Strelka2: Fast and accurate variant calling for clinical sequencing applications

Sangtae Kim*[1], Konrad Scheffler*[1], Aaron L Halpern[1], Mitchell A Bekritsky[2], Eunho Noh[1], Morten Källberg[2,3], Xiaoyu Chen[1], Doruk Beyter[4], Peter Krusche[2], Christopher T Saunders[1]

*These authors contributed equally to this work. [1]Illumina, Inc., 5200 Illumina Way, San Diego, CA 92122, USA. [2]Illumina Cambridge Ltd., Chesterford Research Park, Little Chesterford, Essex CB10 1XL, UK. [3]Present address: Seven Bridges Genomics, 101 Euston Road, London NW1 2RA, UK. [4]Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92023, USA. Correspondence should be addressed to C.T.S. (csaunders@illumina.com)

We describe Strelka2 (https://github.com/Illumina/strelka), an open-source small variant calling method for clinical germline and somatic sequencing applications. Strelka2 introduces a novel mixture-model based estimation of indel error parameters from each sample, an efficient tiered haplotype modeling strategy and a normal sample contamination model to improve liquid tumor analysis. For both germline and somatic calling, Strelka2 substantially outperforms current leading tools on both variant calling accuracy and compute cost.

Whole-genome sequencing is rapidly transitioning into a tool for clinical research and diagnosis, a shift which brings new challenges for sequence analysis methods. While there has been considerable progress in developing methods to improve germline and somatic small variant calling accuracy in research applications[1–6], such methods can be further improved in many respects for the clinical whole-genome sequencing scenario. These improvements include reducing the compute cost/turn-around time of whole-genome analysis, further increasing indel calling accuracy, automating parameter tuning without expert user intervention, and reducing multiple indicators of call quality to a single confidence score for variant prioritization. Here we describe Strelka2, a variant calling method building upon the innovative Strelka somatic variant caller[7], to improve upon these aspects of variant calling for both germline and somatic analysis. We demonstrate that Strelka2 is both more accurate and substantially faster when compared to current best-in-class small variant calling methods.

Strelka2 germline and somatic analyses share a common series of high-level stages, including parameter estimation from sample data, candidate variant discovery, realignment, variant probability inference, and empirical re-scoring/filtration. The composition of these steps is described in more detail for each type of analysis in **Supplementary Fig. 1**.

Strelka2's germline analysis introduces a novel step to adaptively estimate indel error rates from preliminary allele counts in each sample, using a mixture model to estimate both indel variant mutation rates and indel noise rates from a set of error processes (**Supplementary Fig. 2**). This mixture approach mitigates the impact of context-specific indel error rate variation on variant call accuracy and obviates the need to specify a prior set of common population variants.

Similar to previous work[2,3,5,6], Strelka2's germline analysis models haplotypes to provide read-backed variant phasing and reduce the impact of sequencing noise, incorrect read mapping and inconsistent alignment. Strelka2's haplotype model uses an efficient tiered scheme for haplotype discovery,

combining the advantages of a simple model based on input read alignments[3], and a more complex model using local assembly[2,5,6], where the appropriate method is selected based on the properties of each variant locus. This tiered haplotype modeling approach is essential to optimize runtime without precision loss. The haplotype model also introduces a novel heuristic filter for sequencer phasing noise, improving the caller's robustness to a wider variety of potential sequencing artifacts.

In Strelka2's germline and somatic variant probability models, additional runtime improvements are made in the computation of read likelihoods by enumerating a small number of candidate alignments and using the maximum alignment-specific likelihood to approximate the marginal likelihood, avoiding the computational cost of a complete implementation such as a pair HMM[8]. Within the somatic variant probability model, the original Strelka method has been redesigned with a further novel innovation to account for contamination of tumor cells in the matched normal sample such that somatic recall is improved, especially for liquid tumor analysis. Consistent with the emphasis on automated sample adaption in Strelka2, the liquid tumor model is an expansion of the model's state space applied to all cases, and thus does not require prior knowledge of the normal sample contamination level.

For both germline and somatic calling workflows, the variant probability model is supplemented by a final empirical variant scoring (EVS) step, motivated in part by machine learning-based variant classification approaches[9,10]. This step uses a random forest model trained on numerous features indicative of call quality to improve precision by accounting for error phenomena that are not adequately represented in the generative variant probability model. Strelka2's EVS models are pre-trained on data from a variety of sequencing conditions to improve robustness, and produce a single aggregate score which can be used to set application-specific precision levels or prioritize variants for follow-up.

To assess its germline calling performance, we ran Strelka2 on the recent PrecisionFDA Consistency and Truth challenge data[11] and compared its results with the challenge submissions (**Fig. 1a**). This comparison shows that for the noisier sequence datasets in the Consistency challenge, Strelka2's indel accuracy is remarkably higher than the winning challenge submissions, improving upon the indel F-score of the best challenge submission by 3.1%. For the other two Truth challenge data sets with lower sequencing noise, Strelka2 still improves upon the best challenge submission with an indel F-score improvement of 0.08%. For single nucleotide variants, Strelka2 gave competitive results within only 0.05% - 0.1% of the best submissions (**Supplementary Fig. 3**). These results are striking when considering that all Strelka2 analyses used default parameters, a single read mapper and no input from population variant databases, whereas the top results of the PrecisionFDA challenge were obtained using pipelines specially trained for the challenge data or by combining results from multiple read mappers and variant callers.

To assess runtime, we benchmarked Strelka2 against a recently released high-speed GATK Haplotyper reimplementation (Sentieon DNAseq Haplotyper) that is over 10x faster than the original HaplotypeCaller[12]. On the PrecisionFDA datasets discussed above, Strelka2 was 2.1 times faster than Sentieon DNAseq Haplotyper on average on the same computer hardware while also outperforming it in accuracy, with an average F-score improvement of 2.1% for indels and 0.29% for SNVs (**Fig. 1**).

We evaluated Strelka2's somatic variant calling accuracy by mixing sequencing data of unrelated individuals to simulate impure tumor and matched normal samples. For this purpose, we used NA12878 and NA12877 to represent, respectively, the tumor and normal samples. We simulated datasets with

tumor purities of 20%, 50%, and 80%, and one matched normal sample with 90% purity. The truth set for these evaluations were the Platinum Genomes[13] variants in NA12878 where the corresponding NA12877 genotype is homozygous reference. Using the in-silico mixtures, we compared the somatic variant call accuracy of Strelka2 with a recent high-speed MuTect2[4] reimplementation (Sentieon TNseq TNhaplotyper). As summarized in **Fig. 2a**, Strelka2 shows substantially higher precision than TNhaplotyper at all recall thresholds over all test datasets, with an average F-score improvement of 29% for SNVs and 35% for indels. We note in particular Strelka2's superior tolerance to normal sample impurity, reflecting updates in Strelka2's somatic calling model to better support such contamination in liquid and late-stage solid tumor analyses. This was tested using the 80% purity tumor sample and noting the impact on somatic F-score when the normal sample purity changed from 100% to 90%. Strikingly, for TNhaplotyper the F-score dropped from 77% to 30% for SNVs and from 47% to 17% for indels. For Strelka2, the impact was substantially smaller, changing from 96% to 90% for SNVs and from 82% to 65% for indels.

We assessed runtime for Strelka2's somatic analysis and found that, as for the germline analysis, Strelka2 is substantially faster than available alternatives. For the above somatic analysis using in-silico sample mixtures, Strelka2 demonstrated an average runtime advantage of 3.2x over TNhaplotyper, which itself is over 10x faster than the original MuTect2 implementation (**Fig. 2b**)[12].

In the above analyses, we demonstrate the effectiveness of multiple statistical modeling and algorithmic innovations in Strelka2, resulting in remarkable improvements to accuracy and runtime for both germline and somatic calling. We reiterate that results were generated with default method settings appropriate for factory-scale analysis, not requiring human intervention to parameterize or connect complex sequences of tools. All results use only a reference genome and one alignment file per sample as input. Additionally, no prior variant databases are used in the calling process, reducing the potential for bias against rare variants or ancestry-dependent artifacts.

Improvements to Strelka2 continue in several areas. We have already generalized Strelka2's germline analysis for RNA-Seq (not described here) and efforts are ongoing to improve mitochondrial variant calling and to integrate with structural variant predictions. Generalization of Strelka2's adaptive indel error estimation methods to mitigate the impact of context-sensitive base-calling errors on SNV calling has been prototyped and shows promise. The application of these techniques to somatic variant analysis, while considerably more challenging, could substantially improve our ability to call very low-frequency variants. We see such adaptive parameterization improvements as complementing rather than competing with recent trends emphasizing a greater focus on empirical machine learning approaches to variant calling. Indeed, the improvement of generative sequencing error models to more closely represent the sample data should sharpen the effectiveness of downstream machine-learning approaches by reducing confounding error terms, a circumstance we have already leveraged to improve the accuracy of Strelka2.

ACKNOWLEDGMENTS

We thank Eddy Kim and Ying Wei for testing Strelka2, and Semyon Kruglyak, Ben Moore, Jared O'Connell, and Efstanthios Kanterakis for helpful discussions and comments.

AUTHOR CONTRIBUTIONS

All authors designed the algorithms and implemented the Strelka2 software. S.K., and C.T.S. designed and performed the analyses. S.K., K.S., C.T.S. wrote the manuscript with input from all authors.

COMPETING FINANCIAL INTERESTS

S.K., K.S., A.L.H., M.A.B., E.N., X.C., M.K., P.K., and C.T.S. are employees of Illumina Inc., a public company that develops and markets systems for genetic analysis.

Online Methods

Parameter estimation. *Chromosome depth estimation.* An initial step in all workflows is the rapid estimation of the sequencing depth for each chromosome, which for somatic analysis is computed only for the normal sample.

Indel error model. Indel sequencing errors are modeled in the variant calling steps below as a process which occurs independently in each read, with some fixed probability of an indel error occurring as a function of the short tandem repeat (STR) context (Supplementary Fig. 2). For germline variant calling, these error probabilities are estimated from the sequencing data of each input sample in two steps. First, mapped sequencing data are analyzed at a subset of sites across the genome to produce error counts for various sequencing contexts. Second, the counts are used to estimate the parameters of interest. For somatic variant calling, a simpler non-adaptive approach is used in which the indel error parameters are pre-set based on empirical observation of indel calling performance and error rates are a function of the homopolymer context length r only.

Error counting. At every counted site in the genome, the number of reads supporting each potential allele are accumulated by *context*. The counting process uses a read realignment strategy similar to that used by the variant calling process explained below.

Each STR tract with pattern size s and repeat count r at locus l is counted as a single observation for the context $\{s, r\}$ and the resulting locus count vector c_l for that observation (with elements $c_l(y), y \in Y$, corresponding to the observed set of alleles Y) is included in the set of locus count observations $C(s, r)$.

Error rate estimation. Every indel locus is modeled as belonging to either a *clean* state (generating essentially no indel errors) or a *noisy* state (generating indel errors independently across reads according to a set of error probabilities to be estimated), with the overall error probabilities being drawn from the resulting two-state mixture model. The allele counts, in turn, are modeled as drawn from a mixture over possible genotypes, with the genotype-specific distributions being multinomial for homozygous genotypes and mixtures of two multinomials for heterozygous genotypes. The multinomial distributions are governed by the local coverage $X = \sum_y c(y)$ and by rates selected from the vector of available error rates according to the alleles h_1 and h_2 comprising the genotype (Supplementary Fig. 2a).

For every STR context $\{s, r\}$ we define the following parameters. Together, the e parameters comprise $E(s, r)$:

- $p_n(s,r)$: probability of being in the noisy state
- $e_i(s,r)$: noisy-state probability of insertion error resulting in a non-reference variant
- $e_d(s,r)$: noisy-state probability of deletion error resulting in a non-reference variant
- $e_{\text{ref}}(s,r)$: noisy-state probability of indel error resulting in reversion to reference
- $e_{i,\text{clean}}(s,r)$: clean-state probability of insertion error resulting in a non-reference variant
- $e_{d,\text{clean}}(s,r)$: clean-state probability of deletion error resulting in a non-reference variant
- $e_{\text{ref,clean}}(s,r)$: clean-state probability of indel error resulting in reversion to reference
- $\theta(s,r)$: probability that a locus with the given context has an indel allele.

We fix all of the clean-state error probabilities to a small constant value: 1×10^{-8} and set e_{ref} to 0.01 during parameter estimation. During variant calling (described below), the value of e_{ref} is a function of the corresponding e_{i} or e_{d}. To improve robustness, the indel mutation rates θ were pre-estimated from all autosomes of a fixed human training sample and set as constant values in the workflow. The remaining parameters are estimated on a per-sample basis by maximizing the likelihood of the observed counts:

$$P(C(s,r)\,|\,E(s,r),\,p_n(s,r),\theta(s,r)) = \prod_{l:c_l\in C(s,r)} P(c_l\,|\,E(s,r),\,p_n(s,r),\theta(s,r)),$$

where $P(c_l\,|\,E(s,r),\,p_n(s,r),\theta(s,r)) = \sum_{H,N} P(N\,|\,p_n(s,r))P(H\,|\,\theta(s,r))P(c_l\,|\,N,H,E(s,r))$,

$H=(h_1,h_2)$ is a variable indicating the specific allele hypotheses under consideration and $N\in\{\text{noisy},\text{clean}\}$ is a variable indicating whether the observation was generated by the noisy or the clean state. For each possible genotype $G\in\{g_{\text{homref}},g_{\text{het}},g_{\text{homalt}},g_{\text{hetalt}}\}$ we consider only one hypothesis H, obtained by finding the two most likely (by number of supporting counts) non-reference indel alleles y_1 and y_2 and setting

$$(h_1,h_2) = \begin{cases} (ref,ref) & \text{if} \quad G = g_{\text{het}} \\ (y_1,y_1) & \text{if} \quad G = g_{\text{homalt}} \\ (ref,y_1) & \text{if} \quad G = g_{\text{hetalt}} \\ (y_1,y_2) & \text{if} \quad G = g_{\text{homref}} \end{cases}$$

The noisy-state prior is $P(N = noisy|s,r) = p_n(s,r); P(N = clean|s,r) = 1-p_n(s,r)$ and the genotype prior $P(G \mid \theta(s,r))$ is defined in the Germline Probability Model below.

For an allele y other than *ref*, y_1, and y_2, assign the error probability $e(y)$ as $e_i(s,r)$, $e_d(s,r)$, $e_{i,\text{clean}}(s,r)$, or $e_{d,\text{clean}}(s,r)$ according to the value of *N* and whether y is an insertion or deletion with respect to the reference allele. For $y = ref$, assign $e(y) = e_{\text{ref}}(s,r)$ or $e(y) = e_{\text{ref,clean}}(s,r)$ according to the value of *N*. The probabilities $e(y_1)$ and $e(y_2)$ of errors resulting in y_1 and y_2 respectively do not need to be assigned specific values due to the approximation below. Finally, let $\bar{e}(h_i)$ be the probability with which h_i is sequenced correctly, with $\bar{e}(ref) = 1 - e_i - e_d$ in the noisy state and $\bar{e}(ref) = 1 - e_{i,clean} - e_{d,clean}$ in the clean state.

When $h_1 = h_2$ (homref or homalt genotype), the likelihood of the count vector is:

$$P(c \mid E, n, h_1 = h_2) = m_c[\bar{e}(h_1)]^{c(h_1)} \prod_{y \neq h_1} e(y)^{c(y)},$$

where $m_c = \left(\sum_y c(y)\right)! / \prod_y (c(y)!)$ is the corresponding multinomial coefficient. Since this coefficient does not depend on any model parameters, it is ignored during parameter estimation.

When $h_1 \neq h_2$ (het or hetalt genotype), the likelihood is:

$$P(c \mid E, n, h_1 \neq h_2) = m_c[0.5\bar{e}(h_1) + 0.5e(h_1)]^{c(h_1)}[0.5\bar{e}(h_2) + 0.5e(h_2)]^{c(h_2)} \prod_{y \notin \{h_1, h_2\}} e(y)^{c(y)}.$$

Approximating with $\bar{e}(h_1) + e(h_1) \approx 1$ and $\bar{e}(h_2) + e(h_2) \approx 1$, this simplifies to:

$$P(c \mid E, n, h_1 \neq h_2) = m_c 0.5^{c(h_1)+c(h_2)} \prod_{y \notin \{h_1, h_2\}} e(y)^{c(y)}.$$

To reduce the number of estimated parameters, we model homopolymer repeats ($s = 1$) with repeat counts $2 \leq r \leq 16$ and dinucleotide repeats ($s = 2$) with $2 \leq r \leq 9$ as log-linear in *r*, allowing us to estimate values for $(s,r) \in \{(1,1), (1,2), (1,16), (2,2), (2,9)\}$ and interpolate between these values. The values at $(1,16)$ and $(2,9)$ are used for $r > 16$ or $r > 9$ respectively.

When using the estimated parameters for variant calling (Supplementary Fig. 2b, described below), we assume that all sites at which candidate haplotypes have been generated belong to the noisy state, so that the mixture model formulation is not needed. For this reason, only the noisy-state error probabilities are passed on for downstream use. We also fix the insertion and deletion error rates to be used for calling to the geometric mean of the insertion and deletion estimates for each STR context.

Candidate variant discovery. *Input read processing.* The input alignment files are scanned for reads. Reads are filtered out if they are marked as not passing primary analysis filters, PCR/optical duplicates, unmapped, secondary or supplemental. Indels in the remaining reads are left-shifted and normalized.

Tiered haplotype model. Each sample's ploidy introduces constraints that can be used to reduce errors due to sequencing noise, incorrect read mapping, and inconsistent alignment. In a simple haplotype model, such as that described in FreeBayes[3], candidate haplotypes can be identified from existing read alignments. More advanced haplotype generation methods are less dependent on the input read alignments, often using local assembly to identify longer consensus haplotypes, such as the haplotype generation methods used in Platypus[5], Scalpel[6] and GATK HaplotypeCaller[2], among others. Strelka2's germline caller uses a *tiered haplotype model* where a fast alignment-based approach is used to handle simpler variant loci, and an assembly-based approach is selected to improve accuracy in more complex cases.

The haplotyping steps are: detecting short clusters of sequence variation called *active regions*, generating candidate haplotypes in active regions, filtering candidate haplotypes to reduce noise, and discovering primitive SNVs and indels. Haplotyping is currently performed independently for each sample and is not available for somatic variant calling.

Active region detection. To detect active regions, we first identify loci that are likely to have variants, which we call *variant loci*. To identify variant loci, for each locus we calculate a variant evidence score while reading alignments as follows: a mismatch at locus i increases the score at i by 1, an insertion between locus i and $i+1$ increases the scores at i and $i+1$ by 4, a deletion of loci $[i, j]$ increases the scores in $[i-1, j]$ by 4, and a soft-clipped segment ending (starting) at locus i increases the scores at i and $i+1$ ($i-1$) by 4. A locus with a variant evidence score c and a coverage d becomes a variant locus if (1) $c \geq 0.35 \cdot d$ or (2) $c \geq 9$ and $c \geq 0.2 \cdot d$. Afterwards, nearby variant loci are clustered if they are no more than 13 bases of each other. For the clusters including two or more variant loci, the cluster region is further extended to the surrounding loci so that the first and last locus are not within a homopolymer or STR region. This extension is needed because alignments that do not fully span such repeats are often erroneous and relying on them may lead to generating incorrect haplotypes. To accomplish this, we detect *anchor loci* that are not variant loci and also do not belong to a homopolymer (of lengths no less than 3) or STR (of repeat unit lengths between 2 and 50). Given a cluster of variant loci, the active region is created between the closest anchor loci before and after the first and last variant loci.

Haplotype generation. Given an active region of size of 250 or smaller, haplotype generation is attempted using either the alignment-based or assembly-based model. The decision is made based on the fraction of reads that fully cover the active region (called covering reads): the assembly-based model is used if fewer than 65% of all reads that overlap with an active region are covering reads.

If the alignment-based model is selected, then for each covering read, the segment aligned to the active region is extracted as a candidate haplotype. If a candidate haplotype s is extracted from a read r, we call r a supporting read of s, such that for each candidate haplotype s a set of supporting reads is identified.

If the assembly-based model is selected, local de novo assembly is run using a de Bruijn graph approach similar to that described in TIGRA[14]. Prior to assembly, the target active region $[i, j]$ is expanded to $[i', j']$ to improve the identification of contigs which span the full locus. i' and j' are chosen as the mininum and maximum value satisfying the following conditions: $i-9 \le i' \le i$, $j \le j' \le j+9$, and there is no variant locus in $i' \le pos \le i-1$ and $j+1 \le pos \le j'$. This expansion allows the identification of assembled contigs which span the full locus by identifying those that share the same prefix (reference segment at $[i', i]$; denoted by a prefix anchor) and suffix (reference segment at $[j, j']$; denoted by a suffix anchor). All the reads that (fully or partially) overlap with the expanded active region are used as input to the assembly procedure. After assembly is finished, only the contigs including both prefix and suffix anchors are selected and the prefix and suffix anchors are removed. Each such contig becomes a candidate haplotype and the set of reads supporting the contig is identified.

Haplotype generation for an active region is considered unsuccessful if the assembly procedure is selected and assembly is unable to generate at least one non-reference candidate haplotype. If haplotype generation does not succeed, indel candidates can still be generated as detailed below without the benefit of haplotyping.

Haplotype filtration. If haplotype generation is successful (using either alignment-based or assembly-based methods), candidate haplotypes are ranked by decreasing read support; those with fewer than 3 supporting reads or ranking below the top x, for x the expected sample ploidy (assumed to be diploid in the current procedure), are excluded from further processing. If there is more than one remaining haplotype, an additional filtration step is applied to reduce candidates produced by phasing noise in the sequencing process across a homopolymer. The test assumes that the candidate haplotype with the highest read support, h_1, is true, and identifies whether the candidate haplotype with next highest level of read support, h_2, is a phasing noise artifact introduced while reading h_1. The conditions which trigger this filter are (1) h_1 and h_2 are the same length with only one mismatching basecall, (2) all reads supporting h_2 are observed on only one strand, and (3) the basecall mismatch between h_1 and h_2 occurs at one of the ends of the sequence, and causes h_2 to contain an uninterrupted homopolymer at least 11 bases long. If these conditions are met, all haplotype candidates besides h_1 are filtered from further consideration.

Primitive allele discovery. After filtration, the remaining candidate haplotypes are aligned to the reference and primitive alleles (SNVs and indels of size 50 and smaller) are annotated as *discovered*. These discovered primitive alleles are used to improve SNV and indel calling in downstream procedures.

Indel candidacy. Strelka2 uses indel candidacy as a preliminary filter to eliminate indel observations likely to have been generated by error processes. Candidate indels are considered during read realignment and indel genotyping in all samples. To become a candidate, an indel variant must minimally have 2 reads supporting it in at least one sample. If haplotype modeling is enabled, a candidate indel belonging to an active region where haplotyping was successful must also have been discovered through haplotype alignment in at least one sample. If an indel observation satisfies these conditions, Strelka2 evaluates its candidacy status using a one-sided binomial exact test, with the null hypothesis being that the indel

coverage is generated by indel error processes. The indel is considered a candidate variant if the resulting p-value is below 1×10^{-9}.

Read realignment and variant probability inference. *Read realignment.* Following the discovery of candidate alleles, reads are realigned to these candidates. This realignment step has two primary functions. The first is to generate the set of most likely alignments under the assumption that the read was generated by a particular candidate haplotype. Such alignments are used to assess the read's relative support for different indel alleles. The second function is to create a single *representative* alignment to use for SNV calling.

The alignment search uses a starting alignment provided by the input alignment file, as well as a set of intersecting candidate indels. If the read intersects at least one candidate indel, a set of alignments is built from the starting alignment by recursively toggling indels from the candidate set. Each toggling operation produces three alignments: the input alignment itself, and two alignments constructed by adding or removing the indel in question such that the input alignment is unchanged (1) to the left or (2) to the right of the indel. For efficiency, the search recursion is limited to depth 5.

Germline Probability Model. At every locus where candidate variant alleles have been proposed, Strelka2 calculates posterior probabilities for a range of hypotheses in each sample. Each germline hypothesis comprises a specific pair of alleles that determine its genotype $G \in \{g_{\text{homref}}, g_{\text{het}}, g_{\text{homalt}}, g_{\text{hetalt}}\}$, with the potential genotypes respectively corresponding to non-variants and to variants which are heterozygous, homozygous, and heterozygous with two non-reference alleles.

The posterior probability of a hypothesis H conditioned on the observed data D and Q is:

$$P(H \mid D, Q) \propto P(D \mid H, Q) P(H).$$

The hypothesis-specific likelihood $P(D \mid H, Q)$ is described under "Shared Probability Model" below. The hypothesis prior depends on the corresponding genotype prior $P(G)$, defined for variant calls in terms of the prior probability that a chromosome locus is non-reference, θ, as follows:

$$P(G) = \begin{cases} \theta & \text{if} \quad G = g_{\text{het}} \\ \theta/2 & \text{if} \quad G = g_{\text{homalt}} \\ \theta^2 & \text{if} \quad G = g_{\text{hetalt}} \\ 1 - 3\theta/2 - \theta^2 & \text{if} \quad G = g_{\text{homref}} \end{cases}$$

For SNVs, all three possible non-reference alleles are considered at any potential variant site, thus each variant genotype maps to three specific SNV hypotheses with uniform probability: $P(H) = P(G)/3$ when $G \neq g_{\text{homref}}$. For indels, up to two non-reference alleles are considered per sample and $P(G)$ is again divided between matching hypotheses.

Germline Variant Phasing. As previously noted, Strelka2 defines an active region around dense variants and infers 2 haplotypes for the region. These haplotypes are used to phase SNVs and indels within the same active region. The phasing is conducted after scoring and genotyping. For each heterozygous variant belonging to an active region, Strelka2 matches the variant alleles to the active region haplotype to appropriately phase the genotype allele order.

Somatic Model. The somatic calling model assumes that the samples are diploid. For both SNVs and indels, the normal genotype states are $G_n \in \{g_{\text{ref}}, g_{\text{het}}, g_{\text{hom}}\}$, referring to a non-variant and a variant which is heterozygous or homozygous in the normal sample, assuming no more than one variant allele in this sample. The tumor genotype states are $G_t \in \{g_{\text{nonsom}}, g_{\text{som}}\}$, referring to the absence and presence of a somatic variant in the tumor sample, respectively. The method approximates a posterior probability on the joint tumor and normal genotypes:

$$P(G_t, G_n \mid D) \propto P(G_t, G_n) P(D \mid G_t, G_n)$$

Here D refers to the sequencing data from both samples. The likelihood term above is computed by integrating over sample-specific allele frequencies

$$P(D \mid G_t, G_n) = \int_{F_t, F_n} P(D \mid F_t, F_n) P(F_t, F_n \mid G_t, G_n)$$

where F_t and F_n refer to tumor and normal allele frequencies. The allele frequency likelihood $P(D \mid F_t, F_n)$ is decomposed by sample to $P(D_t \mid F_t) P(D_n \mid F_n)$, where D_t and D_n indicate tumor and normal sample data. The sample-specific allele frequency likelihoods $P(D_t \mid F_t)$ and $P(D_n \mid F_n)$ are as described in the Shared Probability Model section below. The genotype prior probability $P(G_t, G_n)$ and the joint allele-frequency distribution $P(F_t, F_n \mid G_t, G_n)$ are detailed in the present section.

The posterior probability over tumor and normal genotypes $P(G_t, G_n \mid D)$ is used to compute the *somatic variant probability*.

$$P(G_t = g_{\text{som}} \mid D) = \sum_{G_n} P(G_t = g_{\text{som}}, G_n \mid D). \quad (1)$$

Somatic variant calls are reported jointly with associated calls for the normal sample. For this, we use the joint probability of somatic variation and the maximum likelihood normal sample genotype:

$$\max_{G_n} P(G_t = g_{\text{som}}, G_n \mid D).$$

Given the expected rate of variants between two unrelated haplotypes θ, the normal sample genotype prior $P(G_n)$ is

$$P(G_n) = \begin{cases} \theta & \text{if } G_n = g_{\text{het}} \\ \theta/2 & \text{if } G_n = g_{\text{hom}} \\ 1-3\theta/2 & \text{if } G_n = g_{\text{ref}} \end{cases}$$

where θ is 10^{-3} for SNVs and 10^{-4} for indels. Given the somatic state prior $P(G_t = \text{som}) = \gamma$, the joint sample prior is

$$P(G_t, G_n) = \begin{cases} (1-\gamma)P(G_n) & \text{if } G_t = g_{\text{nonsom}} \\ \gamma P(G_n) & \text{if } G_t = g_{\text{som}} \end{cases}$$

where γ is set to 10^{-4} for SNVs and 10^{-6} for indels. These values were chosen empirically to provide reasonable variant probabilities and are not adjusted for different samples in practice.

The prior probability on the tumor and normal allele-frequencies $P(F_t, F_n \mid G_t, G_n)$ encodes the concept that the normal sample is a mixture of diploid germline variation and noise while the tumor sample is a mixture of the normal sample and somatic variation. Let $C(F_n, G_n) = 1$ if (F_n, G_n) is $(0, g_{\text{ref}})$, $(0.5, g_{\text{het}})$, or $(1, g_{\text{hom}})$ and $C(F_n, G_n) = 0$ otherwise. The joint frequency prior is then defined as follows.

$$P(F_t, F_n \mid G_t = g_{\text{nonsom}}, G_n) = \begin{cases} 0 & \text{if } F_t \neq F_n \\ 1-\mu & \text{if } F_t = F_n \text{ and } C(F_n, G_n) = 1 \\ \mu U(F_t) & \text{if } F_t = F_n \text{ and } C(F_n, G_n) = 0 \end{cases}$$

$$P(F_t, F_n \mid G_t = g_{\text{som}}, G_n = g_{\text{ref}}) = \begin{cases} U(F_t)U(F_n \mid F_t) & \text{if } F_t \neq F_n, F_n \leq \tau F_t, \text{ and } F_n \leq \delta \\ 0 & \text{otherwise} \end{cases}$$

$$P(F_t, F_n \mid G_t = g_{\text{som}}, G_n \neq g_{\text{ref}}) = \begin{cases} U(F_t) & \text{if } F_t \neq F_n \text{ and } C(F_n, G_n) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Here, τ and δ represent contamination tolerance terms, $U(F_t)$ refers to a uniform distribution over the allowed tumor allele frequencies, $U(F_n \mid F_t)$ refers to a uniform distribution over the normal allele frequencies satisfying $F_n \leq \tau F_t$ and $F_n \leq \delta$, and μ indicates the noise term. The contamination tolerance terms are introduced to allow for contamination in the normal sample by some fraction of tumor cells. This is particularly useful for analyses of liquid tumors, where the normal sample may be contaminated by tumor cells. By default, τ and δ are set to 0.15 and 0.05. The noise term abstracts various sequencing, read mapping and assembly issues which could produce an unexpected allele

frequency shared in the tumor and normal samples. For SNVs, the noise contribution is set to a constant value $\mu_{\text{SNV}} = 5\times10^{-10}$ and for indels it is set as a function of the indel error rate $\mu_{\text{indel}} = e_{\text{ref}}{}^{2.2}$.

The continuous allele frequencies modeled above are efficiently computed by dividing each allele-pair axis into a set of equidistant points and performing the somatic probability computation over the resulting discrete point set. A resolution of 21 points per axis (i.e., points separated by 0.05) is used for all computations by default.

Shared Probability Model. All germline and somatic hypotheses can be generalized as a list of haplotypes h_i with corresponding expected frequencies f_i in each sample. We are interested in the hypothesis-specific likelihood for a given sample $P(D \mid H, Q)$ of the sample's observed set D of individual reads d_j (assumed independent) given the observed set Q of individual basecall quality scores q_j:

$$P(D \mid H, Q) = \prod_j P(d_j \mid H, q_j).$$

The likelihood for an individual read can be expressed in terms of likelihoods conditioned on each of the potential generating haplotypes:

$$P(d \mid H, q) = \sum_i f_i P(d \mid h_i, q)$$

The per-read likelihood $P(d \mid h, q)$ is the probability of an individual read d, given its associated basecall qualities q and a generating haplotype h. In a complete probabilistic implementation (e.g. using a pair HMM), this likelihood would be computed by summing over all possible pairwise alignments A in which d is aligned to h. Strelka2 saves computation by enumerating a small number of candidate alignments and using the maximum alignment-specific likelihood to approximate the marginal likelihood:

$$P(d \mid h, q) = \sum_A P(d, A \mid h, q)$$

$$\approx \max_A P(d, A \mid h, q).$$

The alignment-specific likelihood scores can be factorized as follows:

$$P(d, A \mid h, q) = P(d \mid A, h, q) P(A \mid h, q) = P(d \mid A, h, q) P(A \mid h).$$

Ignoring possible context effects and accepting the basecall quality scores at face value, the first term is a product of emission scores, $P(d \mid A, h, q) = \prod_k P(d_k \mid a_k, q_k)$, where emission scores are:

$$P(d_k \mid a_k, q_k) = \begin{cases} q_k & \text{if} \quad d_k = a_k \\ (1-q_k)/3 & \text{if} \quad d_k \neq a_k \in \{A, C, G, T\} \\ 1/4 & \text{if} \quad a_k = \text{softclip.} \end{cases}$$

Here, d_k and q_k are the k th base in d and the corresponding probability of the call being correct (obtained from the basecall quality score) and a_k is the base in h to which d_k has been aligned or, if d_k is aligned to an insertion relative to h, the corresponding base of the consensus insertion sequence. The second term is a product of state transition probabilities, using the indel error probabilities $e_i(s,r)$, $e_d(s,r)$, and $e_{\text{ref}}(s,r)$ described earlier to penalize alignments whenever the read contains an indel with respect to the generating haplotype:

$$P(A \mid h) = \prod_k \begin{cases} e_i(s_k, r_k)(1 - e_d(s_k, r_k)) & \text{for non - reference insertions} \\ e_d(s_k, r_k)(1 - e_i(s_k, r_k)) & \text{for non - reference deletions} \\ e_{\text{ref}}(s_k, r_k) & \text{for reversion to reference} \end{cases}$$

where the product is taken over all positions k at which a gap is opened and *reversion to reference* refers to indels that result in the reference allele being generated even though the generating haplotype contained a non-reference allele at position k. To compensate for reference-bias in the alignment process, the value of e_{ref} is set to a constant factor (1.8) times the corresponding e_{i} or e_{d}. The probabilities calculated in this equation are unnormalized, due to omission of corresponding terms when a gap fails to open; this is corrected by normalizing explicitly during posterior probability calculation.

Empirical scoring and filtration. The variant calling models (both germline and somatic) provide sufficiently accurate representations of the biology and sequencing process to produce an initial variant probability inference. However, there is additional information not used by the models which is nonetheless predictive of call accuracy. As a final step in the variant calling process, such additional information is extracted as a set of predictive features and used in combination with the probability calculated by the variant calling model to improve call precision. This is done by the Empirical Variant Scoring (EVS) model, a supervised random forest classifier trained on labeled data from sequencing runs performed under a variety of conditions (different sequencers, sample preparation, and coverage). The EVS model provides an aggregate quality score for each variant and allows for convenient exploration of the precision-recall curve.

For each of germline and somatic variant calling, there are two separate feature sets and trained random forest models: one for SNVs and one for indels. In contrast to dynamic re-scoring systems such as the GATK VQSR procedure[9], the EVS models are pre-trained, allowing Strelka2 to avoid the runtime cost, instability and population variant data requirement of a dynamic approach. When the EVS model is not used, simple cutoffs are applied to a set of features (not necessarily the same set used by EVS) to less precisely filter out potentially problematic calls. For details on EVS training and the full lists of EVS and hard-filter features, refer to **Supplementary Note 2**.

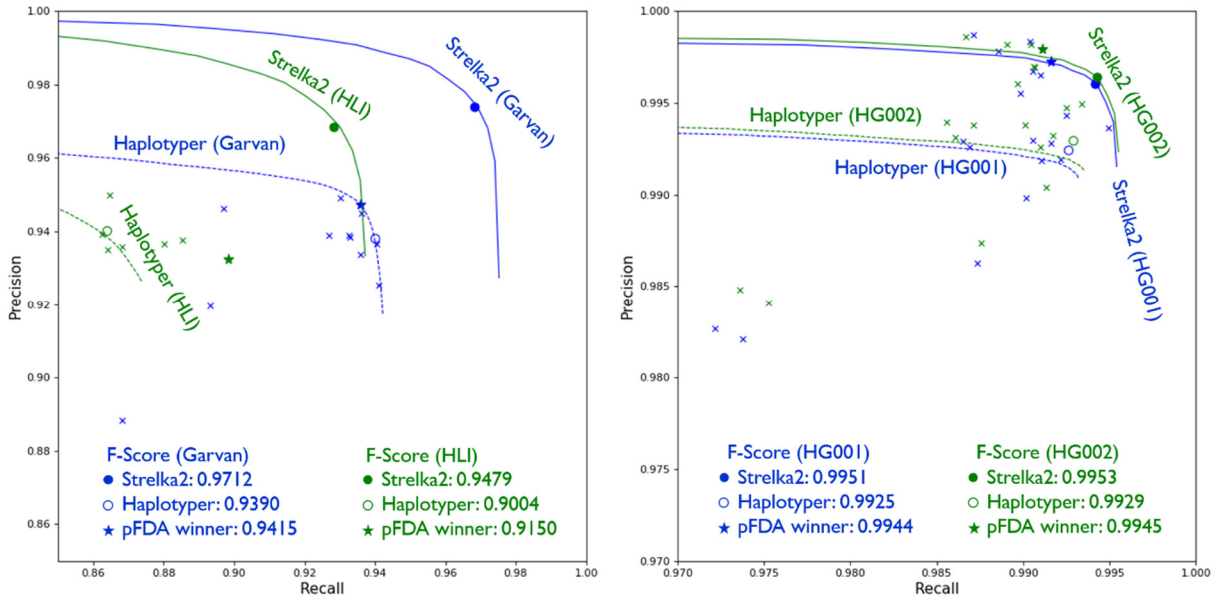
Data availability. The sequencing data and truth sets used in the germline calling benchmarking are publicly available at https://precision.fda.gov/ and https://github.com/genome-in-a-bottle. The sequencing data and truth sets used in the somatic calling benchmarking are available at http://strelka-public.s3.amazonaws.com.

References

1. McKenna, A. *et al.* The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20,** 1297–1303 (2010).
2. DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43,** 491–498 (2011).
3. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. http://arxiv.org/abs/1207.3907 (2012).
4. Cibulskis, K. *et al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.* **31,** 213–219 (2013).
5. Rimmer, A. *et al.* Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat. Genet.* **46,** 912–918 (2014).
6. Narzisi, G. *et al.* Accurate de novo and transmitted indel detection in exome-capture data using microassembly. *Nat. Methods* **11,** 1033–1036 (2014).
7. Saunders, C. T. *et al.* Strelka: Accurate somatic small-variant calling from sequenced tumor-normal sample pairs. *Bioinformatics* **28,** 1811–1817 (2012).
8. Durbin, R., Eddy, S. R., Krogh, A. & Mitchison, G. *Biological sequence analysis: Probabilistic Models of Proteins and Nucleic Acids*. (Cambridge University Press, Cambridge, UK, 1998).
9. Van der Auwera, G. A. *et al.* From fastQ data to high-confidence variant calls: The genome analysis toolkit best practices pipeline. *Curr. Protoc. Bioinforma.* **43,** 11.10.1–11.10.33 (2013).
10. Poplin, R. *et al.* Creating a universal SNP and small indel variant caller with deep neural networks. *bioRxiv* http://dx.doi.org/10.1101/092890 (2016).
11. Altman, R. B. *et al.* A research roadmap for next-generation sequencing informatics. *Sci. Transl. Med.* **8,** 335ps10 (2016).
12. Freed, D. N., Aldana, R., Weber, J. A. & Edwards, J. S. The Sentieon Genomics Tools - A fast and accurate solution to variant calling from next-generation sequence data. *bioRxiv* http://dx.doi.org/10.1101/115717 (2017).
13. Eberle, M. A. *et al.* A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27,** 157–164 (2017).
14. Chen, K. *et al.* TIGRA: A targeted iterative graph routing assembler for breakpoint assembly. *Genome Res.* **24,** 310–317 (2014).
15. Cornish, A. & Guda, C. A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference. *Biomed Res. Int.* **2015,** (2015).
16. Krusche, P. Haplotype Comparison Tools. https://github.com/Illumina/hap.py

Figures

(a)



(b)

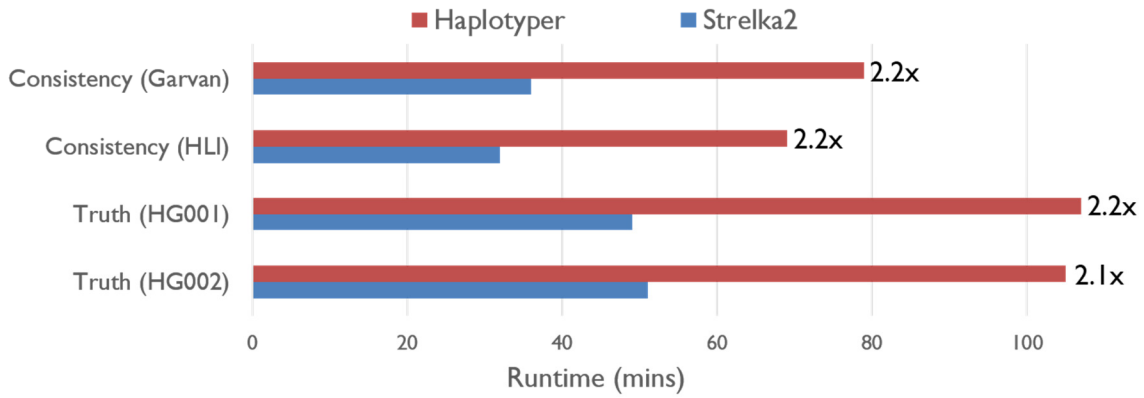
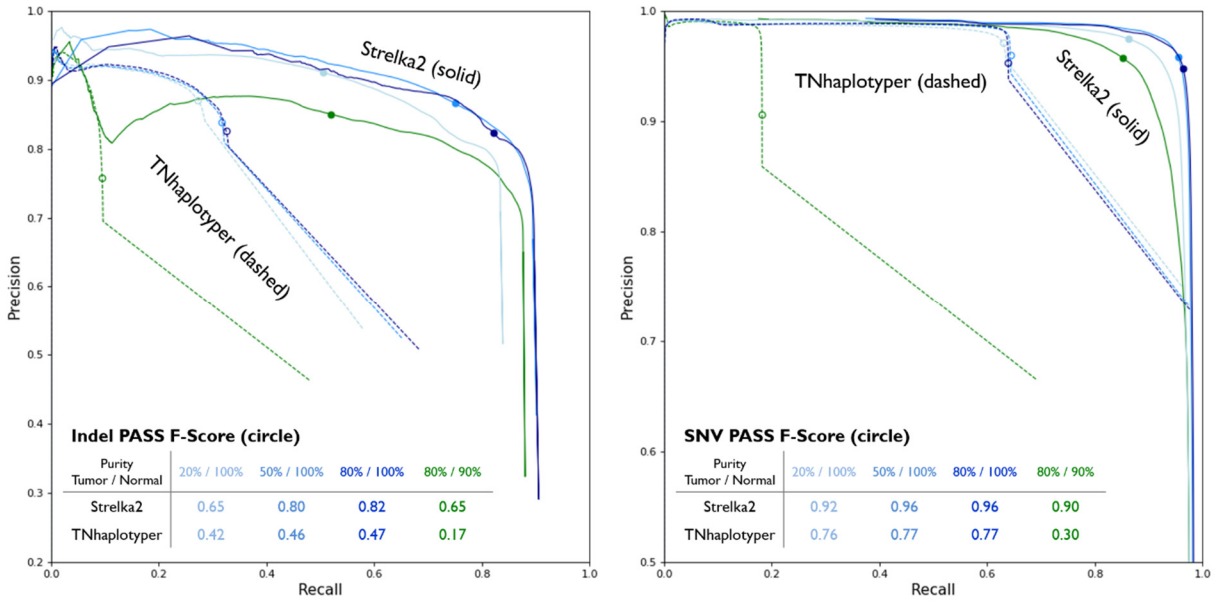


Figure 1. Germline variant calling accuracy and runtime. (a) Germline indel calling accuracy of various pipelines, separately plotted for the Consistency (left) and Truth (right) challenge datasets. Filled and empty circles denote passing calls from Strelka2 and Sentieon DNAseq Haplotyper, respectively. Crosses represent passing calls from PrecisionFDA submissions and stars denote the submissions with the best F-scores. For all 4 datasets from these challenges, we mapped each sample using bwa-mem, and ran Strelka2 on default settings. We then compared results against the latest genome in a bottle truth set[15] using hap.py[16] (see **Supplementary Note 1** for details). **(b)** Runtime for Strelka2 and Haplotyper for the 4 PrecisionFDA datasets measured on the same compute hardware with two Indel Xeon E5-2680 v4 CPUs (total 28 cores). The coverages of the datasets are 40x, 35x, 50x, and 50x for Consistency (Garvan), Consistency (HLI), Truth (HG001), Truth (HG002), respectively.

(a)



(b)

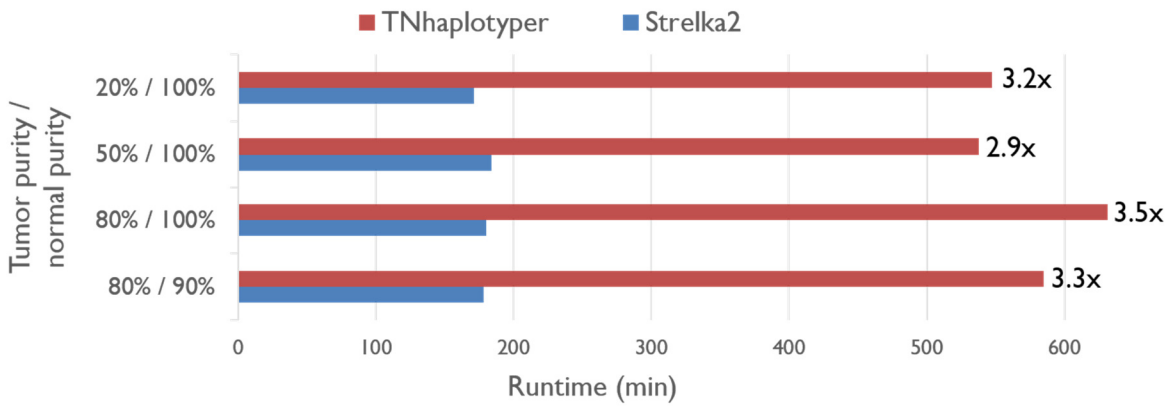
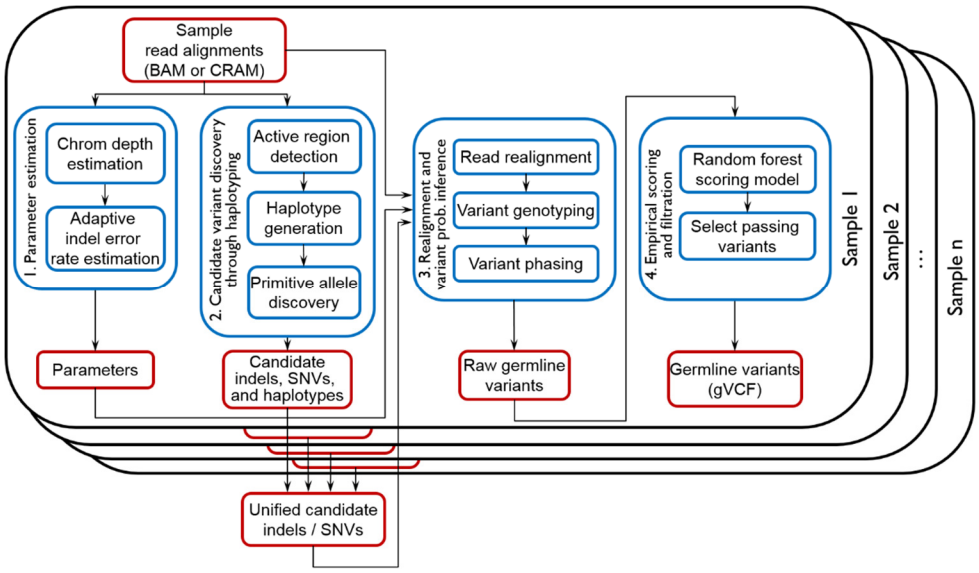


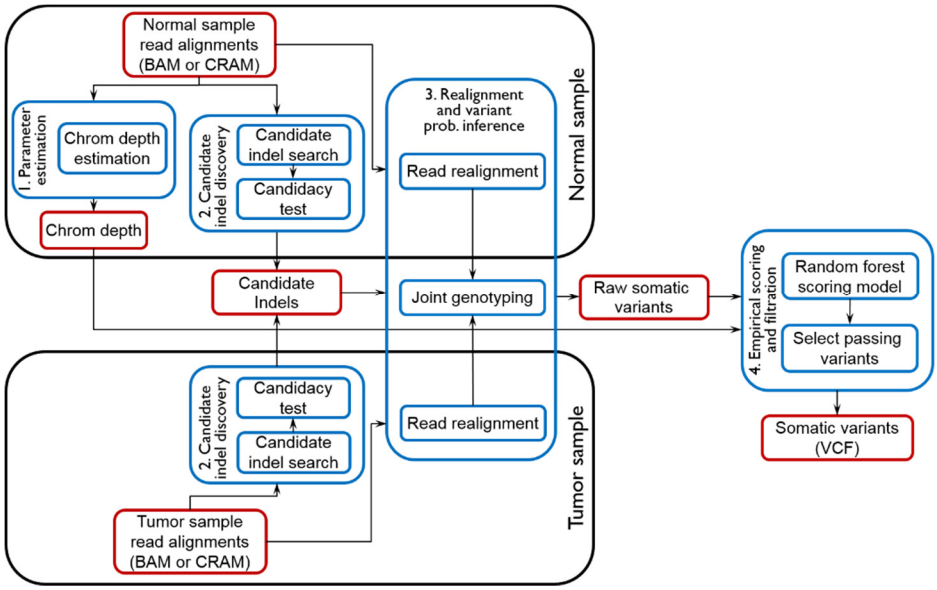
Figure 2. Somatic variant calling accuracy and runtime. (a) Somatic variant calling accuracy for Strelka2 and Sentieon TNseq TNhaplotyper, separately plotted for indels (left) and SNVs (right). Datasets are denoted by x / y, where x and y represent tumor (NA12878) and normal (NA12877) purity, respectively. Filled and empty circles denote the passing calls from Strelka2 and TNhaplotyper, respectively. We mapped each sample using bwa-mem, and ran Strelka2 on default settings. Results were compared against the truth set consisting of the variant calls in NA12878 where the corresponding NA12877 genotype is homozygous reference (see **Supplementary Note 1** for details). **(b)** Runtime for Strelka2 and TNhaplotyper for the 4 admixture datasets, measured on the same compute hardware with two Indel Xeon E5-2680 v4 CPUs (total 28 cores). For each dataset, the coverages of tumor and normal samples are 110x and 37x, resepectively.

Supplementary Figures

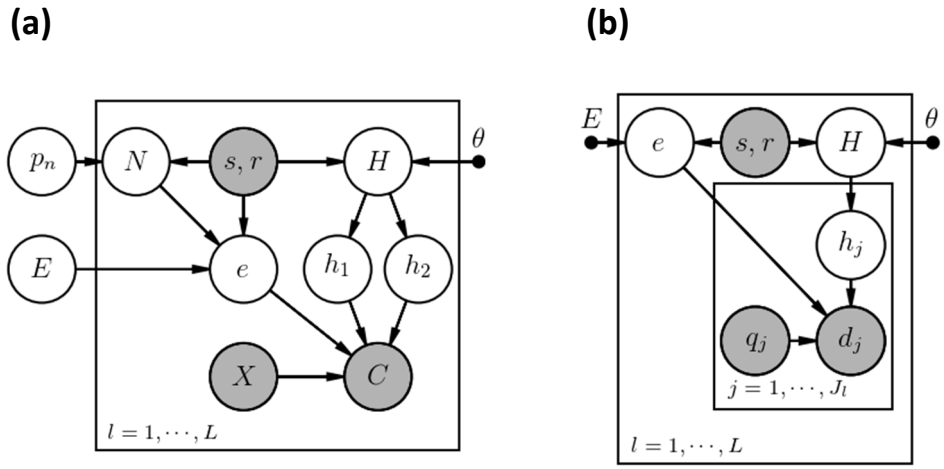
(a)



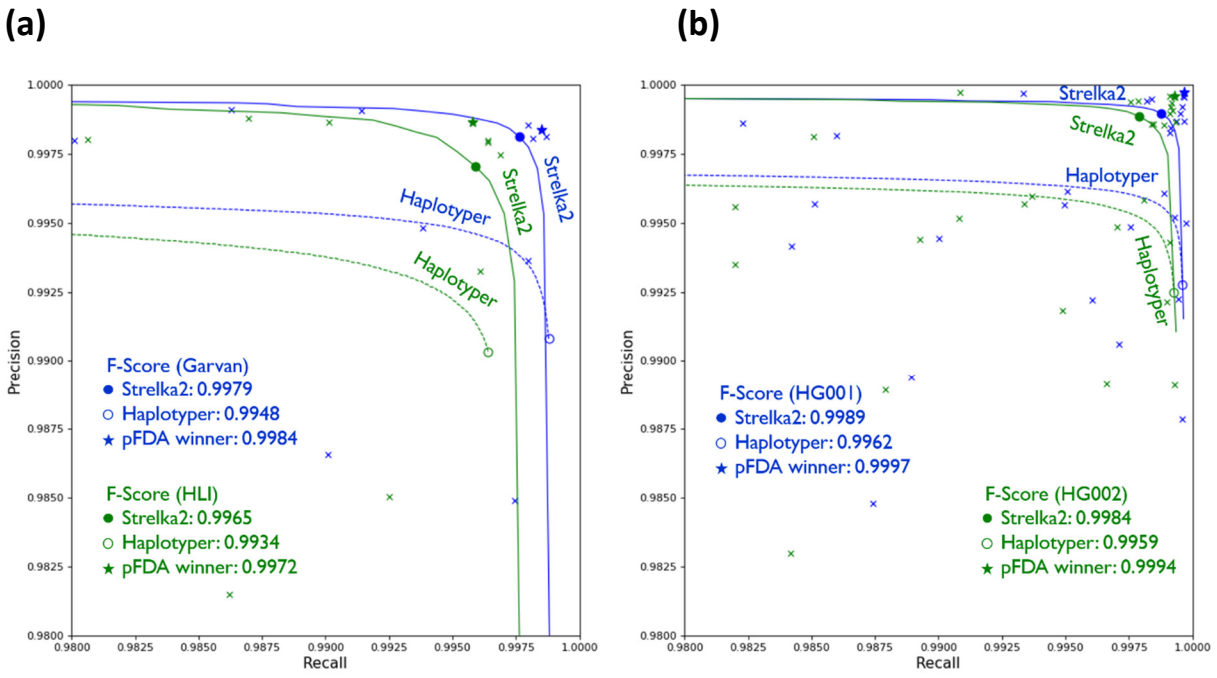
(b)



Supplementary Figure 1. Strelka2 workflows. Strelka2 supports detection of germline variants in small sample cohorts (~10 individuals), and somatic variants from matched tumor-normal sample pairs. These two analyses share several high-level steps, including: (1) parameter estimation, (2) candidate variant discovery, (3) realignment and variant probability inference, and (4) empirical scoring and filtration. Here we diagram an overview of the major workflow components for both **(a)** germline and **(b)** somatic analyses.



Supplementary Figure 2. Structure of the germline indel error and variant calling models in probabilistic graphical model plate notation (some details omitted). **(a) Indel error model.** At each locus *l*, a preliminary estimate of the indel allele count vector *C* is modeled as a mixture binomial distribution governed by the two true haplotypes h_1 and h_2 (a function of the unobserved genotype hypothesis *H*), a set of indel error rates *e* (unobserved) and the total count *X* (observed). The error rates are selected from the full set of error parameters *E* according to the sequence context (summarized as an integer pair denoting the size *s* and number *r* of STR repeats; observed) and a binary state variable *N* (unobserved) categorizing the locus as clean (essentially zero error rates) or noisy (prone to indel errors). The genotype *H* and the noisy-clean state variable *N* are drawn from prior distributions that depend, respectively, on a context-specific mutation rate θ shared across samples and a context-specific noisy-state probability p_n. **(b) Variant calling model.** The reads d_j at every locus are modeled as depending on the corresponding base call quality strings q_j, the unobserved haplotype h_j that generated the read, and the locus-specific error rates *e*. The read-specific haplotype is drawn from the set of haplotypes in the locus-specific hypothesis *H*, of which the prior again depends on a parameter selected from θ according to context. The error rates are again selected from the global vector *E* of error parameters (now treated as fixed), with the difference that all loci analyzed by this model are assumed to be in the noisy state.



Supplementary Figure 3. Germline SNV calling accuracy of various pipelines for the Consistency **(a)** and Truth **(b)** challenge datasets. Filled and empty circles denote the precision and recall of passing calls from Strelka2 and all calls from Sentieon DNAseq Haplotyper, respectively. For Haplotyper, the F-scores of passing calls were far lower than those of all calls, so we chose to plot the F-score for all calls. Crosses represent passing calls from PrecisionFDA submissions and stars denote the submissions with the best F-scores.