Bioinformatics, 2022, 1–8
https://doi.org/10.1093/bioinformatics/btac041
Original Paper

Genome analysis

SCONCE: a method for profiling copy number alterations in cancer evolution using single-cell whole genome sequencing

Sandra Hui [1,*] **and Rasmus Nielsen** [1,2,3,*]

[1]Center for Computational Biology, University of California, Berkeley, Berkeley, CA 94720, USA, [2]Department of Integrative Biology, University of California, Berkeley, Berkeley, CA 94720, USA and [3]Department of Statistics, University of California, Berkeley, Berkeley, CA 94720, USA

*To whom correspondence should be addressed.

Associate Editor: Can Alkan

Received on September 3, 2021; revised on December 23, 2021; editorial decision on January 18, 2022; accepted on January 24, 2022

Abstract

Motivation: Copy number alterations (CNAs) are a significant driver in cancer growth and development, but remain poorly characterized on the single cell level. Although genome evolution in cancer cells is Markovian through evolutionary time, CNAs are not Markovian along the genome. However, existing methods call copy number profiles with Hidden Markov Models or change point detection algorithms based on changes in observed read depth, corrected by genome content and do not account for the stochastic evolutionary process.

Results: We present a theoretical framework to use tumor evolutionary history to accurately call CNAs in a principled manner. To model the tumor evolutionary process and account for technical noise from low coverage single-cell whole genome sequencing data, we developed SCONCE, a method based on a Hidden Markov Model to analyze read depth data from tumor cells using matched normal cells as negative controls. Using a combination of public data sets and simulations, we show SCONCE accurately decodes copy number profiles, and provides a useful tool for understanding tumor evolution.

Availabilityand implementation: SCONCE is implemented in C++11 and is freely available from https://github.com/NielsenBerkeleyLab/sconce.

Contact: sandra_hui@berkeley.edu or rasmus_nielsen@berkeley.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In cancerous cells, somatic driver and passenger single nucleotide polymorphisms (SNPs) and copy number alterations (CNAs) accumulate over time. CNAs are extremely common across cancer types (Beroukhim *et al.*, 2010; Gerstung *et al.*, 2020).

Many large-scale cancer studies are done with bulk samples, and many methods and evaluation techniques (Salcedo *et al.*, 2020; Smolander *et al.*, 2021) have been developed to identify CNAs in bulk sequencing, especially for low coverage data (Poell *et al.*, 2019) and tumor heterogeneity deconvolution (Xiao *et al.*, 2020). However, bulk sequencing averages mutations across many cells and loses the granularity and detail single-cell sequencing (SCS) can provide. Single-cell sequencing facilitates analyses treating each cell as an individual in a population. However, the SCS process is technically challenging and produces noisy low coverage data, due to challenges such as cell dissociation, small amounts of starting DNA and non-uniform whole genome amplification (Kashima *et al.*, 2020). Although the rapidly increasing availability of single-cell RNA sequencing (scRNA-seq) of tumors can yield insights into tumor subpopulations (Patel *et al.*, 2014) and relevant biological pathways and processes (Suvà and Tirosh, 2019; Tirosh and Suvà, 2019), using scRNA-seq for calling CNAs is limited to areas of the genome that are expressed at the time of sequencing and does not directly measure genomic copy number. However, single-cell whole genome DNA sequencing data promises to circumvent these problems, despite the inherent noisiness of the data.

The main components of CNA calling are detecting breakpoints between contiguous regions of the genome with the same copy number and determining the absolute copy number of each region. Previous approaches to calling CNAs using single cells have been based on Hidden Markov Models (HMMs) and change point detection (Mallory *et al.*, 2020). For example, HMMcopy use a Hidden Markov Model to segment tumor genomes, normalized by matched normal cells. Although HMMcopy was originally designed for array comparative genomic hybridization data (Lai *et al.*, 2019; Shah *et al.*, 2006), it has been widely used for single-cell sequencing data (Lai *et al.*, 2019; Mallory*et al.*, 2020).

Another method, CopyNumber (Nilsen *et al.*, 2012), was also designed for microarray use. Although CopyNumber detects breakpoints, it does not output absolute copy number calls. One strength of CopyNumber, however, is that it can be run in individual and multi-sample modes, where breakpoints are forced to be shared across all samples (Nilsen *et al.*, 2012).

A third program, DNAcopy (Olshen *et al.*, 2004; Venkatraman and Olshen, 2007), was designed for microarray use, and uses circular binary segmentation to identify breakpoints, but does not output absolute copy number calls (Olshen *et al.*, 2004; Venkatraman and Olshen, 2007). Although DNAcopy was not originally designed for single-cell sequencing data, it has been applied to such data sets (Baslan *et al.*, 2012; Navin *et al.*, 2011).

A fourth program, AneuFinder (Bakker *et al.*, 2016; Taudt, 2018), which was designed for calling CNAs in whole genome SCS data, uses an HMM (Bakker *et al.*, 2016) or breakpoint detection analysis (Taudt, 2018). To determine absolute copy number, regions are scaled to have integer copy numbers, or so the mean copy number matches a known ploidy [determined by a DNA quantification technique, such as flow cytometry (Gao *et al.*, 2016)].

Finally, SCICoNE (Kuipers *et al.*, 2020) was designed for CNA calling in whole genome SCS data. It uses a likelihood-based model to first detect breakpoints shared across cells, and then builds a CNA-based tree to determine absolute copy number values (Kuipers *et al.*, 2017, 2020).

All of these methods require dividing the reference genome into adjacent bins and using bin or cell specific GC and mappability corrections to adjust read counts and mask out 'bad' bins that exhibit extremely high or low coverage due to centromeres, telomeres or highly repetitive regions. None use stochastic models of tumor evolution to do both breakpoint detection and copy number calling. An objective of this article is to develop models for CNA calling based on explicit models of tumor evolution. The rationale is that the use of such explicit models of evolution might improve inferences similarly to what has been observed in models of molecular evolution used in phylogenetics (Felsenstein, 1981; Yang, 1993, 1994).

Because tumor cells evolve forward in time from an ancestral diploid state through mutations that only depend on the current state of the cell, CNAs are inherently governed by a (possibly time-inhomogeneous) temporal Markov process. However, the read distribution observed along the length of the genome (the spatial process) is not Markovian. To realize this, consider a mutation within a segment of DNA with copy number 4 that reduces the copy number from 4 to 3. When moving from the left to the right along the length of the genome, the copy number would then go from $4 \to 3 \to 4$. There are two transitions (breakpoints) caused by the same single CNA. In many other situations, the rate of mutation from $3 \to 4$ (as in the second breakpoint) might be low, however, because the chromosome previously was in state 4, the rate of transition back from 3 to 4 is in fact high in our example. The process along the length of the genome is not Markovian because CNAs may have finite length and each mutation may induce two breakpoints.

Even though the spatial process is not Markovian, the HMM framework is computationally convenient. An aim of this article is, therefore, to develop Markovian approximations of the spatial process that can be used for inference. We present SCONCE (Single Cell cOpy Numbers in CancEr), a method based on modeling the temporal Markovian evolutionary process and deriving a best approximating spatial HMM from this process. SCONCE also uses diploid data as a null to model the technical noise in single-cell sequencing data and can robustly learn model parameters and detect CNAs. We show on simulated data that the method more accurately estimates the copy number states of a cell than previous state-of-the-art methods, and we analyze real data to show that the observations from simulated data are mirrored by similar differences among methods in analyses of real data.

2 Theory and methods

2.1 Simulations

To robustly evaluate SCONCE, we use two simulation models, one based on treating the genome as a continuous line and modeling CNAs as duplications or deletions of line segments (Line Segment Model), and one based on dividing the genome into discrete bins (Binned Model). Of note, the assumptions of these simulation model are more realistic and differ intentionally from the models implemented in SCONCE described in Section 2.2. We simulate data and estimate parameters and copy number calls under different models, to avoid biasing method comparisons toward our method.

2.1.1 Line segment model

In the Line Segment Model, we assume a genome, G, to have a fixed maximal length, L, and be comprised of c orthologous chromosomes. Each chromosome consists of an ordered list of line segments, which have positions that can be mapped back into $[0, L]$. Amplifications create an extra copy of a chromosome or part of a chromosome. Note there is no maximum copy number imposed by this model, and copy number may go to infinity. A deletion in a chromosome erases part, or the entirety, of one or more line segments in a single chromosome. Once deleted, a segment cannot be regained. A worked example is given in Supplementary Material S1.1.

Rates of amplification and deletion in the Line Segment model: It is assumed that amplifications and deletions initiate at a constant rates φ and δ, respectively, per unit chromosome and per time unit, with lengths drawn from truncated exponential distributions with respective rates τ_a and τ_d, such that the rate at which a particular point in the region is affected by an amplification or deletion is $\frac{\varphi}{\tau_a}$ and $\frac{\delta}{\tau_d}$, respectively.

We assume that amplifications and deletions run from left to right (but by construction the same distribution is obtained if considering the process from right to left), and the truncation occurs when an amplification or deletion extends beyond the end of the chromosome. To remove edge effects, we additionally assume new events can initiate at the left start of each chromosome with the same rates, $\frac{\varphi}{\tau_a}$ and $\frac{\delta}{\tau_d}$. The total genomic rate at which amplifications and deletions occur at any point in time is then $c\left(\frac{\varphi}{\tau_a}+\frac{\delta}{\tau_d}\right)+|G|(\varphi+\delta)$.

Induced marginal process: The process, as defined here, is a Markov process with state space on the infinite set of all possible genomes. It also induces a marginal continuous time Markov process at each position in the genome, $W_t \in \mathbb{Z}$, with transition rates $q_{ij} = i\frac{\varphi}{\tau_a}$ if $j = i+1$, $q_{ij} = i\frac{\delta}{\tau_d}$ and $j = i-1$ and $j \geq 0$, and $q_{ij} = 0$ otherwise, for copy number states i and j. We notice that this is a linear birth–death process with birth rate $\frac{\varphi}{\tau_a}$ and death rate $\frac{\delta}{\tau_d}$.

2.1.2 Binned process

We also consider an alternative and simpler process, termed the *binned process*, where we assume that the genome can be divided into n bins. The state space in each bin is $\mathbb{S} = \{0, 1, 2, \ldots, k\}$, where k is the maximum copy number.

Amplifications and deletion lengths in the Binned process: We assume that the length of amplifications and deletions follows a truncated geometric distribution with parameter p. That is, given that a certain amplification/deletion occurs in bin i, the probability that it extends to the adjacent bin $i+1$ is $1-p$. If the copy number in bin i changes by amount s, the copy numbers in bins affected by the same event change from u to $u' = s+u$ if $0 \leq s+u \leq k$, $u' = 0$ if $s+u < 0$, or $u' = k$ if $s+u > k$.

Marginal process of initiation: We model the marginal process of initiation of new CNAs in each bin as a continuous time Markov chain with rate matrix $Q = \{q_{ij}\}$. The total rate of CNA initiation within any of the n bins, at time t, is then $R_t = \sum_{i=1}^{n}\sum_{j\neq Y_i(t)} q_{Y_i(t)j}$, where $Y_i(t)$ is the state in bin i at time t. Notice, that because of the assumption of geometrically distributed lengths of amplifications and deletions, the marginal process in each bin does not follow Q. Only the initiation process of new amplifications and deletions follows Q.

To ensure an approximately constant rate along the length of the chromosome, amplifications and deletions may also initiate

immediately to the left of the first bin. Such events occur at a rate of $\frac{R_t}{np}$, and the relative probability of change to state j from state $Y_0(t)$ is given by $q_{Y_0(t)j}$.

2.1.3 Read depth simulation

Both the line segment and binned models simulate observed read depth for a given number of genomic windows directly from the simulated genome, G. Read depths are drawn from a user specified negative binomial distribution.

2.1.4 Simulation datasets

We simulated five datasets under the line segment model and seven datasets under the binned model, to generate a variety of types and quantity of copy number events. Each dataset had 100 tumor cells and 100 diploid cells, where read counts from diploid cells were averaged together to form the background model. Full simulation parameter values are given in Supplementary Material S1.2.

2.2 Hidden Markov model

To detect breakpoints and call absolute copy numbers, we define a Hidden Markov Model along the length of the genome informed by a tumor cell's evolutionary history. We define the state space, $\mathbb{S}$, of the HMM as the integer tumor copy number in a given genomic bin, from 0 up to a user specified k (suggested $k = 10$), and the alphabet as the integer observed tumor read depth in that bin.

2.2.1 Emission probabilities

We model emission probabilities for tumor read counts for each bin with a negative binomial distribution (interpreted here as an overdispersed Poisson). We incorporate the mean diploid read count for each bin into the emission probabilities, to normalize for technical noise and sequencing bias. Note that having a matched diploid sample is necessary to account for sequencing errors. We assume the tumor read depth in window i for tumor cell A to be represented by random variable X_{iA}, such that

$$\mathbb{E}(X_{iA}) = \lambda_{iA} = \left(\rho_{iA} \times \frac{\mu_i}{2}\right) \times s_A + \varepsilon \tag{1}$$

$$X_{iA} \sim \text{NegBinom}(\lambda_{iA}, \sigma^2_{iA} = a\lambda^2_{iA} + b\lambda_{iA} + c) \tag{2}$$

where ρ_{iA} is the state in window i for cell A, μ_i is the mean diploid read depth in window i, ε is a constant sequencing error term, s_A is a cell-specific library size scaling factor (see Section 2.5.2), and $\{a, b, c\}$ are constants learned from diploid data (see Supplementary Section S2.2). We use a quadratic relationship between the mean and variance of read depth in Equation 1, as this approximation best fit real diploid data (Navin *et al.*, 2011) (see Supplementary Material S2.1). Therefore, the emission probability for an observed read depth, x_{iA}, is given by Equation 2.

2.3 Joint evolutionary process of two bins forward in time

In Section 2.1, we described two principled models of CNA evolution. However, neither of these models have the property that they are Markovian along the length of the genome. To construct an approximating process that is Markovian, we will first construct a process jointly affecting two bins. From this description of the joint evolution of two bins, we will then derive the approximating Markov process used as the transition probabilities in the HMM.

Consider two adjacent bins in the genome on one lineage, $(U, V) \in \{(0,0), (0,1), \ldots, (k,k)\}$, where U is the copy number in bin i, and V is the copy number in bin $i+1$. The copy numbers in these bins change through continuous time evolutionary history according to rate parameters $\{\alpha, \beta, \gamma\}$:

$$\alpha = \text{rate of } \pm 1 \text{ CNA} \tag{3a}$$

$$\beta = \text{rate of any CNA} \tag{3b}$$

$$\gamma = \text{rate of CNAs affecting both } U \text{ and } V \tag{3c}$$

These rates are encoded in a transition rate matrix $\mathbb{Q} = \{q_{(U,V),(U',V')}\}, U, V, U', V' \in \mathbb{S}$, which gives the instantaneous rate of observing a change from (U, V) to (U', V'):

$$q_{(U,V),(U',V')} = \begin{cases} \gamma(\alpha+\beta) & \text{if } (U',V') = \{\substack{(U+n,V+n)\\(U-n,V-n)\}}, n=1 \\ \gamma\beta & \text{if } (U',V') = \{\substack{(U+n,V+n)\\(U-n,V-n)\}}, n>1 \\ \alpha+\beta & \text{if } (U',V') = \{\substack{(U\pm n,V)\\(U,V\pm n)\}}, n=1 \\ \beta & \text{if } (U',V') = \{\substack{(U\pm n,V)\\(U,V\pm n)\}}, n>1 \\ r_{(U,V)} & \text{if } (U',V') = (U,V) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We set $r_{(U,V)} = -\sum_{(u',v')\neq(U,V)} q_{(U,V),(u',v')}$ such that all rows sum to 0. As only one event can occur in an infinitesimally small time interval, cases where adjacent bins are simultaneously affected by different events, such as $(U', V') = (U+n, V-n), n > 0$, have instantaneous rate 0. However, notice that any time interval > 0, can contain different changes in adjacent bins.

From this rate matrix $\mathbb{Q}$, the time-dependent transition probabilities $\mathbb{P}$ are calculated via the matrix exponential as

$$P_{(U,V),(U',V')}(t) = e^{\mathbb{Q}t} \tag{5}$$

as the solution to the Kolmogorov equations. This gives the probability of observing a transition from (U, V) to (U', V') in evolutionary time t.

2.4 Approximating Markovian process along the genome

We convert the forward-in-time process for two bins into an approximating Markov model along the length of the genome with transition probability matrix $\mathbb{M}_t = \{m_{i,i',t}\}$, $i, i' \in \mathbb{S}$, i.e. we identify the one-step transition probability of moving from state i to i' along the genome in a binned process, after a given evolutionary time t. Under the assumption that the cell has an ancestral diploid state at time $t = 0$, we set $(U, V) = (2, 2)$ and $(U', V') = (i, i')$. To ensure all rows in matrix $\mathbb{M}_t$ sum to 1, we normalize over all states W in $\mathbb{S}$, such that the one-step transition probabilities of the discrete approximating Markov process along the length of the genome are given by

$$m_{i,i',t} = \frac{P_{(2,2),(i,i')}(t)}{\sum_{W\in\mathbb{S}} P_{(2,2),(i,W)}(t)} \tag{6}$$

Given an evolutionary time t, Equation 6 defines the transition matrix for the HMM (described in Section 2.2) along the length of the genome. That is, the HMM transition matrix is fully parameterized by $\{\alpha, \beta, \gamma, t\}$.

The advantage of using a model that approximates a non-Markovian process using an evolutionary time-informed HMM over more generic HMMs is that information about the ancestral diploid state is included in the model specification, allowing, as we will show in the result section, more accurate inference of copy number state. While the model is only an approximation, as it ignores the non-Markovian nature of any realistic model of CNA changes along the genome, we will evaluate it on simulations from the aforementioned more realistic non-Markovian simulation models.

2.5 Model training

The model training has four steps, followed by the copy number profile decoding, shown in Figure 1. We first estimate the emission probability constants in Equation 2, $\{a, b, c\}$, from the diploid data. Second, for each tumor cell, A, we quickly estimate an

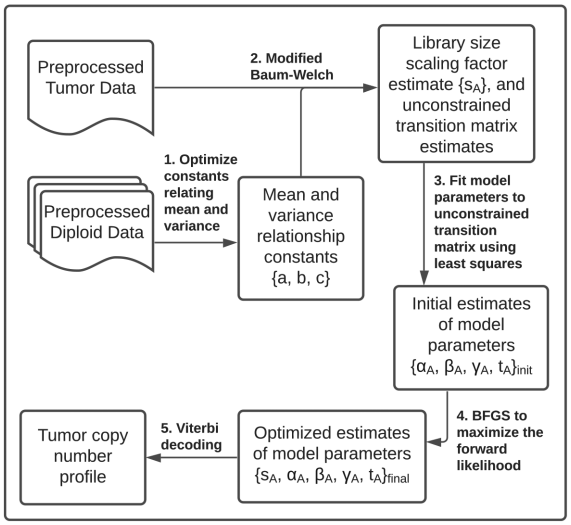

Fig. 1. Overview of the SCONCE model training procedure. Tumor and diploid sequencing files must be preprocessed into bed files of read depth per genomic window

unconstrained transition matrix, initial probability vector, and library size scaling factor, s_A, using a modification of the Baum-Welch algorithm. Third, the model rate and time parameters, $\{\alpha_A, \beta_A, \gamma_A, t_A\}$, are fit to the estimated transition matrix using least squares. Fourth, the initial estimates for $\{s_A, \alpha_A, \beta_A, \gamma_A, t_A\}$ are refined using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization algorithm to maximize the forward likelihood of the observed tumor read depths, and copy number profiles are produced from the Viterbi decoding.

2.5.1 Negative binomial mean and variance calculations

The variance and mean of the negative binomial distribution on read depth are related using a second-degree polynomial, defined in Equation 2. A second-degree polynomial was chosen to maximize the adjusted R-squared value on real diploid data (Navin *et al.*, 2011), without over-specifying the model (see Supplementary Material S2.1).

To determine the constants $\{a, b, c\}$ for a given set of observed diploid data, we calculate the per window expected mean number of reads and variance as specified in Equation 2. Next, we maximize the likelihood of the observed diploid data using the Nelder-Mead method for optimization to find the optimal values of $\{a, b, c\}$. These constants are then used for tumor emission probability calculations (see Supplementary Material S2.2 for technical optimization details).

2.5.2 Library Size scaling factors

Because each sequenced cell will have a different total number of reads, the expected number of reads for each cell needs to be scaled accordingly. Notably, we calculate these cell specific library size scaling factors in a way that accounts for changes in the distribution of reads across the genome caused by CNAs. From Equation 1, let T_A = total reads in tumor cell A (across n windows), such that

$$\mathbb{E}(T_A) = \sum_{i=1}^{n}\left[\left(\rho_{iA} \times \frac{\mu_i}{2}\right) \times s_A + \varepsilon\right] \tag{7}$$

$$\hat{s}_A = \frac{T_A - n\varepsilon}{\sum_{i=1}^{n}\left(\rho_{iA} \times \frac{\mu_i}{2}\right)} \tag{8}$$

We define ρ_{iA} as copy number in the ith window from cell A's Viterbi decoding path, updated after each iteration of the Baum–Welch algorithm, such that the library size scaling factor estimate continually incorporates changes in estimated copy number across the genome. Initial estimates of s_A are based on the ratio of tumor and average diploid library sizes, and updated according to Equation 8 in subsequent iterations of the Baum–Welch algorithm.

Because s_A estimation can get stuck in local minima, we use several initial estimates of $s_{A,\text{initial},b}; b \in \{1, 2, 3\}$. The first is set to $s_{A,\text{initial},1} = \frac{\text{cell } A \text{ library size}}{\text{average diploid library size}}$. Subsequent starting points are set to $s_{A,\text{initial},2} = 2 \times s_{A,\text{final},1}, s_{A,\text{initial},3} = 4 \times s_{A,\text{final},1}$. Skipped and rarely visited intermediate copy number states (e.g. overwhelmingly observing even copy number states genome wide, with odd states observed at 0 or near 0 frequencies) are hallmarks of a local minima for s_A. Estimates of $s_{A,\text{final}}$ that display this pattern are excluded, and the $s_{A,\text{final}}$ estimate with the highest likelihood is used (see Supplementary Material S3 for further details on filtering $s_{A,\text{final}}$ values).

2.5.3 Modified Baum–Welch

We use the standard Baum-Welch algorithm to estimate the transition matrix and initial probability vector, resulting in unconstrained estimates of the transition matrix and initial probability vector. However, we do not use Baum–Welch to directly estimate an emission probability matrix, as emission probabilities are governed by Equation 1, which is only affected by s_A estimates (calculated by Equation 8; see Section 2.5.2).

Next, we fit our model parameters, $\{\alpha_A, \beta_A, \gamma_A, t_A\}$ to the estimated transition matrix using least squares to minimize the sum of squared errors between the Baum-Welch estimated transition matrix and the transition matrix determined by the model parameters.

2.5.4 BFGS parameter estimation and inferring CNAs

Given estimates from previous steps, the parameters $\{s_A, \alpha_A, \beta_A, \gamma_A, t_A\}$ are refined for each tumor cell A, by maximizing the log likelihood (calculated using the Forward Algorithm) using the BFGS optimization algorithm, an unconstrained quasi-Newton optimization method that approximates the second derivative of the log likelihood by iteratively calculating the gradient (Fletcher, 2000) (see Supplementary Material S4 for technical details on BFGS implementation).

Finally, the most likely copy number sequence for each cell is reported using the Viterbi decoding algorithm.

We note that some of the heuristics described in the previous sections could be avoided using a full likelihood estimation using BFGS without the intermediate step of an unconstrained Baum–Welch optimization. However, we find that such optimization is slower, as the Baum–Welch optimization is substantially faster than the BFGS optimization. In addition, using model parameters fitted to the Baum–Welch results using least squares, without BFGS refinement, results in inaccurate CNA calling, thereby showing the importance of well estimated model parameters (see Supplementary Material S5). Finally, using multiple starting points, as described above, was found to be necessary to avoid the optimization getting stuck in local, but not global, optima.

2.6 Real data preprocessing

We applied SCONCE to two published datasets aligned to hg19 [which was discretized into non-overlapping 250 kb uniform bins using bedtools (Quinlan and Hall, 2010)]. The first consists of 34 diploid cells (as determined by cell sorting), and 4 tumor subpopulations (24, 24, 4 and 8 cells, respectively) from one triple negative breast cancer patient (Navin *et al.*, 2011), a cancer type with prevalent CNAs (Li *et al.*, 2020). The second consists of 10k cells across 5 sections of one triple negative ductal carcinoma sample (10x Genomics, 2019). Section A was treated as the diploid sample, as determined by (10x Genomics, 2018). Standard data preprocessing and quality control steps were used to prepare the raw data (see Supplementary Material S6 for details). For both real and simulated datasets, we used the averaged diploid cells to calculate the negative binomial distribution constants, $\{a, b, c\}$, and as the matched

normal sample to determine the somatic copy number for each tumor cell.

2.7 Other methods

To evaluate the accuracy of the inference procedure, we compared SCONCE with HMMcopy (Lai *et al.*, 2019; Shah *et al.*, 2006), CopyNumber (Nilsen *et al.*, 2012), DNAcopy (Olshen *et al.*, 2004; Venkatraman and Olshen, 2007), AneuFinder (Bakker *et al.*, 2016; Taudt, 2018) and SCICoNE (Kuipers *et al.*, 2020). We limited our comparison to methods that have previously been used on the (Navin *et al.*, 2011) dataset (Baslan *et al.*, 2012), and that, similarly to SCONCE, do not require bam files or SNPs. See Supplementary Material S7 for full details for running each method.

3 Results

3.1 GC content and mappability

Because GC content and sequence mappability can bias read distributions, many methods explicitly incorporate corrections for GC content and sequence mappability. However, any technical noise that would affect the tumor sequencing would also affect the diploid sequencing obtained using the same technology, so in SCONCE, these corrections are already directly accounted for in our emission probabilities via the diploid mean.

To verify this, we examined prediction accuracy of expected tumor read counts per window with different amounts of information. For window i, let μ_i be the mean diploid read count, ζ_i be the GC content, and η_i be the mappability from the Duke Uniqueness of 35 bp Windows from ENCODE/OpenChrom (UCSC accession wgEncodeEH000325) (Derrien *et al.*, 2012; Dunham *et al.*, 2012). For each tumor cell, A, from the previously published data in Navin *et al.* (2011), we predicted the ith window tumor read depth, x_{iA}, using various linear regressions on $\{\mu_i, \zeta_i, \eta_i\}$, then calculated the sum of squared errors (SSE) between predicted and actual tumor read depths. Boxplots of the SSE per cell are shown in Figure 2 and empirical cumulative distribution function (ECDF) plots are shown in Supplementary Figure S4 for A : $x_{iA} \sim \mu_i$, B : $x_{iA} \sim \mu_i + \zeta_i$, C : $x_{iA} \sim \mu_i + \eta_i$, D : $x_{iA} \sim \mu_i + \zeta_i + \eta_i$, E : $x_{iA} \sim \zeta_i$, F : $x_{iA} \sim \eta_i$, G : $x_{iA} \sim \zeta_i + \eta_i$.

The sum of squared errors remains consistently low across models that incorporate the diploid mean (models A, B, C and D), and have overlapping ECDF plots, while the SSE increases for models that depend solely on GC content and mappability (models E, F and G). Because adding the GC content and mappability did not perform significantly differently from the diploid mean alone (two sample KS-test on the cumulative distribution of SSE, $D = 0.033333, P - \text{value} = 1$), we conclude that using the diploid mean is sufficient, and do not add GC or mappability corrections. This conclusion is robust to changes in window size and binning method (i.e. uniformly sized bins versus variably sized bins with equal numbers of uniquely mappable bases).

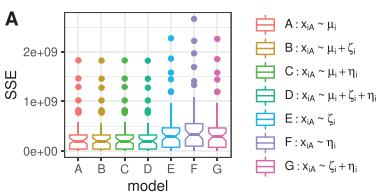

Fig. 2. Sum of squared errors (SSE) is shown for each linear regression of observed tumor read depth in window i and cell A (x_{iA}) on mean diploid read depth (μ_i), GC content (ζ_i) and mappability (η_i). SSE is calculated from the differences between the predicted read count and observed read count for each tumor cell in Navin *et al.* (2011) (uniformly sized 250 kb bins). No statistically significant difference in error is observed by adding GC or mappability information to the diploid null model

3.2 Absolute copy number accuracy

To compare the accuracy of each copy number calling method, we compared the absolute copy number accuracy, scaled copy number accuracy and breakpoint accuracy across eleven simulated datasets. For brevity, representative simulation datasets are shown in Figure 3, and accuracy results across all simulation sets are shown in Supplementary Figures S5–S7. Recall that these datasets were simulated under a more realistic non-Markovian model (described in Section 2.1) that differs from any of the models compared here, including SCONCE. There is, therefore, no reason to presume that the results are particularly biased toward favoring SCONCE because of a match between estimation and simulation model assumptions.

To measure absolute copy number accuracy, we calculated the sum of squared errors (SSE) between true copy number and estimated copy number for each cell and method across all windows. Because CopyNumber and DNAcopy do not output absolute copy number calls, their results were optimally scaled and shifted to minimize SSE. In addition, DNAcopy does not output any calls in regions of 0 reads, so these regions were excluded from all SSE calculations for DNAcopy. Overall, SCONCE has similar or lower error rates than AneuFinder, and consistently significantly lower error rate than CopyNumber, DNAcopy, HMMcopy and SCICoNE (Fig. 3).

For example, in Simulation Set H (consisting of many very short and spiky events per cell under the binned model, described in Supplementary Table S2H and Supplementary Fig. S1H; Fig. 3A), the median SSE for SCONCE is 579.00, 67.00 and 66.50, for $k = 5$, 10, 15, respectively. Of note, because SCONCE cannot call copy numbers above the user specified k, its error rate is significantly

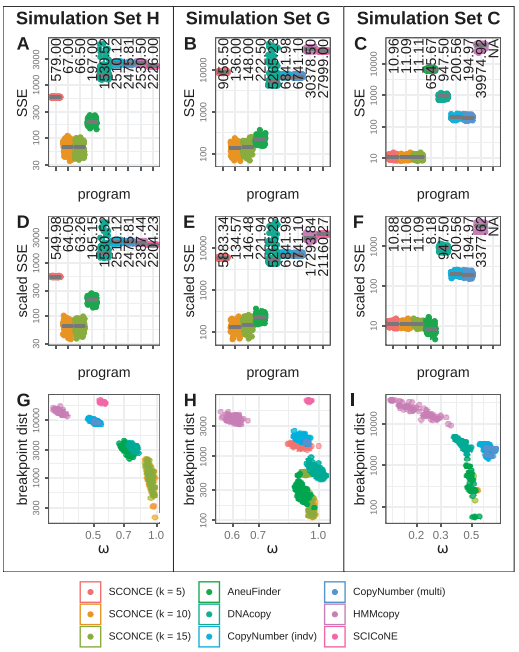

Fig. 3. Various accuracy results are shown for three simulation parameter sets, H (consisting of very short and spiky CNAs under the binned model; Supplementary Table S2H and Supplementary Fig. S1H), G (many overlapping CNAs with maximum $k = 8$ under the binned model; Supplementary Table S2G and Supplementary Fig. S1G) and C (mainly large deletions under the line segment model; Supplementary Table S1C), in the first, second and third columns, respectively. In the first row, the sum of squared errors (SSE) between simulated ploidy and estimated ploidy is shown across parameter sets. Each dot represents the error for one cell and the median SSE is shown with a gray line and printed at the top of each column. In the second row, the SSE is optimally scaled and shifted for the same datasets for each method to remove errors due to scaling. In the third row, ω (defined in Equation 9) and total breakpoint distance is shown for each method. Each dot represents one cell, colored by method. SCONCE consistently has lower SSE values, ω values closer to 1 and lower total breakpoint distance compared to other methods

higher when the true simulated copy number is greater than k (e.g. in the $k=5$ case). The median SSE values for $k=10$, 15, however, are lower than the median SSE of 197.00 for AneuFinder. Meanwhile, the median SSE values for HMMcopy and SCICoNE were 2530.50 and 2226.00, respectively, while the scaled median SSE values for CopyNumber (in individual and multisample modes, respectively) were 2510.12 and 2475.81, and scaled median SSE of 1530.52 for DNAcopy.

In Simulation Set G (made of many overlapping CNAS with maximum $k=8$ under the binned model, described in Supplementary Table S2G and Supplementary Fig. S1G; Fig. 3B), SCONCE with $k=10$, 15 outperforms all other methods, with median SSE values of 136.00 and 148.00, respectively. As expected, SCONCE with $k=5$ has a higher median SSE of 9056.50 as its inference is limited by the maximum k value. Meanwhile, AneuFinder, HMMcopy and SCICoNE have median SSE values of 222.50, 30 378.50 and 27 999.00, respectively. DNAcopy and CopyNumber (in individual and multisample modes) have scaled median SSE values of 5265.23, 6841.98 and 6741.10.

In Simulation Set C (consisting of mainly deletions under the line segment model, described in Supplementary Table S1C; Fig. 3C), AneuFinder and HMMcopy have scaling problems, while SCONCE does not. The median SSE values for SCONCE are 10.96, 11.09 and 11.11 for $k=5$, 10, 15, but the median SSE values for AneuFinder and HMMcopy are 6545.67 and 39 974.92. Both AneuFinder and HMMcopy tend to incorrectly double copy number estimates. The scaled median SSE values for DNAcopy and CopyNumber (in individual and multisample modes) were 947.50, 200.56 and 194.97. Of note, SCICoNE could not detect any breakpoints in this dataset and so could not completely run to produce any copy number profiles.

3.3 Scaled copy number accuracy

To check if the differences in median SSE between methods were due to scaling issues, we applied the previously described scaling and shifting procedure to minimize the SSE between true simulated copy number and estimated copy number for all methods. The median SSE values for CopyNumber and DNAcopy did not change here, as their outputs were already scaled and shifted. With this optimal rescaling, SCONCE consistently outperforms or is on par with other methods.

Although the median SSE for SCONCE with $k=5$ in Simulation Set H (Supplementary Table S2H and Supplementary Fig. S1H) decreases from 579.00 to 549.98, rescaling does not address the underlying upper limit on copy number as determined by k (Fig. 3D). Similarly, under Simulation Set G (Supplementary Table S2G and Supplementary Fig. S1G), rescaling SCONCE with $k=5$ causes the median SSE to drop from 9056.40 to 5883.34, but it does not address same the root problem (Fig. 3E). The median SSEs for the other methods for Simulation Set H and G (Fig. 3D and E) also decrease, but not significantly.

In contrast, the median SSE values for AneuFinder and HMMcopy for Simulation Set C (Supplementary Table S1C) drops significantly from 6545.67 to 8.18, and from 39 974.92 to 3377.62, respectively, while the median SSE for SCONCE changed only slightly, to 10.88, 11.06 and 11.09 for $k=5$, 10, 15. This shows AneuFinder's high median SSE values for Simulation Set C were due to incorrect scaling, rather than incorrect breakpoint detection and segmentation. However, although HMMcopy's median SSE value dropped by an order of magnitude by from rescaling, the remaining high median SSE value implies other issues remain, such as poor breakpoint detection.

3.4 Breakpoint detection accuracy

To evaluate program accuracy without the confounding factors of absolute or scaled copy number estimates, we compared the breakpoint detection accuracy between each program, by measuring the total distance between true and inferred breakpoints, penalized by the number of inferred breakpoints relative to the number of true breakpoints. Specifically, for each true breakpoint, we calculated the distance to the nearest inferred breakpoint in either direction, and summed this distance across the genome. Because inferring many false positive breakpoints would artificially decrease this breakpoint distance, we defined ω as

$$\omega = \frac{\text{inferred breakpoints}}{\text{true breakpoints}} \tag{9}$$

such that lowest total breakpoint distance and ω values closest to 1 indicate highest breakpoint detection accuracy.

Across simulation sets, SCONCE consistently has ω values closest to 1 and total breakpoint distances that are lower or on par with other methods. For example, in Simulation Set H (Supplementary Table S2H and Supplementary Fig. S1H; Fig. 3G), ω values for SCONCE for $k=5$, 10, 15 all cluster near 1, with median ω values of 0.9302, 0.9321 and 0.9321, respectively. Median ω values for AneuFinder, DNAcopy, CopyNumber (in individual and multisample modes), HMMcopy and SCICoNE, are 0.7330, 0.7828, 0.5067, 0.5045, 0.3408 and 0.5451, respectively. In addition, SCONCE has the lowest median total breakpoint distance across $k=5$, 10, 15 values (1028.0, 1013.5 and 1020.0), while median total breakpoint distances for other programs (in the same order as above) are 3119.5, 3253.5, 9255.5, 9501.0, 13 941.0 and 20 087.0. Of note, although SCONCE with $k=5$ had a higher median SSE value than AneuFinder for this dataset because many true copy numbers were above 5 (Fig. 3A), SCONCE still outperformed AneuFinder in terms of breakpoint detection accuracy.

Furthermore, in Simulation Set G (Supplementary Table S2G and Supplementary Fig. S1G; Fig. 3H), SCONCE has median ω values closest to 1 for $k=10$, 15 (0.9499 and 0.9441) and lowest median total breakpoint distances (185.0 and 209.5). However, for $k=5$, SCONCE is unable to detect additional copy number changes for regions with copy number greater than 5, leading to median $\omega=0.8908$ and median distance of 1468. AneuFinder, DNAcopy, CopyNumber (in individual and multisample modes), HMMcopy, and SCICoNE had median ω values of 0.9, 0.9889, 0.8983, 0.9278, 0.6034 and 0.9444, respectively, and median total breakpoint distances of 287, 568, 1904.5, 1608, 3834 and 7439.

In addition, in Simulation Set C (Supplementary Table S1C; Fig. 3I), SCONCE (for $k=5$, 10, 15), AneuFinder and DNAcopy have similar median ω values of 0.49, 0.49, 0.4911, 0.4828 and 0.431. CopyNumber (in individual and multisample modes) has higher median ω values of 0.649 and 0.6638, while HMMcopy has a median ω value of 0.1798. Despite CopyNumber's better ω values, it has much worse median total breakpoint distances (2629 and 2533) than SCONCE (253, 253 and 251.5 for $k=5$, 10, 15) and AneuFinder (301). DNAcopy has a similar median total breakpoint distance of 3299, while HMMcopy is an order of magnitude worse, at 23 925. Due to the absence of copy number calls, SCICoNE is excluded from this panel. Of note, the similar results between SCONCE and AneuFinder are consistent with AneuFinder performing poorly (Fig. 3C) in this setting mostly due to scaling problems. In contrast, these results suggest a combination of scaling and breakpoint errors lead to HMMcopy's poor performance.

Full plots and tables of median ω and median breakpoint distances across all simulation datasets are given in Supplementary Material S10.

3.5 Genome wide decodings

By plotting the genome wide copy number profile for a representative cell from each simulation set, we can learn more about the specific differences between methods that lead to differing error rates. For brevity, only genome decodings for SCONCE (with $k=10$) and AneuFinder are shown in the main text, as AneuFinder consistently performed the best out of other methods (see Supplementary Fig. S8 for decodings with other programs and other values of k for SCONCE across all datasets).

SCONCE is more sensitive to small CNAs. For example, for cell 54 in Simulation Set G (described in Supplementary Table S2G and Supplementary Fig. S1G; Fig. 4A), SCONCE correctly identifies small CNAs that AneuFinder and other methods miss, on chromosomes 10 (right arrow), 11, 12 and 15, ranging in size from 6 to 13 windows (comparisons to other methods are shown in

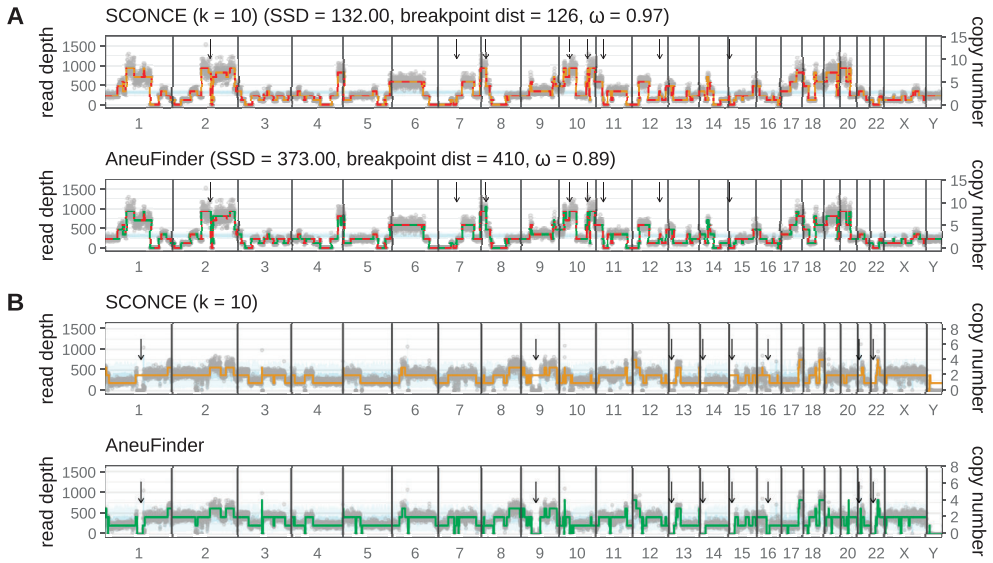

Fig. 4. Genome wide copy number decodings are shown for representative cells from simulations and real data. Cell 54 from simulation Set G (many overlapping CNAs with $k = 8$ under the binned simulation model, described in Supplementary Table S2G and Supplementary Fig. S1G) is shown in (**A**), and cell SRR054570 from Navin *et al.* (2011) is shown in (**B**). Genomic window is plotted along the *x*-axis, per window read depth is shown along the left *y*-axis, and copy number is plotted along the right *y*-axis. Black vertical lines denote chromosome boundaries, gray dots represent observed tumor read depth in each window, the red dotted line denotes the true copy number from simulation (where applicable), the light blue line shows the mean diploid read count, the light blue band shows ±1 standard deviation in the diploid read count, and the colored lines denote the copy number decoding from each method. Black arrows highlight regions with differences in CNA calls between SCONCE and AneuFinder. Genome decodings from other methods and additional datasets are shown in Supplementary Material S11

Supplementary Fig. S8G). In one cell from Simulation Set H (described in Supplementary Table S2H and Supplementary Figs S1H and S8H), SCONCE has a total breakpoint distance at least one order of magnitude smaller than all other methods and ω value closest to 1. In particular, CopyNumber and SCICoNE only call about half as many breakpoints as necessary, while HMMcopy only calls about a third, resulting in high breakpoint distances for all three methods. DNAcopy and AneuFinder have similar total breakpoint distances and predict about three quarters of the necessary breakpoints, but still struggle to call small events.

A similar effect plays out in the real data. For example, by examining cell SRR053675 from the (Navin *et al.*, 2011) dataset in Supplementary Figure S9B, small CNAs (between 5 and 22 250 kb windows in length, on chromosomes 9, 10, 12, 13 and 18) are consistently missed by other methods, while SCONCE calls these. In addition, for the cell with barcode AAACCTGGTTCTTTGT-1 from the (10x Genomics, 2019) dataset, shown in Supplementary Figure S9C, SCONCE detects copy number events on chromosomes 6, 10, 13, 17, 21 and 22 that are not detected by other methods, with sizes ranging from 5 to 20 windows.

SCONCE also calls CNA breakpoints closer to the true breakpoints. In cell 54 from Simulation Set G (Fig. 4A), SCONCE detects breakpoints more accurately than AneuFinder [arrows on chromosomes 2, 7, 8 and 10 (left)], with differences ranging from 3 to 35 windows in size. In cell 95 from Simulation Set J (consisting of many overlapping CNAs, with $k = 8$ and uniform initialization matrix under the binned model, described in Supplementary Table S2J and Supplementary Figs S1J and S8J), although AneuFinder, DNAcopy and CopyNumber all have ω values close to 1, they all have higher total breakpoint distance values than SCONCE (with $k = 10$, 15), resulting from erroneously shifting the boundaries of each CNA. HMMcopy is unable to predict enough copy number events, while SCICoNE predicts too many. In both cases, CNAs are predicted in incorrect positions.

As noted before, the value of k must be set high enough to allow a wide enough copy number range in SCONCE. For example, in cell 54 from Simulation Set G (Supplementary Fig. S8G), this limitation can be seen in chromosomes 1, 2, 4, 10 and 17–20, where the true copy number reaches a maximum of 8, but SCONCE's copy number estimates are limited to $k = 5$. However, once k is set large enough, SCONCE accurately predicts the true copy number state.

In addition, in simulations with mostly deletions (Simulation Set C, under the line segment model, described in Supplementary Table S1C), AneuFinder and HMMcopy consistently and incorrectly double the estimated copy number, leading to high SSE values, while SCONCE does not (Fig. 3C, Supplementary Fig. S8C). Specifically, AneuFinder and HMMcopy mainly call copy numbers of {0, 2, 4}, instead of {0, 1, 2}. As in Section 3.3, AneuFinder's scaled SSE values dropped, thereby verifying the existence of a scaling problem. In contrast, HMMcopy's remaining large-scaled SSE values are caused by not predicting enough CNAs, resulting in high total breakpoint distances and low ω values.

Furthermore, SCONCE considerably outperforms methods like AneuFinder, DNAcopy, HMMcopy and SCICoNE in regions of 0 tumor read coverage. By using the diploid null model, we are able to separate between true deletions and areas that have missing data due to sequencing noise, and make the most parsimonious calls rather than assuming copy number 0. For example, AneuFinder consistently predicts copy number 0 for centromeres and telomeres, highlighted with arrows in Figure 4B in the centromeres of chromosomes 1, 9 and 16, and in the telomeres of chromosomes 13, 14, 15, 21 and 22. In all panels of Supplementary Figure S9, DNAcopy completely skips telomeres with no tumor coverage, HMMcopy occasionally predicts copy number 0 for entire chromosomes when one telomere is missing, and SCICoNE inconsistently predicts copy number 0 for centromeres and telomeres. We note that this problem observed in the real data was not contributing to the performance of these methods in the simulated data, as no regions with missing diploid data were simulated.

4 Discussion

CNAs are an important driver in cancer evolution, and accurately detecting them on a single cell level can deepen our understanding of tumorigenesis. In this article, we derive several models of CNAs for inference and simulation. We show that using HMMs derived from models of the evolutionary process that generate CNAs, more accurate inferences of CNA could be obtained. The method for inference

based on these models, SCONCE, is available as an opensource computer package at https://github.com/NielsenBerkeleyLab/sconce.

One limitation of SCONCE is that it requires data from diploid cells sequenced on the same platform as the tumor cells. While this increases accuracy by accounting for platform-specific biases and single-cell sequencing errors, it also potentially increases sequencing costs to sequence diploid cells, which may not be directly of interest to investigators. However, diploid single cells are often produced incidentally as a by-product of the tumor sequencing strategy. This is, for example, true for the two real datasets analyzed here. In such cases, there is no extra cost involved in the use of diploid cells for calibration.

Another limitation of SCONCE is that no allele specific or phasing information is used. Incorporating allele frequency and genotype likelihoods of heterozygous sites can increase confidence and clarity in copy number calls, and is the subject of future work.

One of the key strengths of SCONCE over competing methods is its principled Markovian approximation to the copy number process along the length of the genome. This allows for future interpretations and applications of model parameters to understand tumor evolution. Specifically, SCONCE learns transition rate parameters $\{\alpha, \beta, \gamma\}$, time t and library size scaling factors, and we note that these evolutionary parameters could potentially be used directly for estimating phylogenies.

Compared to other methods, SCONCE has increased sensitivity in calling very small CNAs, particularly those smaller than 5500 kb. In addition, in cells with substantial copy number losses, SCONCE can accurately create copy number profiles without erroneous copy number doublings. This is due to SCONCE's method of estimating library sizes using the Viterbi decoding to account for how changes in the copy number profile necessarily impact the library scaling factor.

Furthermore, because SCONCE uses the averaged diploid data as a null model, in regions with zero tumor read coverage, it can differentiate between genomic loss and sequencing noise, which other methods cannot do. In particular, in regions with diploid coverage but no tumor reads, SCONCE calls copy number 0 and in regions without coverage in either the diploid cells or the tumor cell, SCONCE makes the most parsimonious call. This increases CNA calling accuracy of hard to sequence regions, such as telomeres, centromeres, and repetitive regions.

In conclusion, we present an accurate and principled evolutionary model for calling CNAs in single-cell whole genome sequencing of tumors, with implications for broader applications.

Funding

This work was supported by the National Institutes of Health [R01GM138634-01 to R.N.].

Conflict of Interest: none declared.

Data availability

SCONCE is implemented in C++11 and is freely available from https://github.com/NielsenBerkeleyLab/sconce. See Supplementary Material S12 for full details.

References

10x Genomics. (2018) *Application Note – Assessing Tumor Heterogeneity with Single Cell CNV,* Document Number LIT000026 Rev A, 10x Genomics.

10x Genomics. (2019) *Breast Tissue nuclei sections A-E (v1, 84x100)*, Breast Tissue nuclei Breast Tumor Tissue Demonstration Dataset by Cell Ranger 1.1.0.

Bakker,B. *et al.* (2016) Single-cell sequencing reveals karyotype heterogeneity in murine and human malignancies. *Genome Biol.*, **17**, 115.

Baslan,T. *et al.* (2012) Genome-wide copy number analysis of single cells. *Nat. Protoc.*, **7**, 1024–1041.

Beroukhim,R. *et al.* (2010) The landscape of somatic copy-number alteration across human cancers. *Nature*, **463**, 899–905.

Derrien,T. *et al.* (2012) Fast computation and applications of genome mappability. *PLoS One*, **7**, e30377.

Dunham,I. *et al.* (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57–74.

Felsenstein,J. (1981) Journal of molecular evolution evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, **17**, 368–376.

Fletcher,R. (2000) *Practical Methods of Optimization*. Chichester, England: Wiley.

Gao,R. *et al.* (2016) Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. *Nat. Genet.*, **48**, 1119–1130.

Gerstung,M. *et al.*; PCAWG Consortium. (2020) The evolutionary history of 2,658 cancers. *Nature*, **578**, 122–128.

Kashima,Y. *et al.* (2020) Single-cell sequencing techniques from individual to multiomics analyses. *Exp. Mol. Med.*, **52**, 1419–1427.

Kuipers,J. *et al.* (2017) Single-cell sequencing data reveal widespread recurrence and loss of mutational hits in the life histories of tumors. *Genome Res.*, **27**, 1885–1894.

Kuipers,J. *et al.* (2020) Single-cell copy number calling and event history reconstruction. *bioRxiv,* page 2020.04.28.065755.

Lai,D. *et al.* (2019) HMMcopy: copy number prediction with correction for GC and mappability bias for HTS data.

Li,Z. *et al.* (2020) Comprehensive identification and characterization of somatic copy number alterations in triple-negative breast cancer. *Int. J. Oncol.*, **56**, 522–530.

Mallory,X.F. *et al.* (2020) Methods for copy number aberration detection from single-cell DNA-sequencing data. *Genome Biol.*, **21**, 208.

Navin,N. *et al.* (2011) Tumour evolution inferred by single-cell sequencing. *Nature*, **472**, 90–94.

Nilsen,G. *et al.* (2012) Copynumber: efficient algorithms for single- and multi-track copy number segmentation. *BMC Genomics*, **13**, 591.

Olshen,A.B. *et al.* (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557–572.

Patel,A.P. *et al.* (2014) Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, **344**, 1396–1401.

Poell,J.B. *et al.* (2019) ACE: absolute copy number estimation from low-coverage whole-genome sequencing data. *Bioinformatics*, **35**, 2847–2849.

Quinlan,A.R. and Hall,I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)*, **26**, 841–842.

Salcedo,A. *et al.*; DREAM SMC-Het Participants. (2020) A community effort to create standards for evaluating tumor subclonal reconstruction. *Nat. Biotechnol.*, **38**, 97–107.

Shah,S.P. *et al.* (2006) Integrating copy number polymorphisms into array CGH analysis using a robust HMM. *Bioinformatics*, **22**, e431–e439.

Smolander,J. *et al.* (2021) Evaluation of tools for identifying large copy number variations from ultra-low-coverage whole-genome sequencing data. *BMC Genomics*, **22**, 1–15.

Suvà,M.L. and Tirosh,I. (2019) Single-cell RNA sequencing in cancer: lessons learned and emerging challenges. *Mol. Cell*, **75**, 7–12.

Taudt,A.S. (2018) Hidden Markov models for the analysis of next-generation-sequencing data. PhD thesis, University of Groningen, Groningen.

Tirosh,I. and Suvà,M.L. (2019) Deciphering human tumor biology by single-cell expression profiling. *Annu. Rev. Cancer Biol.*, **3**, 151–166.

Venkatraman,E.S. and Olshen,A.B. (2007) A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, **23**, 657–663.

Xiao,Y. *et al.* (2020) FastClone is a probabilistic tool for deconvoluting tumor heterogeneity in bulk-sequencing samples. *Nat. Commun.*, **11**, 1–11.

Yang,Z. (1993) Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Mol. Biol. Evol.*, **10**, 1396–1401.

Yang,Z. (1994) Journal of molecular evolution maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J. Mol. Evol.*, **39**, 306–314.