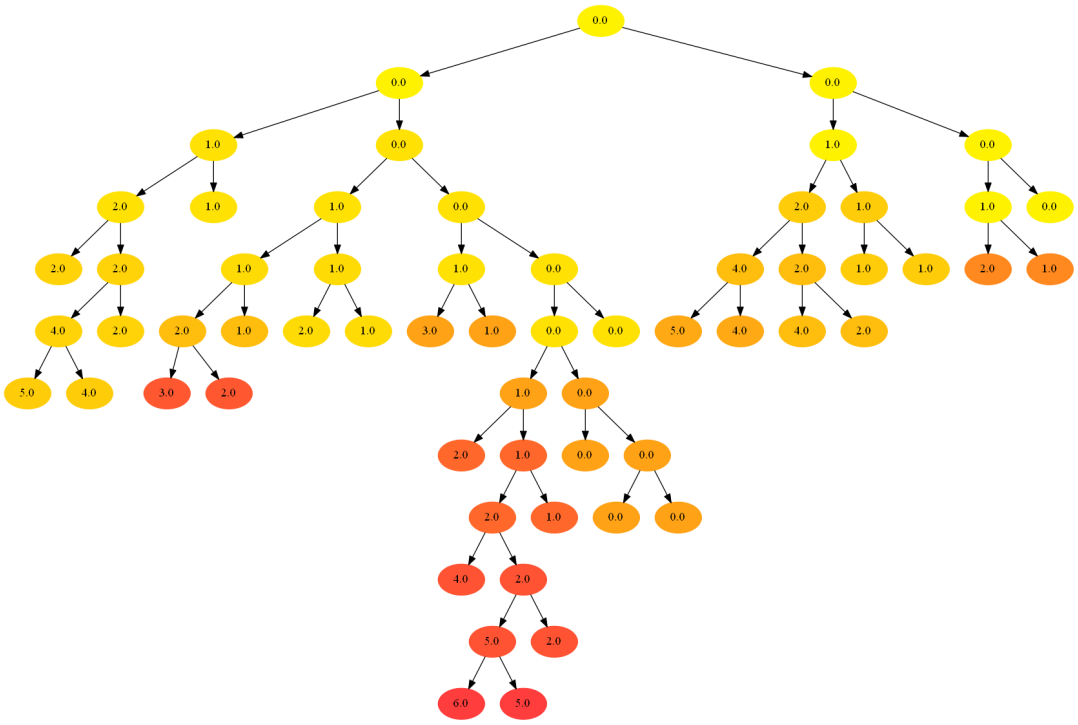
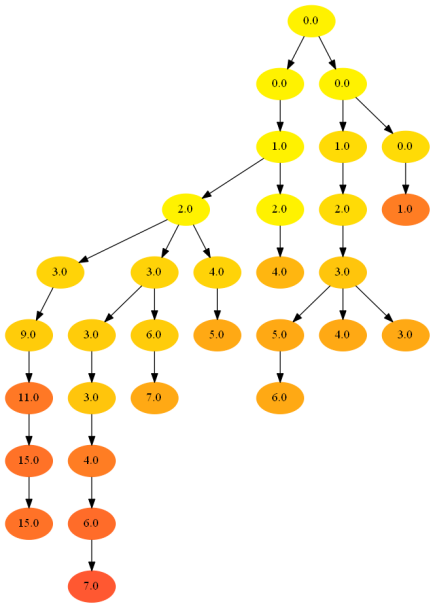


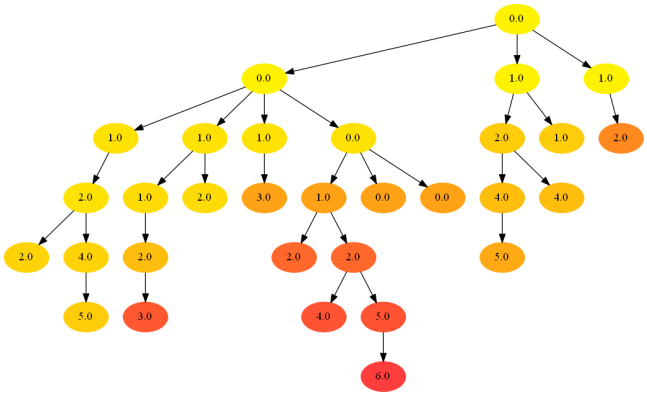
Figure 1: Phylogenetic tree generated by CTMM with no bias on percent population of mutated children. Values of nodes are L1 distances between their copy number profile and normal (root). Yellow color indicates earlier and red indicates later mutations. Single chromosome of length 10,000 with mean mutation size 1,000.

Although Figure 1 is an accurate representation of the way our data was generated from a continuous time Markov model, it contains some redundant information in that all right children are identical to their parents. To better visualize these phylogenies, we collapse all identical nodes. The collapsed version of Figure 1 is shown in Figure 2b.

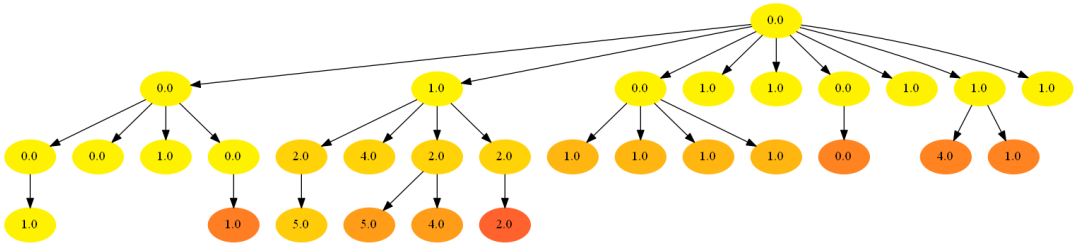
Next we visualize how changing the percent of cells that mutate at each mutation (changing α and β) affect phylogenetic topology. Figure 2a shows that increasing the percent of mutated cells causes those cells to accumulate more mutations relative to a uniform draw (Figure 2b). This makes intuitive sense as the rate of mutation depends on how many cells share a cell type. Conversely, we saw that decreasing the number cells that mutate (Figure 2c) cause mutations to more often occur close to the root. This phenotype is representative of mutations in normal cells or non aggressive tumors as each mutation is not adding any selective advantage. For subsequent analyses and our large data set, we chose to look at the more mutant tumors ($\alpha = 5$, $\beta = 1$ and $\alpha = \beta = 1$) since this is the more interesting and useful condition to run deconvolution software on.



(a) $Beta(\alpha = 5, \beta = 1)$



(b) $Beta(\alpha = 1, \beta = 1) = Unif(0, 1)$



(c) $Beta(\alpha = 1, \beta = 5)$

Figure 2: Phylogenetic trees for varying probability of large population size for mutated offspring. Percent of mutant offspring $p \sim Beta(\alpha, \beta)$. Increased $\frac{\alpha}{\beta}$ leads to larger mutant population and therefore increased chance a mutant cell type will mutate again. Values of nodes are L1 distances between their copy number profile and normal (root). Yellow color indicates earlier and red indicates later mutations. Single chromosome of length 10,000 with mean mutation size 1,000.

We next noted that increasing the mean mutation size from $\ell = 1000$ to $\ell = 5000$ increased the average L1 distance for nodes (Figure 3). This is most due to the increased likelihood in overlapping mutation segments which can cause multiplicative increases or decreases in copy number. Note the increased L1 distance is not due to an increased number of segments as the total number of segments $k = 2 \cdot m + d$ where m is the number of mutations and d is the number of chromosomes. Interestingly, some children decreased their L1 distance relative to their parents (lower left in Figure 3) showing that L1 distance between nodes is not a perfect measure how mutated a cell population is but still works a majority of the time. We additionally showed changing the total number of mutations can drastically decrease or increase the complexity of the phylogeny (Figure 4).

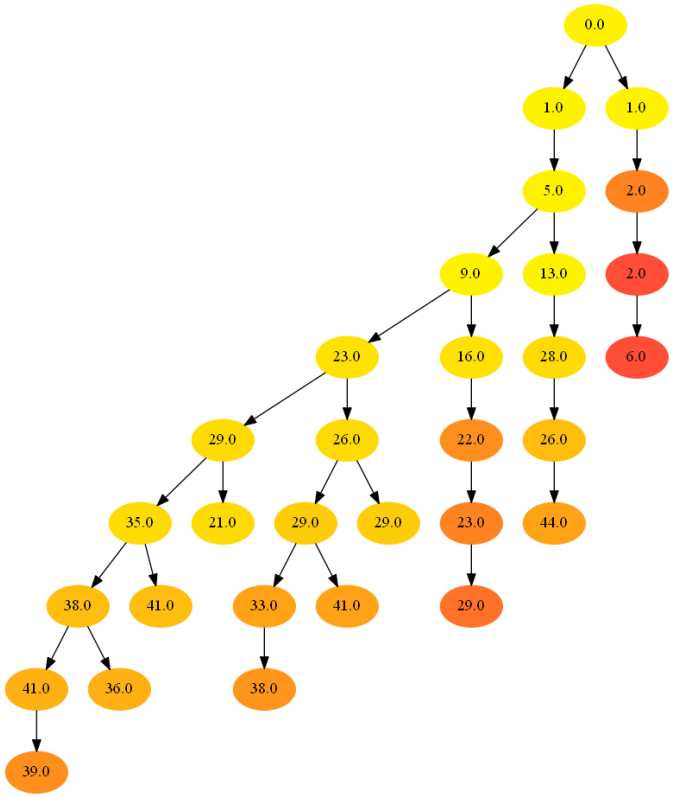
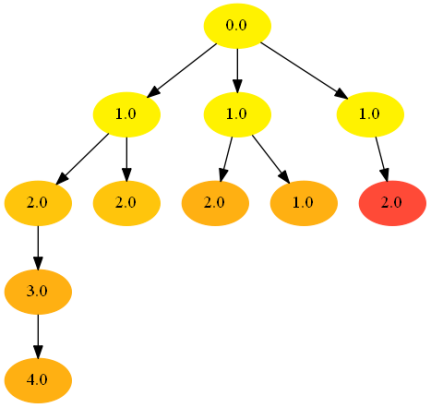
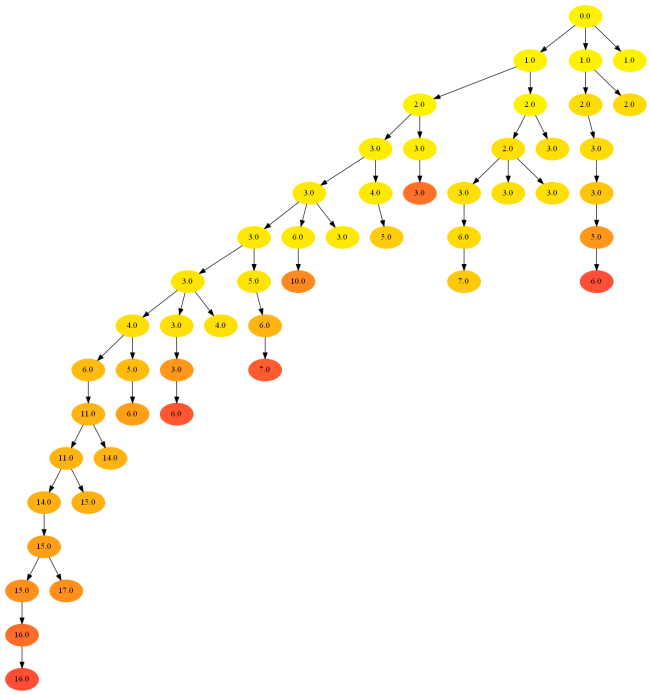


Figure 3: Increasing mean mutation length increases the L1 distance for more mutated nodes. Mean mutation length $\ell = 5000$ and percent of mutant offspring $p \sim Beta(\alpha = 5, \beta = 1)$. Values of nodes are L1 distances between their copy number profile and normal (root). Yellow color indicates earlier and red indicates later mutations. Single chromosome of length 10,000.



(a) Number of mutations $m = 10$



(b) Number of mutations $m = 50$

Figure 4: Phylogenetic topology for varying number of mutations. Values of nodes are L1 distances between their copy number profile and normal (root). Yellow color indicates earlier and red indicates later mutations. Single chromosome of length 10,000 with mean mutation size 1,000.

4 Discussion and Conclusion

We have developed a biologically relevant method for simulating tumor phylogenies at a single mutation resolution using a continuous time Markov model. The various input parameters make this a powerful tool for any team or lab developing a deconvolution tool who desire a standardized data set for testing their tool. We showed that many of these parameters can be altered to create entirely different trees and can be tailored to fit the users' needs. Alternatively, we used predefined input parameters to create a data set for anyone developing a deconvolution tool to use.

Unfortunately, we were unable to successfully test our data set on any deconvolution tools. Namely we attempted to run Simone et al.'s Copy-Number Tree Mixture Deconvolution program [10]. Although we were able to successfully obtain a CPLEX license, the current version of their code still contained a few bugs and we were unable to even run example data through their deconvolution tool. In the future, we will have to run a few of these tools and make comparisons between inferred and actual trees before claiming our data set as a standard.

One limitation of our approach to using a continuous time Markov model lies in that we use continuous time to model when mutations occur but not on population growth. When a mutation occurs, we instantaneously increase the population size of the cells carrying that mutation to a randomly chosen value. To improve on this, the percent of cells containing the mutation relative to those that do not should increase over time as they would in a real tumor. Future work could contain a novel model for determining the population size as a function of time.

Additional future work involves increasing the types of allowed mutations as well as iteratively revising our model based on results from running deconvolution softwares. Many of the deconvolution tools out there require single nucleotide polymorphism (SNP) data and the accompanying variant and total read counts. While we are able to trivially estimate these for entire segments by sampling from a binomial distribution, we have yet to include SNP mutations into our model and can therefore not report SNP specific read counts. However including SNPs into the model in the future should not be a difficult task. In time, we hope these changes will allow our tumor phylogeny simulator to become a useful resource for the community.

References

[1] Damian Yap Adrian Wan Emma Laks Justina Biele Gavin Ha Samuel Aparicio Alexandre Bouchard-Côté Sohrab P Shah Andrew Roth Jaswinder Khattra. "PyClone: Statistical inference of clonal population structure in cancer". In: *nature* (2014). URL: http://www.nature.com/nmeth/journal/v11/n4/full/nmeth.2883.html.

[2] Yung CK Jang GH Stein L Morris Q. Deshwar AG Vembu S. "PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors." In: *PubMed* (2014). URL: https://www.ncbi.nlm.nih.gov/pubmed/25786235.

[3] Oesper L Raphael BJ. El-Kebir M Satas G. "Inferring the Mutational History of a Tumor Using Multi-state Perfect Phylogeny Mixtures". In: *PubMed* (2016). SPRUCE. URL: https://www.ncbi.nlm.nih.gov/pubmed/27467246.

[4] Raphael BJ. Hajirasouliha I Mahmoody A. "A combinatorial approach for analyzing intra-tumor heterogeneity from high-throughput sequencing data." In: *PubMed* (2014). URL: https://www.ncbi.nlm.nih.gov/pubmed/24932008.

[5] Minn AJ Zhang NR. Jiang Y Qiu Y. "Assessing intratumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing." In: *PubMed* (2016). Canopy. URL: https://www.ncbi.nlm.nih.gov/pubmed/27573852.

[6] Raphael BJ Oesper L Satas G. "Quantifying tumor heterogeneity in whole-genome and whole-exome sequencing data." In: *PubMed* (2014). THetA 2. URL: https://www.ncbi.nlm.nih.gov/pubmed/25297070.

[7] Victoria Popic. "POMEGRANATE: Cancer Lineage Tree Simulator". In: (). URL: https://github.com/viq854/pomegranate.

[8] S. J. Spielman and C. O. Wilke. "Pyvolve: A Flexible Python Module for Simulating Sequences along Phylogenies". In: *PLOS ONE* 10 (2015), e0139047. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0139047.

[9] Russell Schwartz Theodore Roman Lu Xie. "Medoidshift clustering applied to genomic bulk tumor data". In: *PMC* (2016). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4895288/.

[10] Simone Zaccaria et al. "The Copy-Number Tree Mixture Deconvolution Problem and Applications to Multi-sample Bulk Sequencing Tumor Data". In: *Research in Computational Molecular Biology: 21st Annual International Conference, RECOMB 2017, Hong Kong, China, May 3-7, 2017, Proceedings*. Ed. by S. Cenk Sahinalp. Cham: Springer International Publishing, 2017, pp. 318–335. ISBN: 978-3-319-56970-3. DOI: 10.1007/978-3-319-56970-3_20. URL: https://doi.org/10.1007/978-3-319-56970-3_20.