Monovar: single-nucleotide variant detection in single cells

Hamim Zafar1,2,4, Yong Wang3,4, Luay Nakhleh1, Nicholas Navin2,3 & Ken Chen2

Current variant callers are not suitable for single-cell DNA sequencing, as they do not account for allelic dropout, false-positive errors and coverage nonuniformity. We developed Monovar (https://bitbucket.org/hamimzafar/monovar), a statistical method for detecting and genotyping single-nucleotide variants in single-cell data. Monovar exhibited superior performance over standard algorithms on benchmarks and in identifying driver mutations and delineating clonal substructure in three different human tumor data sets.

Next-generation sequencing (NGS) technologies have vastly improved our understanding of the human genome and its variation in normal populations and in diseases such as cancer. However, most NGS data sets represent pooled mixtures of genomes derived from millions of cells and therefore mask variation within the tissue sample. Recently, single-cell sequencing (SCS) methods have emerged as powerful tools for resolving genomic variation in complex cell mixtures and for measuring genomic information in rare subpopulations. SCS tools have had a major impact on diverse fields of biology, including neurobiology, microbiology, immunology and development. In cancer research, SCS methods have greatly improved our understanding of intratumor heterogeneity and clonal evolution.

While there have been substantial advances in single-cell DNA and RNA sequencing technologies, computational tools are severely lacking. Some progress has been made with computational methods for estimating DNA copy number and RNA expression in single cells, but methods for calling single-nucleotide variants (SNVs) have not yet been developed. In most studies to date, standard NGS variant callers such as GATK, Samtools, SOAPsnp, SNVMix2 and Varscan2 have been applied. These variant callers, designed for bulk tissue samples, make many assumptions regarding the underlying properties of the data. This is problematic for SCS data, which, on account of extensive whole-genome amplification (WGA), have unique properties and error profiles, including nonuniform coverage depth, allelic dropout (ADO) events, false-positive (FP) errors and false-negative (FN) errors, making it difficult to call SNVs accurately. Consequently, these studies have been challenged by a large number of FP and FN variant calls, and they require extensive orthogonal validation.

To improve the detection of SNVs in SCS data sets, we developed a statistical method called Monovar (Fig. 1a, Online Methods and Supplementary Software). Monovar leverages data from multiple single cells to discover SNVs with high confidence, and it mitigates the effects of uneven or low coverage. The method assumes that data from different loci are independent. For a particular locus, input data consist of an array of observed bases and corresponding base quality scores from multiple single cells. Monovar employs an efficient dynamic programming algorithm to calculate the posterior probability of the locus containing a variant, which forms the basis for SNV classification. This algorithm models the effects of FP errors specific to whole-genome amplification in calculating genotype likelihoods for homozygous genotypes. For heterozygous genotypes, effects of both allelic dropout and FP errors are accounted for. After detecting SNVs at a locus, each cell is genotyped based on the posterior probability of the genotype. After genotyping, an optional consensus-filtering step removes variants with support from only one cell. The final output is a VCF file in which each SNV is a different row, followed by a genotype vector with length equal to the number of single cells (Fig. 1a).

We first evaluated performance on three simulated SCS data sets (Online Methods and Supplementary Note), which showed that Monovar achieved higher precision compared to Samtools, GATK UnifiedGenotyper and GATK HaplotypeCaller (Supplementary Table 1). To validate performance on real data, we analyzed 12 published single-cell exomes (mean coverage depth 65× and breadth 92.7%) generated by single-nucleus exome sequencing (SNES) from an isogenic fibroblast cell line (SKN2). Exome sequencing of a population of these cells at high coverage depth (59x) and breadth (99.76%) was used to construct a reference set of variants (Supplementary Note). We calculated precision and detection efficiency (recall) on the multisample SNV callset. The detection efficiency of an algorithm is defined as the percentage of true SNVs that are discovered in the single cells, and precision denotes the fraction of SNV calls that are true positives. Monovar achieved substantially higher precision (0.8376) than the GATK HaplotypeCaller (0.6641) and Samtools (0.5845), with some advantages in detection efficiency (Fig. 1b,c and Supplementary Table 2). These data showed a major improvement in the reduction of specific FP classes, such as C:G>T:A transitions, which are the most prominent class of FP errors that arise in SCS experiments (Fig. 1d and Supplementary Fig. 1b).

Monovar also achieved the highest dbSNP precision—83.02% of the SNVs detected by Monovar, 67.55% by GATK and 60.71% by Samtools were found in dbSNP (v138) (Supplementary Table 3). Moreover, Monovar performs better than GATK and Samtools, regardless of the threshold used for calling SNVs (Fig. 1e and Table 1).

1Department of Computer Science, Rice University, Houston, Texas, USA. 2Department of Bioinformatics and Computational Biology, the University of Texas M.D. Anderson Cancer Center, Houston, Texas, USA. 3Department of Genetics, the University of Texas M.D. Anderson Cancer Center, Houston, Texas, USA. 4These authors contributed equally to this work. Correspondence should be addressed to K.C. (chen3@mdanderson.org) or N.N. (nnavin@mdanderson.org).

Received 13 October 2015; accepted 18 March 2016; published online 18 April 2016; doi:10.1038/nmeth.3835
Figure 1 | Monovar algorithm and performance in a normal cell line. (a) Monovar variant detection flowchart. (b–e) Evaluation of Monovar, GATK and Samtools for the detection of SNVs in a single-cell exome sequencing data set generated from a normal isogenic fibroblast cell line. (b) Number of FPs called by different algorithms. (c) Number of FPs called by different algorithms. (d) SNV spectrum of FP errors using different variant detection algorithms. (e) Precision-recall (detection efficiency) curve for Monovar generated by systematically varying the SNV calling threshold.

Supplementary Fig. 1a). Consistently better results over GATK HaploTpeCaller were achieved when we downsampled SKN2 data to various coverage depths (Supplementary Note and Supplementary Fig. 2). Monovar was able to detect a high percentage of true mutations with high precision in minor subclones created by intermixing (Online Methods) in silico data from subsets of normal SKN2 single cells with that from subsets of tumor cells from a patient with triple-negative breast cancer (TNBC)8 (Supplementary Note and Supplementary Fig. 3).

We used Monovar to detect somatic mutations and delineate the clonal substructure of three human tumor samples from, respectively, a patient with TNBC, a patient with muscle-invasive bladder cancer17 and a patient with childhood acute lymphoblastic leukemia (ALL)18 (Fig. 2). On single-cell exome data from 16 tumor and 20 normal cells from the TNBC patient, Monovar detected 120 synonymous and 282 nonsynonymous somatic SNVs (Supplementary Table 4). Hierarchical clustering and multidimensional scaling (MDS) identified three major tumor subpopulations that shared a common genetic lineage (Fig. 2a), as evidenced by 269 shared founder mutations that arose early in tumor evolution and unique subclonal (sub) mutations in SYNE2 and PPP2R1A (sub 1), CHRM5 and NSD1 (sub 2), and TNC (sub 3). In addition to the previously reported mutations8, Monovar detected an additional 163 clonal somatic mutations in genes including PTCRA, TLR1, ZNF581, ABCC10, KHDBRS1 and TNFAIP3, as well as subclonal mutations in ZNF266, NCO1, CSR2BP and LILRB3 (sub 1), MOGS, MANEAL and TMEM161A (sub 2), and TUBB4A and CHST7 (sub 3) (Supplementary Table 5).

On single-cell exome data from 42 tumor cells and 11 normal cells from a muscle-invasive bladder carcinoma17, Monovar detected 94 somatic mutations. Hierarchical clustering and MDS analysis identified three major subpopulations of tumor cells (subs 1–3) in addition to the normal cell population. Monovar also detected 54 subclonal mutations that were unique to each subpopulation, including mutations in KIAA1958, NFATC3, VAMP3, NOP56, CYP4A11, RPL3 and PARP4 (sub 1), ZNF785 and ATM (sub 2), and PALB2 and MTTP (sub 3) (Supplementary Table 4). Monovar identified 42 additional somatic mutations that were not detected in the original study17, including clonal cancer gene mutations in FGFR3, CNOTAP3 and ZNF708 and subclonal cancer gene mutations in PCDH19 (sub 1), ZNF785 (sub 2) and PALB2 (sub 3) (Supplementary Table 5).

We also applied Monovar to targeted single-cell DNA sequencing data from a pediatric patient18 (patient no. 3) to analyze 255 single cells. Hierarchical clustering and MDS analysis of somatic SNVs identified five major subpopulations (Supplementary Fig. 2c). In total, Monovar discovered 57 somatic mutations (Supplementary Table 4), including 28 new somatic SNVs (Supplementary Table 5). Monovar identified significant mutations in OR4C3 and GPR107 (all subclones), LRNF5, PD2LI and ZNF781 (present in subs 2, 4 and 5), DNAS7 (sub 1), LYAR and FMNLI (sub 2), RGS3 (subs 4 and 5), and ADAMTS13, PRSS3 and PD2LI (subs 2–5). Among these mutations, the clonal mutations in OR4C3 and GPR107, and the
subclonal mutations in PKD2L1, ADAMTS13, PRSS3 and RGS3 were not identified in the original study\textsuperscript{18} (Supplementary Table 5).

Our data show that Monovar represents an advance for SNV calling in SCS data sets compared to standard NGS variant callers. Recent innovations that enable the analysis of RNA from thousands of single cells in parallel\textsuperscript{19,20} will soon be extended to DNA. Monovar is capable of analyzing large-scale data sets and handling different WGA protocols, and thus it is well suited for addressing the growing need for accurate single-cell DNA variant detection. Although this study focused on cancer data, Monovar can be applied to any SCS data set\textsuperscript{2}. As SCS methods move into the clinic, we expect that Monovar will be used in situations where the accurate detection of SNVs is critical for patient care, such as cancer diagnosis and treatment, personalized medicine and prenatal genetic diagnosis.

**METHODS**

Methods and any associated references are available in the online version of the paper.

**Accession codes.** The data from this study were previously deposited to SRA under accessions: SRP046355, SRA053195, SRA051489, SRP044380.

**Note:** Any Supplementary Information and Source Data files are available in the online version of the paper.

**ACKNOWLEDGMENTS**

This work was supported by grants to N.N. from the Leukodysplasia Foundation, NCI (RO1CA169244-01), NIH (R21CA174397), Agilent University Relations, and MD Anderson Knowledge Gap and Center for Genetics & Genomics. N.N. is a Damon Runyon-Rachleff Innovator (DRR-25-13), ACS Research Scholar, T.C. Hsu Endowed Scholar and Sabin Fellow. K.C. is a Sabin Fellow and was supported by an NCI grant (R01CA172652). The study was supported by the Bosarge, Chapman and Dell Foundations and NCI (CA016672). The authors thank W. Zhou.

**AUTHOR CONTRIBUTIONS**

H.Z. was involved in all aspects. Y.W. analyzed the data. L.N. developed the algorithm. N.N. analyzed the data and wrote the manuscript. K.C. analyzed data and wrote the manuscript.

**COMPETING FINANCIAL INTERESTS**

The authors declare no competing financial interests.

Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. Navin, N.E. *Genome Res.* 25, 1499–1507 (2015).
2. Wang, Y. & Navin, N.E. *Mol. Cell* 58, 598–609 (2015).
3. Navin, N.E. *Genome Biol.* 15, 452 (2014).
4. Navin, N. et al. *Nature* 472, 90–96 (2011).
5. Garvin, T. et al. *Nat. Methods* 12, 1058–1060 (2015).
6. Stegle, O., Teichmann, S.A. & Marioni, J.C. *Nat. Rev. Genet.* 16, 133–145 (2015).
7. Brennecke, P. et al. *Nat. Methods* 10, 1093–1095 (2013).
8. Wang, Y. et al. *Nature* 512, 155–160 (2014).
9. Zong, C., Lu, S., Chapman, A.R. & Xie, X.S. *Science* 338, 1622–1626 (2012).
10. Wang, J., Fan, H.C., Behr, B. & Quake, S.R. *Cell* 150, 402–412 (2012).
11. McKenna, A. et al. *Genome Res.* 20, 1297–1303 (2010).
12. Li, H. et al. *Bioinformatics* 25, 2078–2079 (2009).
13. Li, R. et al. *Bioinformatics* 25, 1966–1967 (2009).
14. Goya, R. et al. *Bioinformatics* 26, 730–736 (2010).
15. Koboldt, D.C. et al. *Genome Res.* 22, 568–576 (2012).
16. Leung, M.L., Wang, Y., Waters, J. & Navin, N.E. *Genome Biol.* 16, 55 (2015).
17. Li, Y. et al. *GigaScience* 1, 12 (2012).
18. Gawad, C., Koh, W. & Quake, S.R. *Proc. Natl. Acad. Sci. USA* 111, 17947–17952 (2014).
19. Klein, A.M. et al. *Cell* 161, 1187–1201 (2015).
20. Macosko, E.Z. et al. *Cell* 161, 1202–1214 (2015).
**ONLINE METHODS**

**Software availability.** Monovar was implemented in Python. The source code and instructions for running Monovar are available at https://bitbucket.org/hamimzafar/monovar.

**Monovar algorithm.** Monovar is a multisample SNV calling method that takes as input aligned read data from multiple single cells. Monovar quantifies the likelihood values of alternate allele count in the population of single cells and utilizes those values to detect the presence of SNV at a particular site. The calculation of the likelihood values of alternate allele count requires summing over all possible combinations of genotype conformations, necessitating the quantification of genotype likelihood values for each cell. Each single cell is assigned the genotype with the highest value of the posterior probability, calculated via a dynamic programming algorithm.

**Model assumptions.** In a single-cell sample, sequence data at different sites are assumed to be completely independent. This assumption follows what is practiced by most of the state-of-the-art NGS SNV callers for the sake of simplicity. Sequencing and mapping being context dependent, this assumption might not always hold for real data. But this assumption should not affect our analysis, as we are interested in calling point mutations. We also assume that the data coming from different single cells are independent. At a genomic site, the mapping and sequencing errors of different reads are assumed to be independent. Since we are interested in finding SNVs, we assume that the variants are biallelic (triallelic SNVs are rare, ~0.2%).

**Calculation of genotype likelihood.** In each single cell, the sequencing data at a site contain an array of bases observed on the sequenced reads and the corresponding base qualities. Considering the variants to be biallelic, we denote the reference allele as \( r \) and the alternate allele as \( a \) at a site. For homozygous reference and variant genotypes (\( g = 0(0r) \) and \( g = 2(aa) \), respectively), the likelihood calculation does not require the effect of allelic dropout (ADO). For the case pertaining to \( g = 1(ra) \) (heterozygous variant genotype), we need to account for allelic dropout. At a genomic site \( s \), for a single cell having sequencing data \( d \) consisting of \( n \) reads, the likelihood of \( g = 0 \) and \( g = 2 \) can be calculated as

\[
\Lambda(g = 0) = p(d \mid g = 0) = \prod_{i=1}^{n} p(d_i \mid g = 0) = \prod_{i=1}^{n} \left[ e_i (1 - p_d^{rr}) / 3 + (1 - e_i) p_d^{rr} \right]
\]

\[
\Lambda(g = 2) = p(d \mid g = 2) = \prod_{i=1}^{n} p(d_i \mid g = 2) = \prod_{i=1}^{n} \left[ e_i (1 - p_d^{aa}) / 3 + (1 - e_i) p_d^{aa} \right]
\]

For the heterozygous genotype \( g = 1 \), the effect of allelic dropout is considered while calculating the genotype likelihood. We assume that the preferential nonamplification due to an ADO event can affect either of the alleles with equal probability. At a particular site, ADO affects all the reads as amplification precedes sequencing. The likelihood of \( g = 1 \) can be calculated as

\[
\Lambda(g = 1) = p(d \mid g = 1) = p_{ad} p(d \mid g = 1, ADO = \text{True}) + (1 - p_{ad}) p(d \mid g = 1, ADO = \text{False})
\]

where

\[
p(d \mid g = 1, ADO = \text{True}) = \frac{1}{2} \left[ p(d \mid g = 0) + p(d \mid g = 2) \right]
\]

\[
p(d \mid g = 1, ADO = \text{False}) = \bar{p}(d \mid g = 1) = \prod_{i=1}^{n} \left[ e_i (1 - p_d^{rr}) / 3 + (1 - e_i) p_d^{rr} \right]
\]

In equations 1 to 4, \( d_i \) represents the observed base in the \( i \)th read. \( p_d^{rr} \) represents the probability of \( \beta \) being the ‘intermediate allele’ given the genotype \( g = g^{[1]}[g^{[2]}] \). \( \beta \) is a variable that takes value from \{\( A, T, G, C \)\}. The term ‘intermediate allele’ refers to the allele that is called after amplification. In the absence of any amplification errors, \( \beta \) should be either \( g^{[1]} \) or \( g^{[2]} \). Due to the errors introduced during preparation of the sample, \( \beta \) can differ from both \( g^{[1]} \) and \( g^{[2]} \). In the context of single-cell sequencing data, \( \beta \) accounts for the FP errors introduced during the amplification process. \( \beta \) is a latent random variable and we assume that it follows a discreet four-point distribution with parameter \( p_c \) (Supplementary Table 6). \( p_c \) represents a prior probability that \( \beta \) equals an allele different from the haplotypes of the given genotype. This type of distribution has been previously proposed in the context of bulk sequencing data. \( p_{ad} \) is the prior probability of allelic dropout.

**Variant calling.** Assuming diploid single cells, \( m \) single cells contain 2\( m \) chromosomes at a site. The posterior probability of the site being an SNV, \( P_{SNV} = p(s = \text{SNV} \mid D) \) is given by the \( S_{SNV} \) probability, that at least one among 2\( m \) chromosomes contains an allele that is different from the reference allele. We introduce a random variable \( l \), named alternate allele count, which gives us the number of chromosomes containing alleles different from the reference allele. \( l \) can vary from 0 to 2\( m \).

\[
P_{SNV} = p(s = \text{SNV} \mid D) = 1 - p(l = 0 \mid D)
\]

\[
P(l = 0 \mid D) \text{ can be calculated using Bayes’ rule as}
\]

\[
p(l = 0 \mid D) = \frac{p(D \mid l = 0) p(l = 0)}{\sum_{l’ = 0}^{2m} p(D \mid l’) p(l’)}
\]

The sequencing data vector is given by \( D = \{D_1, \ldots, D_m\} \). For a random genotype vector for \( m \) cells \( \bar{g} = \{g_1, \ldots, g_m\} \), the likelihood of alternate allele count \( l \) is evaluated by

\[
\Lambda(l) = p(D \mid l) = \frac{1}{(2m)} \sum_{l_1=0}^{2m} \ldots \sum_{l_m=0}^{2m} \delta_{l (2m)}(\bar{g}) \prod_{i}^{(2m)} \Lambda(g_i).
\]
$\delta_{jk}$ is the Kronecker delta function which equals 1 if $l = k$ and equals 0 otherwise.

$$a_m(\vec{g}) = \sum_{i=1}^{m} g_i$$

is the number of alternate alleles in the genotype vector $\vec{g} = \{g_1, \ldots, g_m\}$.

To employ dynamic programming for the efficient computation of these likelihood values, we define $h_{l,j}$ as follows:

$$h_{l,j} = \left\{ \begin{array}{ll} \sum_{g_1 = 0}^{2} \cdots \sum_{g_j = 0}^{2} \delta_{j,l}(\vec{g}) \prod_{i=1}^{j} \Lambda(g_i) & \text{for } 0 \leq l \leq 2 j \\ 0 & \text{otherwise.} \end{array} \right.$$  \hspace{1cm} (8)

$h_{l,j}$ can be iteratively calculated using

$$h_{l,j} = h_{l,j-1}p(D_j | g_j = 0) + 2h_{l-1,j}p(D_j | g_j = 1) + h_{l-2,j}p(D_j | g_j = 2).$$  \hspace{1cm} (9)

The base cases are as follows:

$$h_{0,1} = p(D_1 | g_1 = 0), h_{1,1} = 2p(D_1 | g_1 = 1), h_{2,1} = p(D_2 | g_1 = 2).$$

Likelihood of alternate allele count can be obtained from $h_{l,j}$ values using

$$\Lambda(l) = \frac{h_{l,m}}{\binom{2m}{l}}. \hspace{1cm} (10)$$

This type of dynamic programming approach has previously been explored in the context of NGS data on a population of individuals.

The prior distribution on the alternate allele count is inspired by a population genetic prior:

$$p(l) = \begin{cases} \frac{\theta}{l} & 0 < l < 2m \\ \frac{1}{2} \left( 1 - \theta \sum_{i=1}^{2m-1} \frac{1}{i} \right) & \text{otherwise} \end{cases}$$  \hspace{1cm} (11)

In equation 11, $\theta$ represents population-level mutation rate, which is set to 0.001. Higher prior probability was assigned to alternate allele frequency of 0 or 1 because we expect that, at the vast majority of sites, a population of single-cell genomes will have identical homozygous genotypes. This prior can help limit false positives introduced by whole-genome amplification and sequencing, which occur randomly at single-cell level.

If the value of $p(l = 0 | D)$ is smaller than 0.05, then the site is called as variant. The variant quality score in Phred scale is computed as

$$Q_{\text{var}} = -10 \log_{10} p(l = 0 | D)$$  \hspace{1cm} (12)

Genotyping of single cells. After a site is declared to be a variant, each single cell is genotyped. For a variant site with reference allele $r$ and alternate allele $a$, the genotype of a single cell can be either of $\{rr, ra, aa\}$ corresponding to $v \in \{0,1,2\}$, indicating the number of alternate alleles. The posterior probability for the genotype of $j^{th}$ single cell, $P_{j}^{D}$, is given by

$$P_{j}^{D} = p(g_j = v | D) = \frac{p(D | g_j = v)}{p(D)}$$

$$= \frac{p(d_j | g_j = v)p(D \setminus d_j | g_j = v)}{p(D)}$$

$$= \frac{p(d_j | g_j = v) \sum_{l=0}^{2m-2+v} c_{l,v}p(l)h_{l-v,m}^j}{\sum_{l'=0}^{2m} p(D | l')p(l')}$$  \hspace{1cm} (13)

where, $c_{l,v}$ is given by

$$c_{l,v} = \begin{cases} \left( \frac{l}{v} \right) \binom{2m-l}{2-v} & \text{if } v \leq l \\ \binom{2m}{2} & \text{otherwise.} \end{cases}$$

$h_{l,v}^j$ is the value of $h_{l,v}$ calculated for $m - 1$ cells excluding $j^{th}$ cell, $\{1,2,\ldots,j-1,j+1,\ldots,m\}$. For the estimation of the posterior genotype probabilities of the single cells, the values of $h_{l,v}^j$ are recalculated for all $m$ possible subsets found by excluding one cell from the data. The genotype with the highest posterior probability is assigned to the single cell. A similar genotyping approach has been used previously for bulk sequencing data. The genotyping results are stored as a string, called genotype vector that contains one character corresponding to one single cell. The character corresponding to a single cell can be ‘0’: homozygous reference, ‘1’: heterozygous variant, ‘2’: homozygous variant and ‘×’: insufficient coverage depth.

Consensus filtering using multiple cells. To achieve a higher quality call set, a filtering step is introduced after genotyping. The consensus filter removes low-quality variants that have lower support. Depending on the genotype vector, the SNVs that are detected only in one single cell are removed as low quality. This step helps to remove spurious FP errors that occur at random positions in the single-cell data set. This step is optional but recommended for achieving a high-quality call set.

Computational complexity of variant and genotype calling.

The variant calling and genotyping step contributes to the major computational complexity of Monovar. For the variant discovery process for a site of the genome, $s$, the dynamic programming algorithm comprises most of the computation. Let us assume we have $m$ single-cell samples. The average number of reads per single cell is denoted by $n_s$. If the total number of reads at site $s$ combining all cells is denoted by $N_s$,

$$N_s = \sum_{i=1}^{m} n_{s,i}$$

then $\overline{n}_s = \frac{N_s}{m}$. During the dynamic programming, for each single cell, amount of calculation is $O(m + \overline{n}_s)$. The genotype likelihood calculation for each cell is $O(\overline{n}_s)$ and, for each single cell, we need
to fill $O(m)$ entries of the DP matrix. We need to do this for $m$

single cells. Therefore, the asymptotic complexity of the variant
discovery algorithm for a single site is $O(m^2 + m\tilde{n}_u)$, i.e., $O(m^2 + N_u)$. $N_u$ varies over different sites and the variant discovery has
linear complexity on the size of $N_u$.

In the genotyping step, Monovar genotypes each single cell at the
site $s$, where a variant has been discovered. To genotype a single
cell, we need to find the genotype likelihood, which is $O(n_t)$. Also, we need to redo the dynamic programming, excluding the
current single cell. Therefore, cost of genotyping a single cell is
$O(n_t(m^2 + N_u))$. Asymptotic complexity of genotyping $m$

single cells is given by $O(m(n_t(m^2 + N_u)))$, i.e., $O(N(m^2 + N_u))$. If we store the genotype likelihood values found in the variant
discovery process, then the asymptotic complexity of genotyping of each single cell is $O(1). O(m^2)$, i.e., $O(m^2)$. Therefore, asymptotic complexity of genotyping $m$ single cells is $O(m^3)$.

Simulation of single-cell sequencing data set. A 1-Mbp region of
chromosome 20 of the human genome (hg19) was chosen as the
reference genome. Assuming $n_{cell}$ to be the number of single cells
in the population, $n_{cell}$ synthetic genomes were constructed from
the reference genome. The SNVs introduced in synthetic single-cell
DNAs are the true SNVs. 1,000 SNVs (SNV rate 0.001/bp) were introduced in the reference region and those were shared by the
single cells. These 1,000 SNVs served as the gold standard set.
One-third of the SNVs were present in all the cells. Other one-third
SNVs were present in half of the single cells. The rest of the SNVs
had frequency less than 0.5 or 1 in the population and were either
shared by a number of single cells or present as singletons in different
single cells. Amplification errors were introduced in the single-cell
dNA. Allelic dropout rate was set to 20% and false-positive error rate was set to $3.2e-5$ (ref. 16). Paired-end sequencing reads were generated for each single cell using program dwgsim (http://
davetang.org/wiki/tiki-index.php?page=DWGSIM). Sequencing error rate was set to 0.01% while generating the reads. dwgsim also simulated base quality scores for each sequenced nucleotide. Reads were discarded at random intervals to emulate the coverage variation in single-cell sequencing data. The coverage depth of the simulated data was 24×. Three data sets varying in the number of

cells (10, 15 and 20) were generated.

Isogenic cell line data. Single-cell sequencing data from an isogenic
fibroblast cell line (SKN2) was used for the validation of
Monovar. SKN2 is an isogenic human fibroblast cell line that was
obtained from the Cold Spring Harbor Laboratory (M. Wigler).
SKN2 was cultured using Dulbecco’s Modified Eagle Medium
with 10% FBS, penicillin–streptomycin and L-glutamine. The data consisted of exome sequencing data from 12 single cells and bulk
sequencing data (reference population) from millions of cells.

Sequencing data from human tumor samples. We applied
Monovar to three different human tumor samples that were previously
published: a triple-negative breast cancer (TNBC) patient9, a
muscle-invasive bladder cancer patient17 and a childhood acute
lymphoblastic leukemia (ALL) patient18.

Sequence alignment and data processing. For the simulated data set,
raw FASTQ files were aligned to the reference genome using BWA-
MEM (v0.7.12)26. For the SKN2 data set, BWA-MEM (v0.7.12)26
was used to align the raw reads (FASTQ files) to the human genome
(hg19). For all three human tumor data sets, sequenced reads in
FASTQ format were mapped to the human genome assembly US
National Center for Biotechnology Information (NCBI) build 36
(hg18) using the Burrows-Wheeler alignment tool (BWA version
0.7.12)26 with default parameters and $sampe$ option to create SAM
files with correct mate-pair information, and read group tag that
includes sample name. Samtools (0.1.19)12 was used to convert SAM
files to compressed BAM files and sort the BAM files by chromosome
coordinates. The reads with lower mapping quality (≤40) were removed from the BAM files. This removed about 5% of the total
reads. For the SKN2 and TNBC data sets, the Genome Analysis Toolkit (GATK v1.4-37)13 was used to locally realign the BAM files
at intervals that have indel mismatches before PCR duplicate marking
with Picard (version 1.56) (http://broadinstitute.github.io/picard/).

Comparison of algorithms for performance evaluation. For the
simulated data and SKN2 data, Monovar’s performance was com-
pared against GATK11 (v3.5) and Samtools12 (v0.1.19), two widely
used NGS SNV callers. Monovar was run with default parameter
values (https://bitbucket.org/hamimzafar/monovar) on the set of
simulated data obtained from the BAM files of all single cells in the data
set. For GATK, we used two variant callers, UnifiedGenotyper and
HaplotypeCaller. Each of them was run with default parameters as per GATK best practices recommendation (https://www. 
broadinstitute.org/gatk/guide/tooldocs/org_broadinstitute_gatk_ tools_walkers_genotyper_UnifiedGenotyper.php, https://www. 
broadinstitute.org/gatk/guide/tooldocs/org_broadinstitute_gatk_ tools_walkers_haplotypecaller_HaplotypeCaller.php). For the experiments with SKN2 data, HaplotypeCaller was used in most
comparisons as per GATK best practices recommendation. For
Samtools, the Samtools mpileup command was used, followed by
bcftools for detecting variants. Maximum read depth for calling
SNV was set to 10,000. For each data set, each algorithm was run on
data pooled from all single cells in the data set.

Construction of the validation set for SKN2 data. For the SKN2
data, the gold standard variant set was constructed based on the
results of GATK and Monovar on the reference population
sequencing data. A union of the variant sets called by GATK and
Monovar consisting of 51,154 SNVs was used as the gold standard
variant set. 50,374 (98.5%) SNVs in the gold standard set were
called by both GATK and Monovar. The rationale for computing
the union is to have a gold standard variant set that is unbiased
when comparing the variant calling algorithm, ensuring a fair comparison. The set of variants that Samtools discovered from the reference
population sample was a subset of the gold standard variant set.

Downsampling experiments. DownsampleSam program of the Picard toolkit (version 1.56) (http://broadinstitute.github.io/picard/)
was used to downsample the exome sequencing data from SKN2 single cells. DownsampleSam allows a user to randomly
extract a certain percentage of reads from the original input BAM file. For example, the following command extracts 37.7% of the reads from the input sample, which has an average
coverage depth of 53×, to generate a downsampled BAM file that
has a coverage depth of 20×.

$java -jar DownsampleSam.jar I= SKN2.bam O=SKN2.20X. 
bam P=0.377$
Each single cell in the SKN2 data set was downsampled to 40×, 30×, 20× and 10×, respectively. Monovar and GATK HaplotypeCaller were run on each downsampld data set. Precision and detection efficiency were measured for each algorithm on each downsampled data set.

**Tumor-normal mixing experiments.** Six in silico mixed data sets were prepared by mixing a subset of normal SKN2 cells with a subset of tumor cells from a triple-negative breast cancer (TNBC) patient. Such mixed data sets mimic a heterogeneous DNA sample where a set of SKN2 cells forms a subclone. The SKN2 subclone size varied from 7.6% (i.e., 7.6% of the cells in the population are normal SKN2 cells) to 50%. More specifically, the number of SKN2 cells were 1, 2, 3, 6, 9 and 12, respectively, in the six mixed data sets, while keeping the number of TNBC cells fixed at 12. Monovar was run on pooled data from all the cells for each data set. Monovar’s precision and detection efficiency were measured for each data set.

**Calling somatic mutations in human tumor data sets.** For the human tumor data sets, from the set of SNVs called by Monovar, somatic mutations were identified by filtering the germline variants. The bulk normal tissue sequencing data worked as the source of germline variants for the triple-negative breast cancer and the muscle-invasive bladder cancer data sets. For the acute lymphoblastic leukemia data set, germline variants were obtained from highly targeted amplicon sequencing data.

**Clustered filtering.** A common technical artifact that occurs in single-cell sequencing data is genomic regions with clusters of false-positive (FP) mutations. These regions correlate with known areas of the human genome that have poor mappability and repetitive elements. To remove these FP artifacts from human tumor data sets, we filtered ‘clustered regions’ from the VCF files in which more than 1 SNV is detected within a 10-bp window using a custom Perl script.

**Genotype passcodes.** In order to subset mutations, a binary string ‘passcode’ is added to each line in the VCF file that represents the genotype of each sample for each mutation: homozygous variant (2), heterozygous variant (1), absence of mutation (0) and insufficient coverage depth (×). For tumor samples or normal single cells, the minimum coverage we use is 10× and the minimum number of reads required to call a variant is 3. However, to correct for high-coverage samples, we use different thresholds depending on the coverage depth. When coverage is more than 20× and less than 100×, we require a variant allele frequency of 15%. When coverage is more than 100×, we require a variant allele frequency of at least 10%. For the matched normal population sample, we require a more stringent cut off, coverage depth at least 6× and at least 2 variant alleles, in order to detect germline mutations during the filtering steps. The ‘passcode’ also indicates whether a mutation resides within the targeted region or exome region or not. An example ‘passcode’ is <01X02101X21120X>. Here ‘<’ and ‘>’ represent the start and end of the ‘passcode’, respectively. ‘E’ indicates that this mutation is within the exome or targeted region; alternatively, ‘N’ indicates that the variant is present outside the targeted region. The number and order of samples in a ‘passcode’ is the same as the sample number and order at the VCF header.

**Annotation of somatic mutations.** Mutations were annotated with ANNOVAR to integrate multiple databases and classify mutations as nonsynonymous, synonymous, intergenic and noncoding mutations. We then determined if mutations intersect with known cancer genes using the ‘intersect’ function of BEDTools. The cancer gene list was compiled from multiple sources, including the Cosmic database and the cancer gene census. We developed a custom Perl script that reads a VCF file as input and runs through the annotation steps automatically and combines all annotation results into one tab-delimited text output file. Another Perl script was used to extract ‘passcode’ and allele frequency information of each sample from the input VCF file. The final annotation output can then be imported into Microsoft Excel, R or MatLab for statistical analysis or for visualization by building a heatmap.

**Predicting the damaging impact of mutations.** To evaluate whether a mutation is likely to affect protein structure or function, we used two databases: Polyphen and SIFT. Mutations with Polyphen score >0.5 and SIFT score <0.05 were considered significant. We considered mutations that were predicted to be significant by both databases as protein structure–function damaging.

**Multidimensional scaling (MDS) analysis.** Nonsynonymous and synonymous mutations were parsed from the VCF file containing single-cell exome and targeted variant data to construct a binary distance matrix for sites where coverage depth was ≥6×. Hamming distance was used as the distance metric and missing values with no coverage were replaced by value 0.5. The resulting binary matrix was used to perform multidimensional scaling (MDS) analysis in R. The MDS coordinates 1 and 2 were plotted on the X and Y axes, respectively, to identify clusters of cells with similar genotypes or mutations.

**Hierarchical clustering and heatmaps.** A binary matrix was calculated using nonsynonymous and synonymous mutations from the single-cell genotype ‘passcode’ strings. Heterozygous and homozygous mutation sites were converted to a value of 1. For sites without mutations, we used a value 0. Sites with coverage depth less than 6× were assigned value 0.5. The heatmap was generated using the heatmap.2 function in R and two-dimensional hierarchical clustering was performed using both rows (mutations) and columns (cells).

21. Li, H. *Bioinformatics* **27**, 2987–2993 (2011).
22. Hodgkinson, A. & Eyre-Walker, A. *Genome Res.* **20**, 233–241 (2010).
23. You, N. *et al.* *Bioinformatics* **28**, 643–650 (2012).
24. Le, S.O. & Durbin, R. *Genome Res.* **21**, 952–960 (2011).
25. Nielsen, R., Korneliussen, T., Albrechtsen, A., Li, Y. & Wang, J. *PloS ONE* **7**, e37558 (2012).
26. Li, H. & Durbin, R. *Bioinformatics* **25**, 1754–1760 (2009).
27. Wang, K., Li, M. & Hakonarson, H. *Nucleic Acids Res.* **38**, e164 (2010).
28. Quinlan, A.R. & Hall, I.M. *Bioinformatics* **26**, 841–842 (2010).
29. Forbes, S.A. *et al.* *Nucleic Acids Res.* **43**, D805–D811 (2015).
30. Futreal, P.A. *et al.* *Nat. Rev. Cancer* **4**, 177–183 (2004).
31. Azadhubei, I., Jordan, D.M. & Sunyaev, S.R. *Curr. Protoc. Hum. Genet.* Chapter 7, Unit 7.20 (2013).
32. Ng, P.C. & Henikoff, S. *Nucleic Acids Res.* **31**, 3812–3814 (2003).