Journal of Data Science 19 (3), 465–484
July 2021

DOI: 10.6339/21-JDS1016
Data Science in Action

Mutstats: An Ultra-fast Computational Method to Determine Clonal Status of Somatic Mutations

Dehua Bi[1], Subhajit Sengupta[2], Tianjian Zhou[3], and Yuan Ji[1,*]

[1]*Public Health Sciences, University of Chicago, Chicago, IL, USA*
[2]*Cytel Inc, Cambridge, MA, USA*
[3]*Department of Statistics, Colorado State University, Fort Collins, CO, USA*

Abstract

Tumor cell population is a mixture of heterogeneous cell subpopulations, known as subclones. Identification of clonal status of mutations, i.e., whether a mutation occurs in all tumor cells or in a subset of tumor cells, is crucial for understanding tumor progression and developing personalized treatment strategies. We make three major contributions in this paper: (1) we summarize terminologies in the literature based on a unified mathematical representation of subclones; (2) we develop a simulation algorithm to generate hypothetical sequencing data that are akin to real data; and (3) we present an ultra-fast computational method, Mutstats, to infer clonal status of somatic mutations from sequencing data of tumors. The inference is based on a Gaussian mixture model for mutation multiplicities. To validate Mutstats, we evaluate its performance on simulated datasets as well as two breast carcinoma samples from The Cancer Genome Atlas project.

Keywords *cancer genomics; next-generation sequencing; subclone; tumor heterogeneity*

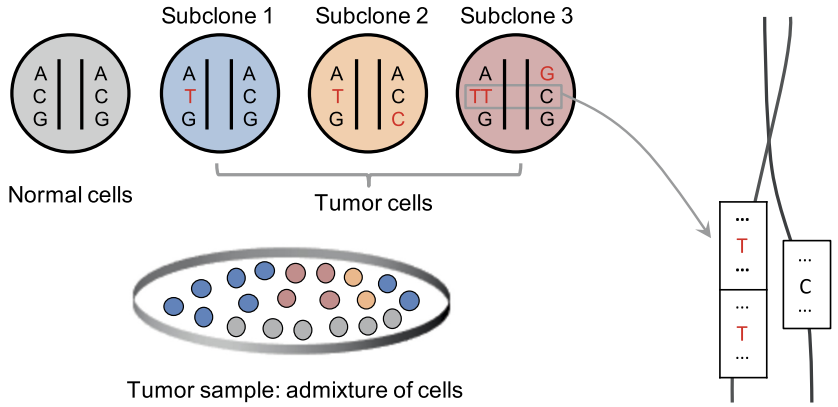
1 Introduction

Tumor cell population is known to be a mixture of heterogeneous cell subpopulations (Nowell, 1976; Marusyk and Polyak, 2010; Swanton, 2012; Yates and Campbell, 2012). Each subpopulation is characterized by its distinct genome and is referred to as a *subclone*. With the advancement of next generation sequencing (NGS) technology, whole-genome or whole-exome sequencing data have enabled researchers to study the genomic profile of tumor subclones in detail within the same patient or among different patients. In particular, data of sequence variations such as single nucleotide variants (SNVs) and structural variations such as copy number aberrations (CNAs) are widely used for subclone inference.

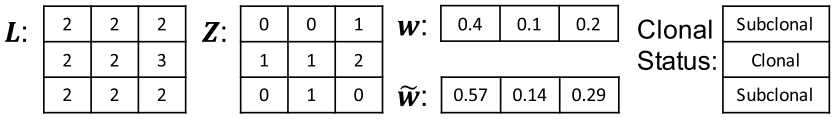
For a heterogeneous tumor sample with multiple subclones, understanding of its mutational profile, such as the *clonal status* of the mutations, is crucial for accurate disease prognosis and precision medicine. A mutation is called *clonal* if it occurs across all the tumor cells. On the other hand, a mutation is called *subclonal* if it only occurs in a subset of tumor cells. Figure 1(a) provides a stylized illustration of tumor heterogeneity and clonal and subclonal mutations. The "G" mutation at the first locus and the "C" mutation at the third locus are only possessed by a fraction of tumor cells and are subclonal, while the "T" mutation at the second locus is possessed by all tumor cells thus is clonal. Inference on clonal status can shed light on the timing of mutational processes and thus provides important information for personalized treatment

*Corresponding author. Email: yji@health.bsd.uchicago.edu.

Received January 23, 2021; Accepted May 16, 2021



(a) Illustration of tumor heterogeneity



(b) Representation of tumor subclones

Figure 1: (a) Illustration of tumor heterogeneity, clonal and subclonal mutations, and copy number aberrations. In this example, we have 3 genomic loci that we record mutations, and a total of 3 tumor subclones. (b) Representation of tumor subclones using $\boldsymbol{L}$, $\boldsymbol{Z}$ and $\boldsymbol{w}$.

strategy. For example, for patients with chronic lymphocytic leukemia, the presence of subclonal driver mutations leads to more rapid disease progression (Landau et al., 2013). The acquisition of a drug resistance allele in a fraction of tumor cells can result in clinical drug resistance and thus can reduce the efficacy of cancer therapies (Schmitt et al., 2016). For patients with colorectal cancers, the acquisition of subclonal KRAS mutation leads to resistance to cetuximab (Misale et al., 2012). Intuitively, clonal mutations generally occur earlier in time compared to subclonal mutations. A greater burden of subclonal mutations indicates a greater degree of intra-tumor heterogeneity which could be associated with worse outcome for patients with various cancer types.

There is an extensive literature on tumor heterogeneity and subclone inference. See, for example, Beerenwinkel et al. (2014) for a review. Different methods study tumor heterogeneity from different perspectives. For example, McGranahan et al. (2015) focus on deciphering the clonal status of mutations. Roth et al. (2014) propose the PyClone method, which represents subclones as clusters of SNVs and infers SNV clusters and their cellular prevalence. Deshwar et al. (2015) further impose phylogenetic constraint on SNV clusters and infer subclone phylogeny. Lastly, Sengupta et al. (2015), Lee et al. (2016), Zhou et al. (2019) and Zhou et al. (2020) directly model subclonal genomes which are characterized by overlapping sets of SNVs.

Our work complements existing methods in several aspects. First, a number of terminologies are presented in the existing methods, defined under different characterizations of tumor heterogeneity. Some terminologies have been defined conceptually rather than quantitatively. This could potentially lead to confusions and makes it hard linking one term with another. We discuss a unified mathematical representation of subclones and define the related terminologies under the unified representation in a rigorous way. Second, many existing methods, such as

Sengupta et al. (2015), are based on direct inference about subclonal genomes. Direct inference is promising, because it fully characterizes the genetic structure of the subclones and implies all other quantities of interest. However, such inference is very challenging and has identifiability issue. Moreover, direct inference is computationally expensive and finds it hard to analyze thousands of mutations. Therefore, in this work, we choose to focus on identifying the clonal status of somatic mutations to achieve computational feasibility. Lastly, we recognize that validation is crucial to the development of inferential algorithms. We develop a simulation algorithm to generate hypothetical sequencing data that are akin to real data. We conduct extensive simulation studies to assess the performance and speed of our proposed method. Simulated datasets with clearly labeled clonal status could greatly facilitate the development of novel inferential algorithms, both for sanity check and for comparison to alternative methods in the field.

The remainder of the paper is organized as follows. In Section 2, we give a quantitative representation of subclones and define commonly used terminologies under the unified framework. In Section 3, we present the Mutstats method of designating clonal status of mutations. In Section 4, we propose a general algorithm to simulate read counts data and evaluate the performance of Mutstats through simulation studies. In Section 5, we analyse two samples of a breast cancer patient from The Cancer Genome Atlas (TCGA) dataset. Section 6 concludes the paper with a discussion.

2 Representation of Subclones

We follow the notations from the direct inference literature (e.g. Lee et al. 2016) to give a quantitative representation of subclones. Consider one tumor tissue sample that is dissected from either primary or metastatic sites. Typically, tumor samples contain normal cells to a certain extent. The fraction of tumor cells in the entire cell population is called tumor *purity* and is denoted by μ. For example, in Figure 1(a), the hypothetical tumor sample contains 30% normal cells (6 grey cells) and 70% tumor cells (14 cells of color blue, pink, and orange), leading to a tumor purity of $\mu = 0.7$. This mixture could be thought of as the first level of tumor heterogeneity. By definition, the normal cells do not possess any somatic mutation and have copy number 2 at any genomic locus as we restrict our attention to autosome.

Let C denote the number of subclones, which also needs to be estimated in practice. For example, in Figure 1(a), we have $C = 3$ subclones. We first construct a $S \times C$ dimensional integer-valued matrix $\boldsymbol{L}$ to characterize subclonal copy numbers, where S is the number of genomic loci that we record SNVs. The (s, c)-entry of $\boldsymbol{L}$, l_{sc}, represents the total copy number of subclone c at locus s. An example of the $\boldsymbol{L}$ matrix is shown in Figure 1(b). Since CNAs occur in segments of genomic regions, we assume the S loci fall into K copy number segments (CNSs), with $k(s)$ index the CNS of locus s. The loci in the same CNS have the same copy number status. Let l^*_{kc} denote the total copy number of CNS k. For two loci s_1 and s_2 in the same CNS k, i.e. $k(s_1) = k(s_2) = k$, we have $l_{s_1c} = l_{s_2c} = l^*_{kc}$.

Next, we introduce a $S \times C$ dimensional integer-valued matrix $\boldsymbol{Z}$ to record subclonal SNVs. The (s, c)-entry of $\boldsymbol{Z}$, z_{sc}, represents the number of variant alleles at locus s of subclone c. We always have $z_{sc} \leqslant l_{sc}$. Again, an example of the $\boldsymbol{Z}$ matrix is shown in Figure 1(b).

Finally, we use a C dimensional vector $\boldsymbol{w}$ to represent the population frequencies of the subclones. The c-th element of $\boldsymbol{w}$, w_c, represents the population frequency of subclone c. We have $\sum_{c=1}^{C} w_c = \mu$. Denote by $\tilde{w}_c = w_c/\mu$ the proportion of subclone c in all tumor cells; we have $\sum_{c=1}^{C} \tilde{w}_c = 1$. Examples of the $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ vectors are provided in Figure 1(b).

Table 1: Terminologies in the subclone inference literature, explanations and derivations from the basic quantities μ, $\boldsymbol{L}$, $\boldsymbol{Z}$ and $\boldsymbol{w}$.

Terminology	Explanation	Derivation
Purity	Fraction of cancer cells	μ
Mutation multiplicity	Locus-specific average allele copies in cancer cells carrying a mutation	$\sum_{c=1}^{C} \tilde{w}_c z_{sc}$
Variant allele fraction	Locus-specific fraction of allele copies carrying a mutation	$\frac{\sum_{c=1}^{C} w_c z_{sc}}{\mu \cdot \sum_{c=1}^{C} \tilde{w}_c l_{sc} + (1-\mu)\cdot 2}$
Cancer cell fraction	Locus-specific fraction of cancer cells carrying a mutation	$\sum_{c=1}^{C} \tilde{w}_c \mathbf{1}(z_{sc} > 0)$
Cellular prevalence	Locus-specific fraction of cells carrying a mutation	$\sum_{c=1}^{C} w_c \mathbf{1}(z_{sc} > 0)$
Ploidy	Average copy number of the entire tumor genome	$\left(\sum_{s=1}^{S} \sum_{c=1}^{C} \tilde{w}_c l_{sc}\right) / S$

The quantities $\boldsymbol{L}$, $\boldsymbol{Z}$, $\boldsymbol{w}$ and μ, which are illustrated in Figure 1, provide a unified representation to fully describe the genetic structure of the subclones, and all other quantities of interest can be derived from these four. For example, SNV s is clonal if $z_{sc} \geqslant 1$ for all c, and SNV s is subclonal if $z_{sc} = 0$ for some c. In the literature, some other terminologies are used to characterize tumor heterogeneity. Nevertheless, these terminologies can be represented by μ, $\boldsymbol{L}$, $\boldsymbol{Z}$ and $\boldsymbol{w}$. Table 1 summarizes some commonly used terminologies, and we will refer to some in later discussions.

3 The Mutstats Method

3.1 Data Preparation

Denote the total number of reads and the number of reads with variant sequences as N_s and n_s for SNV locus s, respectively, the read counts are the only data that are directly observed. The Mutstats method starts with tumor and matched normal `bam` (Barnett et al., 2011) files which are generated by mapping the raw short reads from `fastq` (Cock et al., 2010) files to the appropriate reference genome. Using tumor and normal `bam` files, somatic mutation calling tools find out all the loci on chromosomes that bear SNVs. Commonly used tools include, for example, `Varscan2` (Koboldt et al., 2012), `MuTect` (Cibulskis et al., 2013) or `Muse` (Fan et al., 2016).

Following existing subclone reconstruction methods, we obtain estimates of tumor purity (μ) and copy numbers in copy-number segments (l_{kc}) using existing purity caller and copy number caller, such as `Battenberg` (Nik-Zainal et al., 2012), `ABSOLUTE` (Carter et al., 2012) or `FACETS` (Shen and Seshan, 2016). In this work, we use the `Battenberg` caller to obtain estimates of tumor purity and copy numbers. In the following discussion, we will focus on determining the clonal status of the SNVs.

Next, we estimate mutation multiplicity for each SNV. *Mutation multiplicity* is defined as the average allele copies in cancer cells carrying a mutation. Denote by m_s the multiplicity of SNV s. Using the notation in Section 2, $m_s = \sum_{c=1}^{C} \tilde{w}_c z_{sc}$. For example, in Figure 1(a),

$m_s = 0.29, 1.29$ and 0.14 for $s = 1, 2$ and 3, respectively. A point estimate of m_s based on the read counts is computed by

$$\hat{m}_s = \frac{\bar{l}_s}{\mu} \cdot \frac{n_s}{N_s}, \tag{1}$$

where $\bar{l}_s$ is the average copy number of SNV s, $\bar{l}_s = \mu \cdot \sum_{c=1}^{C} \tilde{w}_c l_{sc} + (1 - \mu) \cdot 2$. Recall that μ and $\bar{l}_s$ are estimated by upstream bioinformatics tools. The point estimate $\hat{m}_s$ is unbiased with the understanding that $\mathrm{E}(n_s/N_s) = \sum_{c=1}^{C} w_c z_{sc}/\bar{l}_s$, where the right hand side of the equation is referred to as the variant allele fraction (VAF). To distinguish between the mutation multiplicity and its point estimate, we refer to m_s and $\hat{m}_s$ as expected mutation multiplicity and observed mutation multiplicity, respectively.

We note that, instead of mutation multiplicity, many methods (e.g. Roth et al. 2014) use cancer cell fraction (CCF) to infer clonal status. CCF is defined as the fraction of cancer cells carrying a mutation, $\sum_{c=1}^{C} \tilde{w}_c \mathbf{1}(z_{sc} > 0)$. For example, in Figure 1(a), the CCF values for mutations 1, 2 and 3 are 0.29, 1 and 0.14, respectively. By definition, CCF should be $\in [0, 1]$, i.e., not exceed 1. However, due to mapping or sequencing errors, estimates of CCF could become greater than one, which is difficult to interpret biologically since fractions cannot be greater than 1. To remedy this issue, in some methods, in practice CCFs are often artificially truncated at 1. Apparently, these artificial truncations are ad-hoc and not desirable. In contrast, multiplicity does not have a theoretical upper bound and can take any non-negative real values. Therefore, we choose to use mutation multiplicity rather than CCF to infer clonal status.

3.2 Model for Mutation Multiplicities

We model the observed mutation multiplicity for SNV s with a Gaussian distribution centered at the expected mutation multiplicity,

$$\hat{m}_s \mid m_s, \tau_s \sim N(m_s, \tau_s^2).$$

The SNVs co-occurred in the same set of subclones (i.e. the SNVs having the same z_{sc}'s) should have the same expected mutation multiplicities. Therefore, we assume (m_s, τ_s) can only take R possible values, with

$$\Pr\big((m_s, \tau_s) = (u_r, \sigma_r) \mid u_r, \sigma_r, \pi_r\big) = \pi_r, \quad \text{for } r = 1, 2, \ldots, R,$$

where R is unknown a priori and needs to be estimated. Denote by $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_R, u_1, \ldots, u_R, \sigma_1, \ldots, \sigma_R)$. Integrating out (m_s, τ_s), the marginal distribution of $\hat{m}_s$ is a finite mixture of Gaussian distributions (Melnykov et al., 2010),

$$p(\hat{m}_s \mid \boldsymbol{\theta}) = \sum_{r=1}^{R} \pi_r \phi(\hat{m}_s \mid u_r, \sigma_r), \tag{2}$$

where π_r represents the weight of mixture component r, and $\phi(\cdot \mid u_r, \sigma_r)$ denotes a Gaussian density with mean u_r and standard deviation σ_r.

We use the expectation-maximization (EM) algorithm (Dempster et al., 1977) to obtain the maximum likelihood estimate (MLE) of the parameters $\boldsymbol{\theta}$. Let ξ_{sr} be a cluster membership indicator such that $\xi_{sr} = 1$ (or 0) represents observation s belongs to (or does not belong to)

cluster r, respectively. Denote by e_{sr} the conditional probability that observation s belongs to cluster r given the parameters,

$$e_{sr} = \Pr(\xi_{sr} = 1 \mid \hat{m}_s, \boldsymbol{\theta}) = \frac{\pi_r \phi(\hat{m}_s \mid u_r, \sigma_r)}{\sum_{r'=1}^{R} \pi_{r'} \phi(\hat{m}_s \mid u_{r'}, \sigma_{r'})}.$$

For a fixed R, the EM algorithm starts from random initial values of $\boldsymbol{\theta}$, and then iterates between an E-step and an M-step until convergence. In the E-step, we compute e_{sr} for all $s = 1, \ldots, S$ and $r = 1, \ldots, R$ given the current values of the parameters $\boldsymbol{\theta}$. In the M-step, we compute the MLE of $\boldsymbol{\theta}$ given e_{sr}'s. It can be shown that the parameters converge to the MLE of the Gaussian mixture model (Equation 2).

To select an optimal number of mixture components R, we run the EM algorithm with different $R = 1, \ldots, 7$. We then choose the optimal R based on the Bayesian information criterion (BIC, Schwarz et al. 1978). The R package `mclust` (Fraley et al., 2016; Scrucca et al., 2016) is used for implementation.

3.3 Determining Clonal Status

After parameter estimates of the Gaussian mixture model are obtained, we use the following procedure to determine the clonal status of an SNV.

Recall that $m_s = \sum_{c=1}^{C} \tilde{w}_c z_{sc}$. If $m_s < 1$, at least one $z_{sc} = 0$ for $c = 1, \ldots, C$, which suggests SNV s is subclonal. On the other hand, $m_s \geqslant 1$ does not necessarily mean SNV s is clonal, but can be due to, for example, high copy number. Therefore, additional copy number information is needed to determine the clonal status of SNV s when $m_s \geqslant 1$. Let

$$\lambda_s = \lambda_s(\hat{m}_s, \boldsymbol{\theta}) = \Pr(\hat{m}_s^{\text{rep}} < 1 \mid \hat{m}_s, \boldsymbol{\theta}), \tag{3}$$

where $\hat{m}_s^{\text{rep}}$ is a hypothetical replication of the observed mutation multiplicity for SNV s. A larger λ_s means a larger probability of $\hat{m}_s^{\text{rep}} < 1$ thus indicates a higher chance of $m_s < 1$, and vice versa. Therefore, λ_s can be used as a proxy to determine whether $m_s < 1$. The advantage of using λ_s is that it takes into account the uncertainty associated with the point estimates.

The probability $\lambda_s = \Pr(\hat{m}_s^{\text{rep}} < 1 \mid \hat{m}_s, \boldsymbol{\theta})$ is calculated as follows,

$$\begin{aligned} \Pr(\hat{m}_s^{\text{rep}} < 1 \mid \hat{m}_s, \boldsymbol{\theta}) &= \sum_{r=1}^{R} \Pr(\xi_{sr} = 1 \mid \hat{m}_s, \boldsymbol{\theta}) \Pr(\hat{m}_s^{\text{rep}} < 1 \mid \xi_{sr} = 1, \hat{m}_s, \boldsymbol{\theta}) \\ &= \sum_{r=1}^{R} e_{sr} \Phi(1 \mid u_r, \sigma_r), \end{aligned} \tag{4}$$

where $\Phi(\cdot \mid u_r, \sigma_r)$ denotes the cumulative distribution function of a Gaussian distribution with mean u_r and standard deviation σ_r.

If λ_s is large, say, $\lambda_s > H_1$ for a certain threshold H_1, we think $m_s < 1$ thus determine SNV s is subclonal. Otherwise, if $\lambda_s \leqslant H_1$, we think $m_s \geqslant 1$ and use additional copy number information to determine the clonal status of SNV s. Suppose SNV s resides inside CNS k. Using the notation introduced in Section 2, let l_{kc1}^* and l_{kc2}^* denote the major and minor allele-specific copy numbers (ASCNs) of CNS k in subclone c. Here, the major and minor alleles represent the most common and less common alleles in the population, and the ASCN of an allele refers to the number of copies of that allele. Note that the reference/alternative allele is different from

the major/minor allele: the former is defined with respect to an individual patient, while the latter is based on the population. The following relationship is assumed: $z_{sc} \leqslant \max(l^*_{kc1}, l^*_{kc2})$, meaning that the copy number of the alternative allele should be less than or equal to the maximum ASCN. The `Battenberg` caller provides aggregated information about ASCNs, and we utilize such information to determine the clonal status of SNV s. The following two scenarios are possible.

1. If `Battenberg` determines that CNS k has one copy number state, it outputs a pair of ASCNs (q_{k1}, q_{k2}). Essentially, `Battenberg` thinks that $l^*_{kc1} = q_{k1}$ and $l^*_{kc2} = q_{k2}$ for all c. Denote by $\tilde{q}_k = \max(q_{k1}, q_{k2})$. If $m_s > \tilde{q}_k$, we identify s as clonal, since a subclonal SNV cannot produce a mutation multiplicity greater than $\sum w_c \max(l^*_{kc1}, l^*_{kc2}) = \tilde{q}_k$. To determine whether $m_s > \tilde{q}_k$, we calculate

$$\Pr(\hat{m}_s^{\text{rep}} > \tilde{q}_k \mid \hat{m}_s, \boldsymbol{\theta}) = \sum_{r=1}^{R} e_{sr} \left[1 - \Phi(\tilde{q}_k \mid u_r, \sigma_r)\right], \tag{5}$$

   and if this is greater than some threshold H_2, we think $m_s > \tilde{q}_k$ and classify s as clonal. Otherwise, we let s be unclassified at this time.
2. If `Battenberg` determines that CNS k has two copy number states, it outputs two pairs of ASCNs, (q^1_{k1}, q^1_{k2}) and (q^2_{k1}, q^2_{k2}), as well as their population frequencies in the tumor cells, ρ_1 and ρ_2. Implicit in the two sets of ASCNs is a partition of the C subclones, $\{1, \ldots, C\} = \mathcal{C}_1 \cup \mathcal{C}_2$. For all $c \in \mathcal{C}_1$, $l^*_{kc1} = q^1_{k1}$, $l^*_{kc2} = q^1_{k2}$ and $\sum_{c\in\mathcal{C}_1} \tilde{w}_c = \rho_1$. On the other hand, for all $c \in \mathcal{C}_2$, $l^*_{kc1} = q^2_{k1}$, $l^*_{kc2} = q^2_{k2}$ and $\sum_{c\in\mathcal{C}_2} \tilde{w}_c = \rho_2$. Denote by $\tilde{q}^1_k = \max(q^1_{k1}, q^1_{k2})$ and $\tilde{q}^2_k = \max(q^2_{k1}, q^2_{k2})$. If $m_s > \rho_1\tilde{q}^1_k + \rho_2\tilde{q}^2_k$, s should be clonal. Again, this is because a subclonal SNV cannot produce a mutation multiplicity greater than $\sum w_c \max(l^*_{kc1}, l^*_{kc2}) = \rho_1\tilde{q}^1_k + \rho_2\tilde{q}^2_k$. Calculate

$$\Pr(\hat{m}_s^{\text{rep}} > \rho_1\tilde{q}^1_k + \rho_2\tilde{q}^2_k \mid \hat{m}_s, \boldsymbol{\theta}) = \sum_{r=1}^{R} e_{sr} \left[1 - \Phi(\rho_1\tilde{q}^1_k + \rho_2\tilde{q}^2_k \mid u_r, \sigma_r)\right], \tag{6}$$

   and if this is greater than H_2, we think $m_s > \rho_1\tilde{q}^1_k + \rho_2\tilde{q}^2_k$ and classify s as clonal. Otherwise, we temporarily assign unclassified status to s. For simplicity, we use the same threshold H_2 for both scenarios, but it is possible to consider different thresholds.

The values of H_1 and H_2 can be specified through simulation. Specifically, in the simulation studies to be presented in Section 4, we first simulate 10 tumor samples, for which the true clonal status of the mutations are known. Next, we run Mutstats on the simulated data with a grid of values of (H_1, H_2). Lastly, we find the optimal threshold values that lead to the highest classification accuracy on the 10 simulated samples. The value of H_1 can also be chosen based on a desirable maximum false discovery rate (MFDR) threshold, denoted by f_0. Let $\lambda_{(k)}$ denote the k-th largest value of $\{\lambda_1, \lambda_2, \ldots, \lambda_S\}$, and let

$$s^* = \max\left\{s : \frac{\sum_{k<s}(1 - \lambda_{(k)})}{|k : k < s|} < f_0\right\}, \tag{7}$$

where $|A|$ denotes the cardinality of the set A. We may set $H_1 = \lambda_{s^*}$. This error rate is denoted as MFDR because $1 - \lambda_s = Pr(\hat{m}_s^{rep} \geqslant 1 | \hat{m}_s^{rep}, \theta)$ does not necessarily imply that the SNV is clonal, due to the existance of unknown status. Therefore, it is an overestimation of the true false discovery rate. In the real data analysis (Section 5), the threshold H_1 is selected using this approach (while H_2 is still chosen based on simulation).

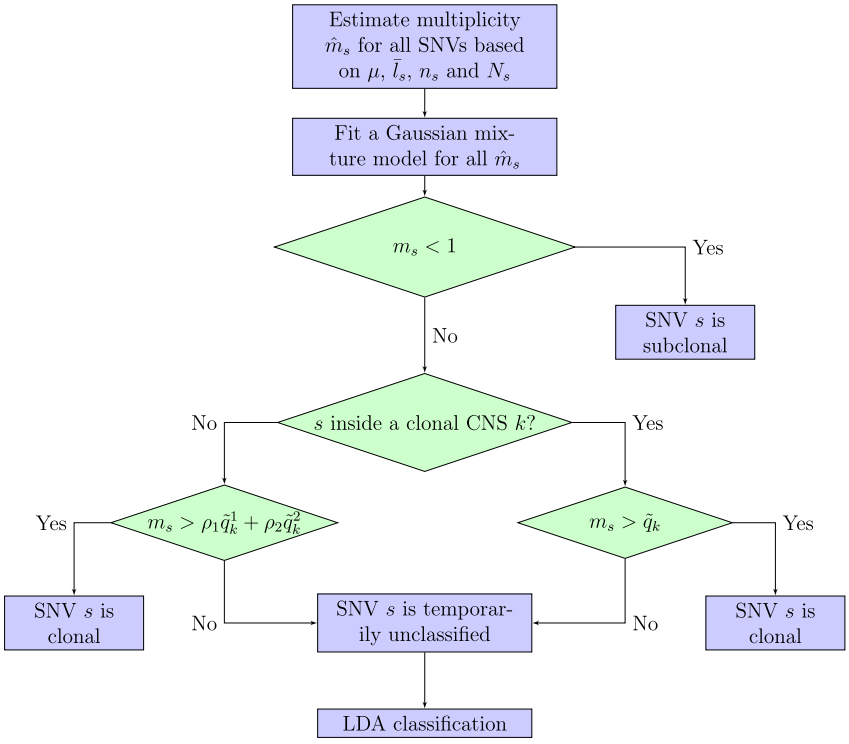


Figure 2: Diagram for determining clonal status.

The procedure described above leaves some SNVs unclassified. We further use linear discriminant analysis (LDA) to find their clonal status, treating the already-classified SNVs as training data. For SNV s, let $\boldsymbol{x}_s = (N_s, n_s, \hat{m}_s, \lambda_s)$, and let $y_s \in \{0, 1\}$ denote its clonal status (0 for subclonal and 1 for clonal). Recall that N_s and n_s are the total number of reads and number of reads with variant sequences mapped to the location of SNV s, respectively, $\hat{m}_s$ is the observed mutation multiplicity, and λ_s is defined in Equation (3). We use $\boldsymbol{x}_s$ as covariates to predict y_s. The LDA method assumes that the conditional distributions for $[\boldsymbol{x}_s | y_s = 0]$ and $[\boldsymbol{x}_s | y_s = 1]$ are normal with parameters $(\boldsymbol{\mu}_0, \Sigma)$ and $(\boldsymbol{\mu}_1, \Sigma)$, respectively. The predicted clonal status for an unclassified SNV s is

$$i^* = \arg\max_i \phi(\boldsymbol{x}_s \mid \boldsymbol{\mu}_i, \Sigma) \cdot \kappa_i, \quad \text{for } i = 0, 1,$$

where $\kappa_i = \dfrac{\#\text{ of SNVs with clonal status } i}{\text{Total } \#\text{ of SNVs}}$ is the relative frequency of clonal status i, and $\phi(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \Sigma)$ denotes a multivariate Gaussian density with mean $\boldsymbol{\mu}_i$ and covariance matrix Σ. The `R` package `lda` is used to implement the LDA algorithm. With the additional LDA step, the proposed Mutstats algorithm is able to determine the clonal status for all SNVs. The flow of the entire Mutstats algorithm is given in Figure 2.

4 Simulation Studies

4.1 Brief Review of Data Simulation Approaches

Despite the increasing effort in generating and providing access simulation approaches that may generate and mimic real-world NGS data could be extremely helpful for research and method development. Numerous simulators have been developed for simulating DNA sequencing data for various applications. For instance, Huang et al. (2012) developed the ART simulator that generate synthetic NGS reads, Shcherbina (2014) constructed the FASTQSim simulator that provides dual functionality of NGS dataset characterization and metagenomic data generation, Qin et al. (2015) proposed the SCNVSim tool for simulating somatic CNVs and structure variations SVs, and Xia et al. (2017) developed the Pysim-sv that simulates HTS data to evaluate performance of structural variation detection algorithm. And recently, Yu et al. (2020) developed the SimuSCoP tool to emulate complex DNA sequencing data. While these methods simulate the lower level data such as reads data, we proposed a general simulation algorithm to directly generate read counts data, which could be directly used by researchers for developing novel methods that take read counts data as input.

Our proposed simulation scheme is anchored by the theory of tumor cell evaluation based on clonal and subclonal somatic mutations (Figure 1). We 1) generate the proportion of the tumor cells and normal cells, 2) generate the allele specific copy numbers that could be clonal or subclonal, and 3) simulate the number of somatic mutations at each loci given in step 2. These are well-known cancer cell genomics that have been explained in Nik-Zainal et al. (2012) and the more recent ICGC landmark publication (The et al., 2020).

In this work, we assess Mutstats via simulation studies. The detailed data simulation scheme consists of the following steps.

1. Generate tumor purity μ, number of subclones C, number of loci S and number of CNSs K from uniform and discrete uniform distributions,

$$\mu \sim \text{Unif}(\mu_{\min}, \mu_{\max}), \qquad C \sim \text{DU}(C_{\min}, C_{\max}),$$
$$S \sim \text{DU}(S_{\min}, S_{\max}), \qquad K \sim \text{DU}(K_{\min}, K_{\max}).$$

   Generate population frequencies of the subclones $\tilde{\boldsymbol{w}}$ from a Dirichlet distribution,

$$\tilde{\boldsymbol{w}} \sim \text{Dir}(a_w, a_w, \ldots, a_w).$$

   Here, $\mu_{\min}, \mu_{\max}, \ldots, K_{\min}, K_{\max}$ are user specified lower and upper bounds for the simulation parameters, and a_w is a user specified hyperparameter.
2. For each CNS $k \in \{1, 2, \ldots, K\}$, let α_k be an indicator of its clonal status. If $C = 1$, $\alpha_k = 1$; otherwise, generate α_k from a Bernoulli distribution,

$$\alpha_k \sim \text{Ber}(p_\alpha),$$

   where p_α is a user specified hyperparameter. According to the value of α_k, there are two possibilities. (a) If $\alpha_k = 1$, CNS k is clonal. Draw common ASCNs (q_{k1}, q_{k2}) for all subclones from discrete uniform distributions,

$$q_{k1} \sim \text{DU}(0, q_{\max}), \quad q_{k2} \sim \text{DU}(1, q_{\max}),$$

   where $q_{\max}$ is a user specified maximum copy number. Set $(l_{sc1}, l_{sc2}) = (q_{k1}, q_{k2})$ for all loci in CNS k and all subclones, i.e. for all $s \in \{s : k(s) = k\}$ and $c \in \{1, 2, \ldots, C\}$. (b) Otherwise,

if $\alpha_k = 0$, CNS k is subclonal. Draw two sets of ASCNs (q^1_{k1}, q^1_{k2}) and (q^2_{k1}, q^2_{k2}) from discrete uniform distributions,

$$q^1_{k1}, q^2_{k1} \sim \mathrm{DU}(0, q_{\max}), \quad q^1_{k2}, q^2_{k2} \sim \mathrm{DU}(1, q_{\max}).$$

For each subclone $c \in \{1, 2, \ldots, C\}$, set $(l_{sc1}, l_{sc2}) = (q^1_{k1}, q^1_{k2})$ or (q^2_{k1}, q^2_{k2}) with equal probability for all loci in CNS k.

3. For each SNV $s \in \{1, 2, \ldots, S\}$, let β_s be an indicator of its clonal status. If $C = 1$, $\beta_s = 1$; otherwise, generate β_s from a Bernoulli distribution,

$$\beta_s \sim \mathrm{Ber}(p_\beta).$$

According to the value of β_s, there are two possibilities. (a) If $\beta_s = 1$, SNV s is clonal, thus $z_{sc} > 0$ for all c. For each subclone c, generate z_{sc} with

$$p(z_{sc} = \zeta) = v_\zeta \quad (1 \leqslant \zeta \leqslant \tilde{l}_{sc}),$$

where $\tilde{l}_{sc} = \max(l_{sc1}, l_{sc2})$, and v_ζ's are user specified hyperparameters. (b) Otherwise, if $\beta_s = 0$, SNV s is subclonal, thus $z_{sc} = 0$ for some c. Denote by $\mathcal{C}_0$ the index set such that $z_{sc} = 0$ for $c \in \mathcal{C}_0$, and let C_0 be the cardinality of $\mathcal{C}_0$. To construct $\mathcal{C}_0$, first generate $C_0 \sim DU(1, C-1)$, and then uniformly choose C_0 indices from the set $\mathcal{C} = \{1, 2, \ldots, C\}$ as $\mathcal{C}_0$. For all $c \in \mathcal{C}_0$, set $z_{sc} = 0$. For $c \in \mathcal{C} \backslash \mathcal{C}_0$, generate z_{sc} with

$$p(z_{sc} = \zeta) = v_\zeta \quad (1 \leqslant \zeta \leqslant \tilde{l}_{sc}).$$

4. Let ϕ denote the expected total number of reads for each locus assuming the locus has no CNA. Generate ϕ from a Gamma distribution,

$$\phi \sim \mathrm{Ga}(a_\phi, b_\phi).$$

Then, for each SNV s, generate the total number of reads from a Poisson distribution,

$$N_s \sim \mathrm{Poi}(\phi \bar{l}_s / 2),$$

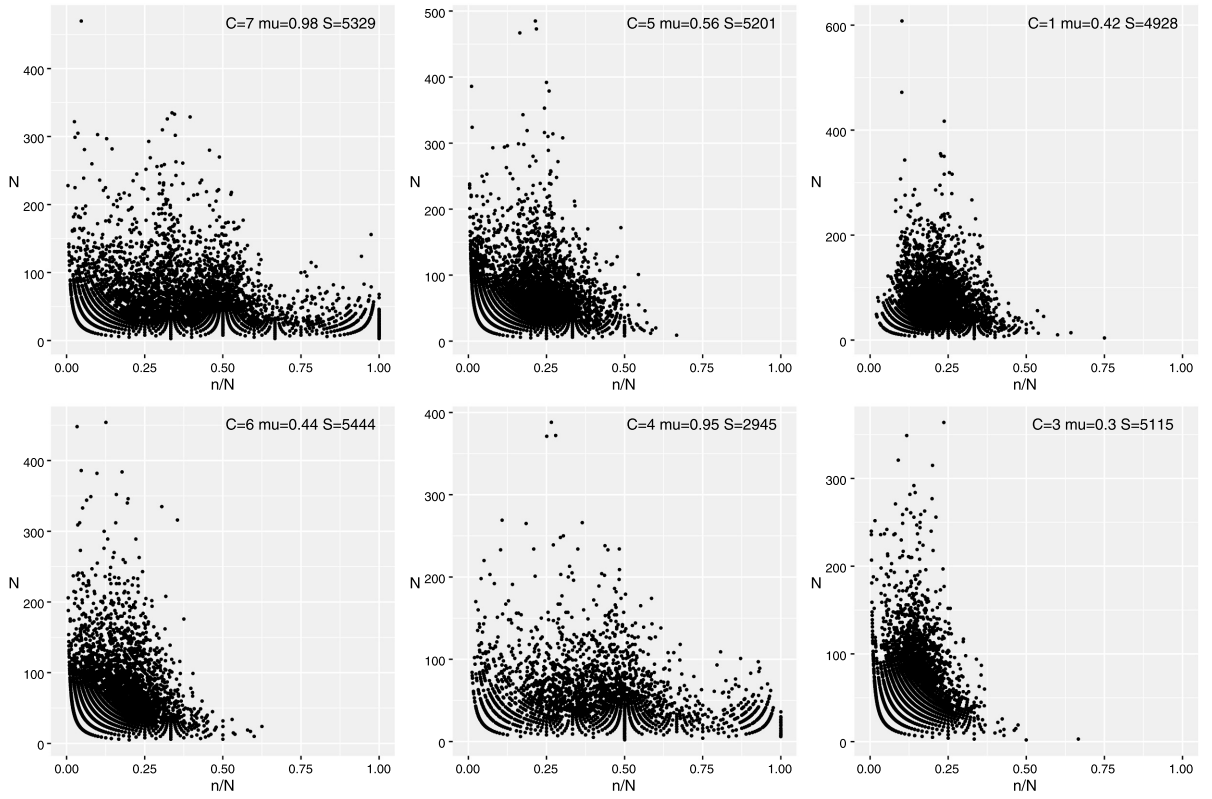
where $\bar{l}_s = \mu \cdot \sum_{c=1}^{C} \tilde{w}_c l_{sc} + (1 - \mu) \cdot 2$ is the average copy number of SNV s. Next, for each SNV s, calculate its VAF by

$$v_s = \frac{1}{\bar{l}_s} \left(\sum_{c=1}^{C} w_c z_{sc} \right).$$

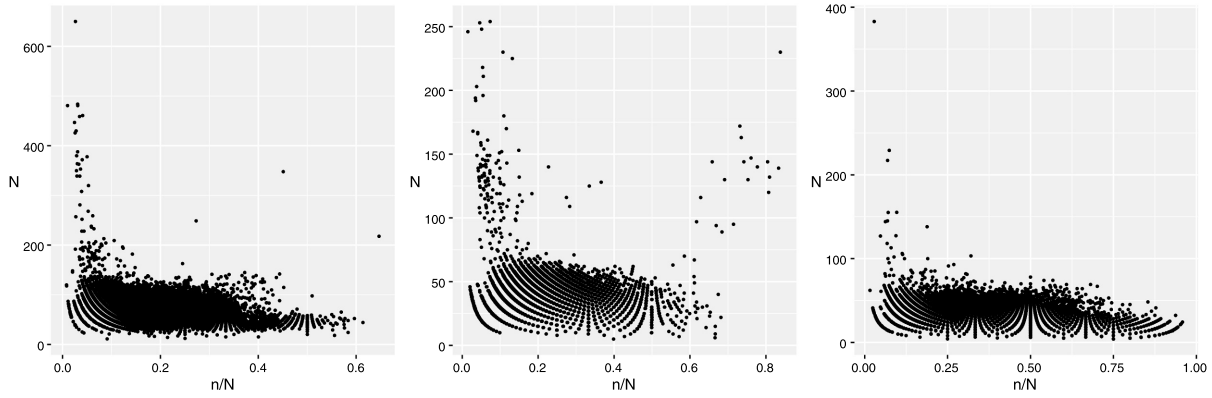
Finally, generate the number of reads with variant sequences from a binomial distribution,

$$n_s \sim \mathrm{Bin}(N_s, v_s).$$

Following the steps above, the simulation algorithm generates tumor purity μ, subclonal copy number matrix $\boldsymbol{L}$, variant allele count matrix $\boldsymbol{Z}$ and subclonal population frequency vector $\tilde{\boldsymbol{w}}$, as well as total number of reads N_s and number of reads with variant sequences n_s for each SNV s. A web-based tool for simulating and visualizing read counts data can be found at https://compgenome.shinyapps.io/tumorsim. Figure 3 shows the plots of N against n/N for six simulated tumors with different purities and numbers of subclones, as well as three samples from real data. It is clear that the simulated data highly resembles that of the real data.



(a) Simulated data



(b) Real data

Figure 3: Plots of N against n/N for (a) six simulated tumors with different purities (mu) and different number of subclones (C), and (b) three real tumor samples.

4.2 Performance on Simulated data

Using the simulator proposed in Section 4.1, we simulate 1000 samples of various choices of purity, number of subclones, number of SNVs and copy number profiles. Furthermore, we excluded the simulated samples with no subclonal SNVs, leaving 868 samples. To select the threshold values H_1 and H_2 (see Section 3.3), we randomly generate 10 additional tumor samples. We define a

Table 2: Accuracy, sensitivity and specificity of Mutstats on the simulated samples.

	Accuracy	Sensitivity	Specificity
Mean	0.7793	0.8671	0.6796
Median	0.8000	0.8736	0.7004
SD	0.0550	0.0876	0.1139

two-dimensional grid of $\{0.1, 0.12, 0.14, \ldots, 0.96, 0.98, 1.0\} \times \{0.1, 0.12, 0.14, \ldots, 0.96, 0.98, 1.0\}$, calculate the classification accuracy for each combination of H_1 and H_2, and then find the optimal threshold values that lead to the highest classification accuracy on the 10 samples. The resulting thresholds are $H_1 = 0.86$ and $H_2 = 0.16$. Based on the selected thresholds, we run the proposed Mutstats algorithm on the 868 samples. The results are reported in Table 2. We use three metrics to evaluate the performance of the proposed method: classification accuracy (or accuracy), sensitivity, and specificity, defined as

$$\text{Accuracy} = \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}(y_s = y_s^{\text{true}}),$$

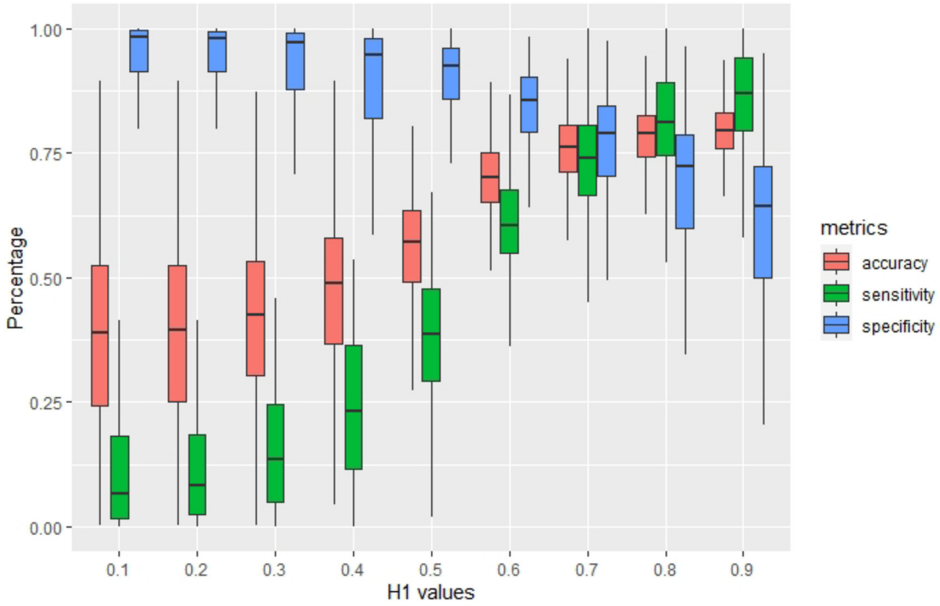
$$\text{Sensitivity} = \frac{\sum_{s=1}^{S} \mathbf{1}(y_s = y_s^{\text{true}}, y_s^{\text{true}} = 1)}{\sum_{s=1}^{S} \mathbf{1}(y_s^{\text{true}} = 1)},$$

$$\text{Specificity} = \frac{\sum_{s=1}^{S} \mathbf{1}(y_s = y_s^{\text{true}}, y_s^{\text{true}} = 0)}{\sum_{s=1}^{S} \mathbf{1}(y_s^{\text{true}} = 0)}.$$

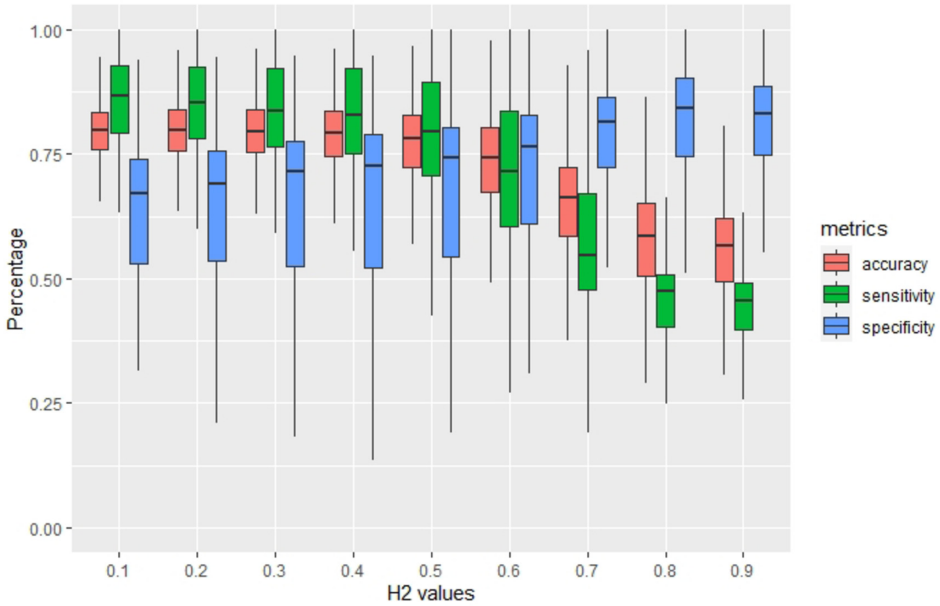
Here, S is the total number of SNVs in a simulated sample, y_s^{true} refers to the true clonal status of SNV s, and y_s is the clonal status of SNV s determined by Mutstats. From Table 2, Mutstats performs well on the simulated data with a high accuracy (around 80%), a high sensitivity (more than 85%), and a good specificity (around 70%).

4.3 Sensitivity Analysis of H_1 and H_2

Since the status of the outcome are determined based on the threshold values H_1 and H_2, they are important components of our proposed algorithm and we are performing some simulation studies with various values of H_1 and H_2 to investigate their effects. To investigate the effect of H_1, we hold H_2 constant at 0.16, and vary H_1 from 0.1 to 0.9. The result is shown in Figure 4a. We observed that while holding H_2 constant, an increase in H_1 results in an increase in both accuracy and sensitivity (fraction of correct subclonal calls among the true subclonal SNVs) but a decrease in specificity (fraction of correct clonal calls among the true clonal SNVs). The reason for increasing sensitivity and accuracy in Figure 4a is due to the fact that higher H_1 values give fewer but more confident subclonal calls, while holding H_2 constant. When H_1 increases to an extreme large value (say 0.99), sensitivity eventually starts to drop due to too few subclonal calls although they are accurate. Specificity is slightly decreased due to more SNVs are not classified as subclonal and sent to the next step for clonal calls. On the other hand, in 4b we fix $H_1 = 0.86$ and increase the value of H_2's. The pattern shows that both sensitivity and accuracy decreases when H_2 is over 0.5. This is because when H_1 is fixed, the number of subclonal SNVs called by our algorithm is fixed and increasing H_2 will result in fewer and more accurate clonal calls. When



(a) Varying H_1 with fixed $H_2 = 0.16$.



(b) Varying H_2 with fixed $H_1 = 0.86$.

Figure 4: Plots of Accuracy, Sensitivity, and Specificity of simulated data for different values of (a) H_1, and (b) H_2.

H_2 is large enough, the reduction in the number of clonal calls eventually causes specificity to drop and also leads to decreasing accuracy.

So to summarize, to achieve a balanced performance, we suggest a reasonably large fraction for H_1 (around 0.9) and a reasonably small fraction for H_2 (around 0.2). In our analysis, we used $H_1 = 0.86$ and $H_1 = 0.16$.

4.4 Comparison with Existing Methods

For comparison, we run two existing tools, PyClone (Roth et al., 2014) and the method proposed by McGranahan et al. (2015) (which we denote by NMCS, from the first author's initials), on a randomly selected set of 100 samples out of the 868 simulated samples.

PyClone does not directly infer the clonal status of mutations. Instead, it clusters SNVs based on their cancer cell fractions (CCFs), and some post-processing steps are necessary to identify the clonal status of SNVs. Recall that the CCF of a SNV is a value between 0 and 1 and represents the fraction of cancer cells carrying it. The SNVs belonging to the same cluster are thought of having the same CCF value. By definition, SNVs with CCF values close to 1 are classified as clonal, while SNVs with CCF values much lower than 1 are classified as subclonal. As PyClone outputs each SNV with its cluster membership as well as the posterior mean of its CCF value, we use the following three different post-processing procedures to determine the clonal/subclonal status for each cluster of SNVs based on the posterior CCF value, using a probability threshold of 0.9.

1. PyClone Median: For an SNV cluster with multiple SNVs, we obtain the median of the posterior means of their CCF values. Based on whether or not this median value is greater than 0.9, we assign clonal or subclonal status to all SNVs of the cluster, respectively. For a cluster with only one SNV, we obtain the posterior median of its CCF value and compare this value to 0.9 to determine its clonal status.
2. PyClone 75th-tile: For an SNV cluster with multiple SNVs, we obtain the 75th percentile of the posterior means of their CCF values. For a cluster with only one SNV, we obtain the 75th percentile of the posterior draws of its CCF value. We compare these values with 0.9 to determine the clonal status of the SNVs.
3. PyClone 95th-tile: For an SNV cluster with multiple SNVs, we obtain the 95th percentile of the posterior means of their CCF values. For a cluster with only one SNV, we obtain the 95th percentile of the posterior draws of its CCF value. We compare these values with 0.9 to determine the clonal status of the SNVs.

NMCS, on the other hand, determines the clonal status of a mutation based on the confidence interval of its CCF value. If the 95% CCF confidence interval of a mutation overlaps 1, the mutation is classified as clonal; otherwise, the mutation is classified as subclonal.

Figure 5 summarizes the accuracy, sensitivity and specificity of all methods on the 100 simulated tumor samples. We find that PyClone performs well in terms of accuracy and sensitivity, while NMCS is better in terms of specificity. Mutstats, on the other hand, achieves good performance on all three metrics. Specifically, the accuracy and sensitivity of Mutstats are comparable to those of PyClone, and the specificity of Mutstats is clearly higher than that of PyClone. Compared to NMCS, although Mutstats has a somewhat lower specificity, it has significantly higher accuracy and sensitivity. Overall, Mutstats is able to combine the strength of PyClone and NMCS.

Furthermore, Mutstat is an ultra-fast method. Table 3 shows the average running time of the three methods. The running time of Mutstats is similar to that of NMCS (in seconds) and on average can be 9000 times faster than PyClone (which runs in hours). Moreover, Figure 6 shows the ratio of classification accuracy to running time for each method and each simulated sample. The higher the ratio, the better the method in terms of balancing between classification accuracy and computation time. On average, Mutstats has the highest accuracy-to-time ratio, and on 92 out of 100 samples, Mutstats outperforms PyClone and NMCS.

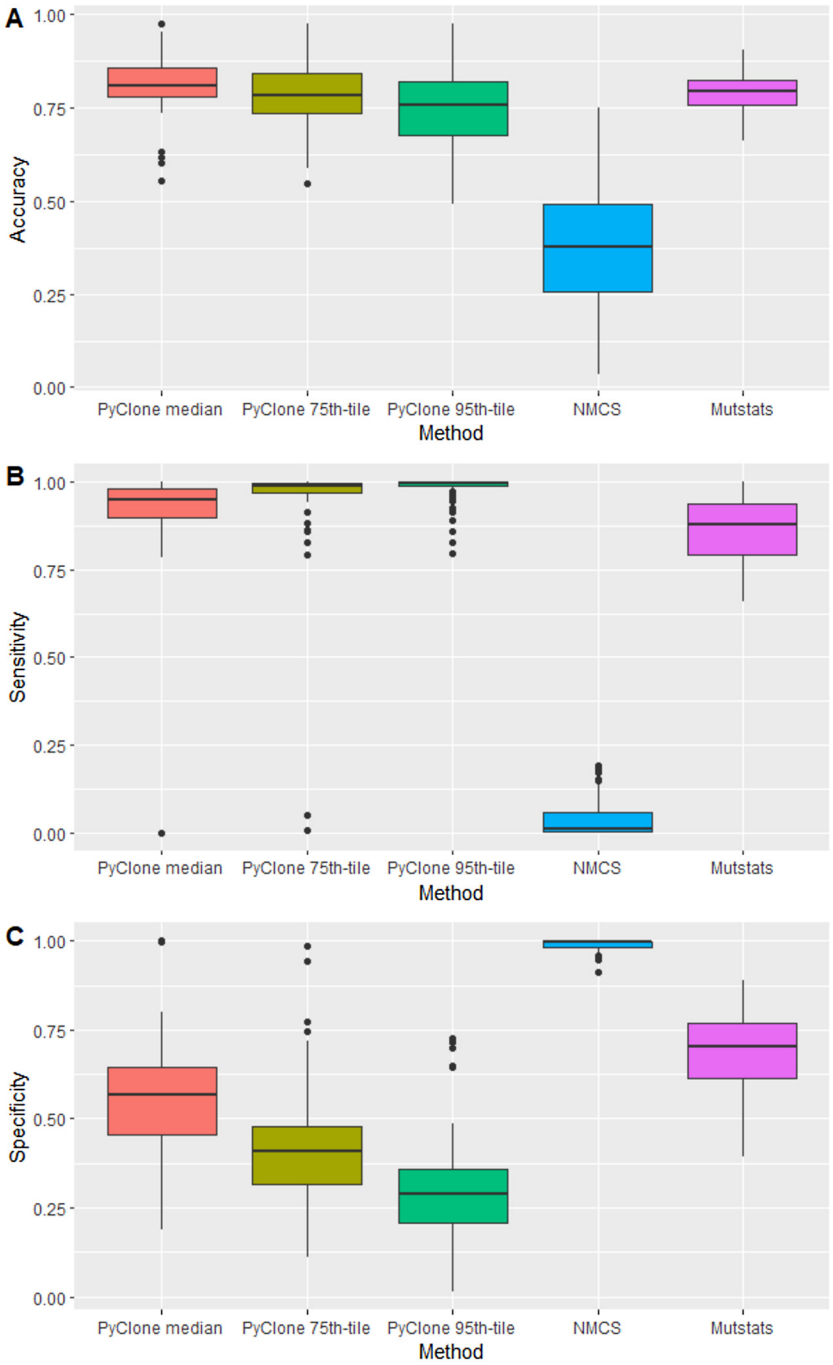


Figure 5: Boxplot of the accuracy, sensitivity and specificity values of the three methods.

5 TCGA BRCA Data Analysis

To demonstrate the practical usage of our method, we apply Mutstats to the analysis of two breast invasive carcinoma (BRCA) samples from The Cancer Genome Atlas (TCGA) project. The two samples are from the same TCGA donor (A15E), with one sample dissected from the

Table 3: The running time of the PyClone, NMCS and the Mutstats method in seconds.

	PyClone	NMCS	Mutstats
Mean	21,270.33	2.25	2.29
Median	12,007.50	2.09	2.01
SD	19,301.68	0.95	1.19

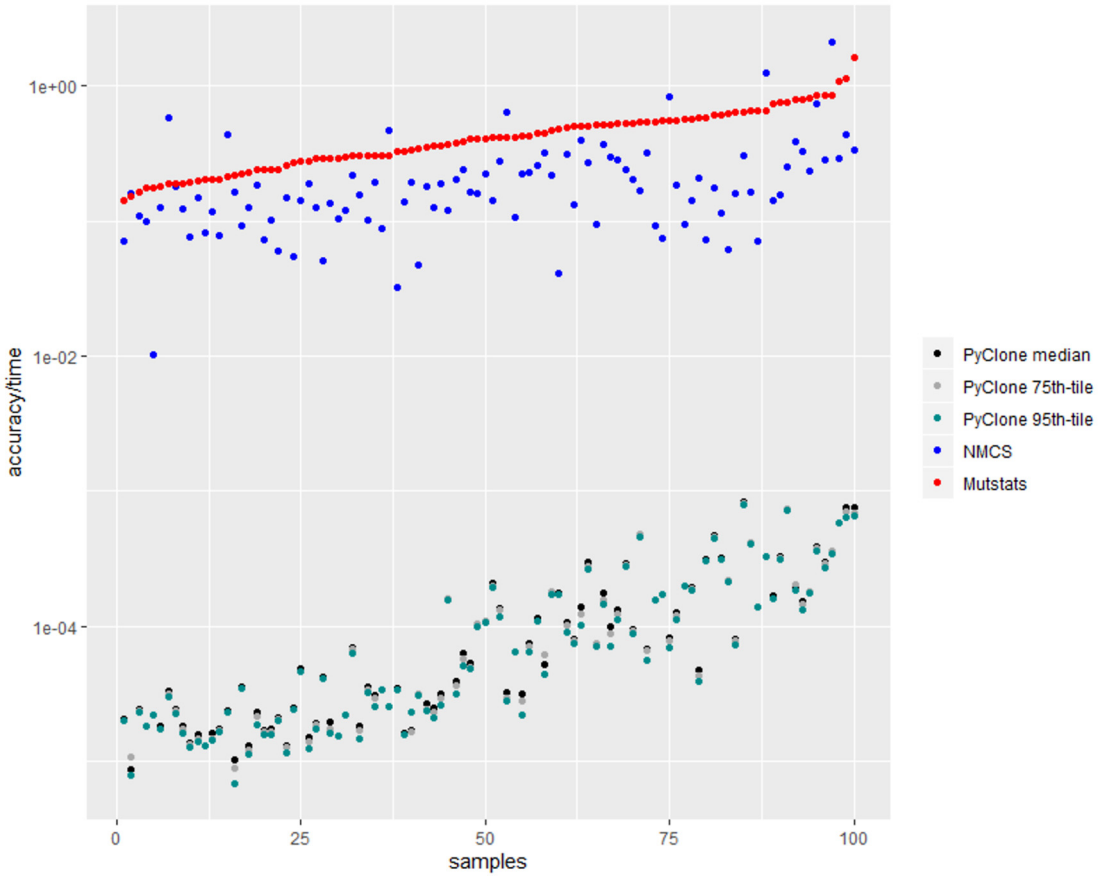


Figure 6: Scatterplot of the accuracy/time vs. the sorted index of the samples according to the values of accuracy/time for the Mutstats method.

primary tumor site and the other dissected from the metastatic tumor site. From the TCGA website, we download `bam` files for the primary tumor, metastatic tumor and matched normal; all are whole-genome sequencing data with an average depth of coverage 25×–30×. Using somatic mutation caller `MuTect` (Cibulskis et al., 2013) integrated inside `GATK 3.6` (McKenna et al., 2010), loci of the SNVs for the two tumor samples are detected. We also record the total number of reads and number of variant reads mapped to these loci. Next, we employ purity and copy number caller `Battenberg` (Nik-Zainal et al., 2012) to retrieve CNA information and ASCNs for each SNV. We remove the loci with copy numbers greater than 17. At the end of this procedure, we have 12,720 SNVs for the primary tumor sample and 15,248 SNVs for the metastatic tumor sample. Among these SNVs, 5,534 are shared between the two samples.

Table 4: Count and proportion of the clonal and subclonal mutations determined by the three methods for the primary and metastatic samples of the A15E donor.

Method	Clonal	Subclonal	Total	Time
Primary Sample				
Mutstats	53.57%	46.43%	12,720	< 1 min
NMCS	31.77%	68.23%	12,720	< 1 min
PyClone	61.45%	38.55%	2,000*	43 min*
Metastatic Sample				
Mutstats	0.27%	99.73%	15,248	< 1 min
NMCS	27.78%	72.22%	15,248	< 1 min
PyClone	0.00%	100.00%	2,000*	48 min*

*: The full sample requires longer than a week, thus, we randomly sampled 2,000 SNVs to compare with the other methods.

We run Mustats, PyClone, and NMCS on both samples to determine the clonal status of the SNVs. The threshold value H_1 is chosen based on a FDR threshold of $f_0 = 0.05$ (see Equation 7). For H_2, we keep the same value of $H_2 = 0.16$ as in the simulation studies. In addition, due to the large number of SNVs in the data, we randomly selected 2,000 SNVs from both the primary and the metastatic samples to reduce the computation burden of the PyClone algorithm. The results are shown in Table 4 below. All three methods yield similar distributions of SNV classifications for the Metastatic sample. For the primary sample, Mutstats is also similar to the PyClone method. Furthermore, for the proposed Mutstats method, among the 5,534 shared mutations, 1,834 are identified as having the same clonal status in both samples with 1,826 subclonal and 8 clonal mutations. On the other hand, 3,700 shared mutations are classified differently in the primary and metastatic samples. In particular, 3,696 SNVs are classified as clonal in primary but subclonal in metastatic, while 4 are classified as subclonal in primary but clonal in metastatic. Lastly, among the unique mutations, Mutstats identifies that 4,991 are subclonal and 4,962 are clonal in the primary sample, and 12,446 are subclonal and 35 are clonal in the metastatic sample. In summary, compared to the primary tumor, the metastatic tumor has a much larger portion of subclonal mutations and a greater degree of intra-tumor heterogeneity. This finding suggests that the tumor progression is worrisome.

6 Discussion

We have made three major contributions in this paper. First, we have clarified several terminologies used by the community and have provided their definitions from a unified model (Table 1). Second, we have developed a simulator from that unified model to generate realistic tumor data with multiple subclones and different purities. Finally, we have presented an ultra-fast method to distinguish between clonal and subclonal mutation given read count data and output from a copy number caller. With the help of the tumor data simulator, we have evaluated our method on 1000 synthetic tumors. We have also run our algorithm on two real samples from the same TCGA patient.

In our method, the two threshold values H_1 and H_2 expressed the confidence on classifying the current SNV as subclonal and clonal, respectively. Through sensitivity analysis, we found that a reasonably large H_1 around 0.9 and reasonably small H_2 around 0.2 provide a good tradeoff of sensitivity and specificity.

The speed of Mutstats is the main attractive feature for practical applications. For example, compared with the popular PyClone method, Mutstats achieves over 9,000 fold reduction in computing time in the comparative study in §4.4. The main reason is that Mutstats avoids lengthy MCMC computation that is required by the PyClone method.

The proposed Mutstats methods can be modified and extended in several ways. First, we have utilized ASCN information from the `Battenberg` caller to determine the clonal status of a mutation. Although empirically we find that ASCN information improves our classification accuracy, many other copy number callers do not provide such information. In the absence of ASCN information, we may replace $\tilde{q}_k$ and $\rho_1\tilde{q}_k^1 + \rho_2\tilde{q}_k^2$ in Equations (5) and (6) by the average copy number $\bar{l}_s$. Intuitively, if $\bar{l}_s$ is large, then m_s should also be large for SNV s to be clonal, which can be determined by whether $\Pr(\hat{m}_s^{\text{rep}} > \bar{l}_s \mid \hat{m}_s, \boldsymbol{\theta})$ is greater than some threshold. Another interesting future direction is to develop alternative machine learning algorithms for clonal status classification. The proposed tumor simulator can generate a large number of tumor samples that resemble real-world datasets. The simulated tumor samples have labeled clonal status (ground truth) thus can be used as training data to train a machine learning algorithm for classification of the mutations in a real tumor sample.

Understanding the clonal status of mutations is very important for understanding the overall evolution process of a tumor. Such knowledge is essential for understanding of timing of mutations. By finding clonal status for actionable driver mutations, we hope to improve future drug design and strategies for advanced treatment in cancer.

Supplementary Material

We include an Appendix on the Bayes model used by the PyClone method. In addition, the simulation data can be obtained from the website https://compgenome.shinyapps.io/tumorsim. Finally, the code of the Mutstats method and the real data used in this analysis can be found in the author's Github page https://github.com/edwardbi/Mutstats.

Funding

Yuan Ji's research is partly supported by NIH R01 CA132897.

References

Barnett DW, Garrison EK, Quinlan AR, Strömberg MP, Marth GT (2011). Bamtools: a C++ api and toolkit for analyzing and managing bam files. *Bioinformatics*, 27(12): 1691–1692.

Beerenwinkel N, Schwarz RF, Gerstung M, Markowetz F (2014). Cancer evolution: Mathematical models and computational inference. *Systematic Biology*, 64(1): e1–e25.

Carter SL, Cibulskis K, Helman E, McKenna A, Shen H, Zack T, et al. (2012). Absolute quantification of somatic DNA alterations in human cancer. *Nature Biotechnology*, 30(5): 413–421.

Cibulskis K, Lawrence MS, Carter SL, Sivachenko A, Jaffe D, Sougnez C, et al. (2013). Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology*, 31(3): 213–219.

Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM (2010). The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic Acids Research*, 38(6): 1767–1771.

Dempster AP, Laird NM, Rubin DB (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, Methodological*, 39(1): 1–38.

Deshwar AG, Vembu S, Yung CK, Jang GH, Stein L, Morris Q (2015). PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biology*, 16(1): 35.

Fan Y, Xi L, Hughes DS, Zhang J, Zhang J, Futreal PA, et al. (2016). MuSE: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data. *Genome Biology*, 17(1): 178.

Fraley C, Raftery A, Scrucca L (2016). mclust: Gaussian mixture modelling for model-based clustering, classification, and density estimation. URL https://CRAN.R-project.org/package=mclust. *R package version*, 5: 1.

Huang W, Li L, Myers JR, Marth GT (2012). Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4): 593–594.

Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, et al. (2012). VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research*, 22(3): 568–576.

Landau DA, Carter SL, Stojanov P, McKenna A, Stevenson K, Lawrence MS, et al. (2013). Evolution and impact of subclonal mutations in chronic lymphocytic leukemia. *Cell*, 152(4): 714–726.

Lee J, Müller P, Sengupta S, Gulukota K, Ji Y (2016). Bayesian inference for intratumour heterogeneity in mutations and copy number variation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(4): 547–563.

Marusyk A, Polyak K (2010). Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA) – Reviews on Cancer*, 1805(1): 105–117.

McGranahan N, Favero F, de Bruin EC , Birkbak NJ, Szallasi Z, Swanton C (2015). Clonal status of actionable driver events and the timing of mutational processes in cancer evolution. *Science Translational Medicine*, 7(283): 283ra54–283ra54.

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, et al. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9): 1297–1303.

Melnykov V, Maitra R, et al. (2010). Finite mixture models and model-based clustering. *Statistics Surveys*, 4: 80–116.

Misale S, Yaeger R, Hobor S, Scala E, Janakiraman M, Liska D, et al. (2012). Emergence of KRAS mutations and acquired resistance to anti-EGFR therapy in colorectal cancer. *Nature*, 486(7404): 532–536.

Nik-Zainal S, Van Loo P, Wedge DC, Alexandrov LB, Greenman CD, Lau KW, et al. (2012). The life history of 21 breast cancers. *Cell*, 149(5): 994–1007.

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

Qin M, Liu B, Conroy JM, Morrison CD, Hu Q, Cheng Y, et al. (2015). Scnvsim: somatic copy number variation and structure variation simulator. *BMC Bioinformatics*, 16(1): 1–6.

Roth A, Khattra J, Yap D, Wan A, Laks E, Biele J, et al. (2014). PyClone: statistical inference of clonal population structure in cancer. *Nature Methods*, 11(4): 396–398.

Schmitt MW, Loeb LA, Salk JJ (2016). The influence of subclonal resistance mutations on targeted cancer therapy. *Nature Reviews. Clinical Oncology*, 13(6): 335–347.

Schwarz G, et al. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464.

Scrucca L, Fop M, Murphy TB, Raftery AE (2016). mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *The R Journal*, 8(1): 289.

Sengupta S, Wang J, Lee J, Müller P, Gulukota K, Banerjee A, et al. (2015). Bayclone: Bayesian nonparametric inference of tumor subclones using NGS data. In: *Pacific Symposium on Biocomputing*, volume 20, 467.

Shcherbina A (2014). Fastqsim: platform-independent data characterization and in silico read generation for ngs datasets. *BMC Research Notes*, 7(1): 1–12.

Shen R, Seshan VE (2016). FACETS: allele-specific copy number and clonal heterogeneity analysis tool for high-throughput DNA sequencing. *Nucleic Acids Research*, 44(16): e131–e131.

Swanton C (2012). Intratumor heterogeneity: evolution through space and time. *Cancer Research*, 72(19): 4875–4882.

The I, et al. (G of Whole TPCA, Consortium) (2020). Pan-cancer analysis of whole genomes. *Nature*, 578(7793): 82.

Xia Y, Liu Y, Deng M, Xi R (2017). Pysim-sv: a package for simulating structural variation data with gc-biases. *BMC Bioinformatics*, 18(3): 23–30.

Yates LR, Campbell PJ (2012). Evolution of the cancer genome. *Nature Reviews. Genetics*, 13(11): 795–806.

Yu Z, Du F, Ban R, Zhang Y (2020). Simuscop: reliably simulate illumina sequencing data based on position and context dependent profiles. *BMC Bioinformatics*, 21(1): 1–18.

Zhou T, Müller P, Sengupta S, Ji Y (2019). PairClone: a Bayesian subclone caller based on mutation pairs. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, 68(3): 705–725.

Zhou T, Sengupta S, Müller P, Ji Y (2020). RNDClone: Tumor subclone reconstruction based on integrating DNA and RNA sequence data. *Annals of Applied Statistics*, 14(4): 1856–1877.