

# Notes on Mutect2

David Benjamin,\* Takuto Sato, Lee Lichtenstein, and Megan Shand  
*Broad Institute, 415 Main Street, Cambridge, MA 02142*  
(Dated: February 24, 2021)

## I. MUTECT2

Here we describe the command-line program `Mutect2` itself, which takes us from aligned reads to unfiltered, annotated variant calls. Code block 5 shows how to invoke `Mutect2` using the `gatk` launch script.

```
gatk Mutect2 -R reference.fasta \
  -L intervals.interval_list \
  -I tumor1.bam \
  # Mutect2 may input more tumor samples from the same individual
  [-I tumor2.bam -I tumor3.bam . . .] \
  # Mutect2 may input matched normals from the same individual
  [-I normal1.bam -I normal2.bam . . .] \
  # For most purposes Mutect2 should be supplied with gnomAD
  [-germline-resource af-only-gnomad.vcf] \
  # Mutect2 may input a panel of normals to help identify technical artifacts
  [-pon panel_of_normals.vcf ] \
  # Mutect2 may output orientation bias data
  # for LearnReadOrientationModel
  [--fir2-tar-gz fir2.tar.gz] \
  -O unfiltered.vcf
```

Listing 1: Mutect2 command

### A. Additional modes

The command above encompasses tumor-only, tumor-normal, and multiple-tumor somatic variant calling.

#### 1. Mitochondria mode

For mitochondrial calling one should add the `--mitochondria-mode` flag to the `Mutect2` command line. This switches several defaults from values appropriate to somatic variant calling to values that reflect the greater density of mitochondrial mutations. It also activates annotations relevant to alignment artifacts involving nuclear paralogs to mitochondrial DNA.

#### 2. Force calling

One can force `Mutect2` to assemble and genotype all the variants in `force-calls.vcf` by adding `--genotyping-mode GENOTYPE.GIVEN.ALLELES -alleles force-calls.vcf` to the command line. This injects all alleles in `force-calls.vcf` into the assembly graph, deactivates pruning of subgraphs contained them, and forces `Mutect2` to emit them in the output `vcf` regardless of their evidence. In order to include even filtered alleles in `force-calls.vcf`, use the `--genotype-filtered-alleles` flag. Force-called alleles are emitted *in addition* to any alleles that `Mutect2` would otherwise discover. This so-called GGA mode is useful for studying known driver mutations, for monitoring tumors after chemotherapy, and for validating calls from `Mutect2` or other tools against some orthogonal sequencing data.

---

\*Electronic address: [davidben@broadinstitute.org](mailto:davidben@broadinstitute.org)

## B. Finding Active Regions

**Mutect2** triages sites based on their pileup at a single base locus. If there is sufficient evidence of variation **Mutect2** proceeds with local reassembly and realignment. As in the downstream parts of **Mutect2** we seek a likelihood ratio between the existence and non-existence of an alt allele. Instead of obtaining read likelihoods via Pair-HMM, we assign each base a likelihood. For substitutions we can simply use the base quality. For indels we assign a heuristic effective quality that increases with length.

We use a version of the somatic likelihoods model, described below, with the following modifications:

- We treat every ref base as completely certain. That is,  $\ell_{r,\text{ref}} = 1$  for ref bases.
- We combine all alt alleles and reads supporting them into a single effective alt allele.
- Starting from the same initialization as below, we stop after a single iteration.

Under the first assumption the contribution of ref bases to the sum in Equation 9 vanishes—errorless ref bases contribute no entropy—so we only need to sum over alt bases. By the second assumption the likelihood of an alt read is related to its base quality (and associated error rate  $\epsilon$ ) as  $\ell_{r,\text{alt}} = 1 - \epsilon_r$ .

We compute Equation 9 for two possibilities; first, that an alt allele exists, second that only the ref allele exists. In the former case, Initializing  $\bar{z}_{r,\text{ref}(\text{alt})} = 1$  for ref (alt) bases a single iteration for  $q(\mathbf{f})$  gives  $\beta = (N_{\text{alt}} + 1, N_{\text{ref}} + 1)$ . For alt reads  $r$  we then obtain from Equation 6

$$\bar{z}_{r,\text{alt}} = \frac{\tilde{f}_{\text{alt}}(1 - \epsilon_r)}{\tilde{f}_{\text{alt}}(1 - \epsilon_r) + \tilde{f}_{\text{ref}}\epsilon_r}, \quad (1)$$

where  $\tilde{f}_{\text{ref}(\text{alt})} = \exp(\psi(N_{\text{ref}(\text{alt})} + 1) - \psi(N + 2))$ . We also have  $g(\alpha) = 0$  for a flat prior and  $g(\beta) = \ln(N + 1) + \ln\left(\frac{N}{N_{\text{alt}}}\right)$ . For the ref-only case  $g(\alpha) = g(\beta) = 0$  and Equation 9 reduces to  $\sum_{\text{alt reads}} \ln \epsilon_r$ . Subtracting the variational log likelihoods for the two cases gives

$$\log \text{ odds} \approx -\ln(N + 1) - \ln\left(\frac{N}{N_{\text{alt}}}\right) + \sum_{\text{alt reads}} H(\bar{z}_{r,\text{alt}}) + \bar{z}_{r,\text{alt}} (\ln(1 - \epsilon_r) - \ln \epsilon_r), \quad (2)$$

where  $H(x) = -x \ln x - (1 - x) \ln(1 - x)$  is the Bernoulli entropy.

## C. Local Assembly, Pair-HMM, and Realignment

These topics, which are common to **Mutect2** and **HaplotypeCaller**, are discussed in docs/local\_assembly.pdf, docs/pair\_hmm.pdf, and docs/variants\_from\_haplotypes.pdf in the gatk git repository. As a black box, whenever the evidence in the previous section suffices to trigger local assembly and realignment, we end up at each candidate variant site with one read-vs-allele log likelihood matrix  $\ell$  for each sample, where  $\ell_{ra}$  is the log probability of sequencing read  $r$  given its base qualities and given that read  $r$  is derived from a molecule exhibiting allele  $a$ .<sup>1</sup>

## D. Handling Paired Reads

The fundamental unit of evidence in paired-end sequencing is not the read but the fragment of DNA from which two reads were sequenced. If we apply the somatic likelihoods model below directly to the read-vs-haplotype likelihoods matrix from Pair-HMM our model would in effect allow a read and its mate to come from two different biological haplotypes. To prevent this absurdity, we transform the read-vs-haplotype likelihoods matrix into a fragment-vs-haplotype likelihood matrix. The simplest approach, which **Mutect2** uses, is to multiply the likelihoods of all paired reads in a fragment (in log space, add) to obtain the fragment’s likelihood. That is,

---

<sup>1</sup> Technically, pair-HMM produces a read-vs-haplotype likelihood matrix, which is then “marginalized” to produce a set of read-vs-allele likelihood matrices. In the future, **Mutect2** may operate directly on this read-vs-haplotype matrix in order to exploit the biological fact that there are only a few haplotypes in any region.

$P(\text{fragment}|\text{haplotype}) = P(\text{read 1}|\text{haplotype})P(\text{read 2}|\text{haplotype})$ . Multiplying likelihoods is justified as far as sequencing error is concerned, since sequencing errors on paired reads are statistically independent. There are, of course, shared covariates such as sequence context that influence errors on both reads. These, however, are the domain of **FilterMutectCalls**'s downstream filtering. The somatic likelihoods model of **Mutect2** is *only* concerned with distinguishing sequencing errors (in the narrow sense of an error that occurred on the sequencer itself) from possible somatic variants. **FilterMutectCalls** is responsible for distinguishing somatic variants from all other errors, such as those that occur during sample preparation and alignment.

Simply multiplying likelihoods in this way is not justified when paired reads overlap because while sequencing errors are independent, PCR errors are not. That is, a substitution occurring on both reads may be due to independent sequencing errors or to the amplification of a single PCR error. In order to force the possibility of PCR error into a model of independent reads, we enforce that the effective, multiplied, likelihood must not exceed a bound given by the probability of PCR error. To achieve this, at bases where read pairs overlap, **Mutect2** caps the total (that is, the sum, because qualities are measured in a logarithmic phred scale) base and indel qualities to user-defined PCR SNV and indel qualities. This can be adjusted for PCR-free protocols. For example, if the PCR quality is 40 and two reads overlap with base qualities of 25 and 30, each base quality is replaced with  $40/2 = 20$ . If the base qualities were both, say, 15, no adjustment would be needed because the probability of sequencing error dominates the probability of PCR error. This capping of overlapping base qualities occurs before Pair-HMM, while merging reads into fragments occurs after Pair-HMM.

### E. Somatic Likelihoods Model

We have a set of potential somatic alleles and fragment-allele likelihoods  $\ell_{ra} \equiv P(\text{fragment } r|\text{allele } a)$ . We don't know which alleles are real somatic alleles and so we must compute, for each subset  $\mathbb{A}$  of alleles, the likelihood that the fragments come from  $\mathbb{A}$ . A simple model for this likelihood is as follows: each fragment  $r$  is associated with a latent indicator vector  $\mathbf{z}_r$  with one-hot encoding  $z_{ra} = 1$  iff fragment  $r$  came from allele  $a \in \mathbb{A}$ . The conditional probabilities of fragments given alleles is  $\ell_{ra}$ . There is a latent vector  $\mathbf{f}$  of allele fractions such that  $f_a$  is the allele fraction of allele  $a$ , that is, the prior probability that any given fragment comes from allele  $a$ . Giving  $\mathbf{f}$  a Dirichlet prior, we have a full-model likelihood

$$P(\mathbb{R}, \mathbf{z}, \mathbf{f}|\mathbb{A}) = P(\mathbf{f})P(\mathbf{z}|\mathbf{f})P(\mathbb{R}|\mathbf{z}, \mathbb{A}) = \text{Dir}(\mathbf{f}|\boldsymbol{\alpha}) \prod_a \prod_r (f_a \ell_{ra})^{z_{ra}}. \quad (3)$$

We want to marginalize the latent variables to obtain the evidence  $P(\mathbb{R}|\mathbb{A})$ , which we make tractable via a mean-field approximation  $P(\mathbb{R}, \mathbf{z}, \mathbf{f}|\mathbb{A}) \approx q(\mathbf{z})q(\mathbf{f})$ , which is exact in two limits. First, if there are many fragments, each allele is associated with many fragments and therefore the Law of Large Numbers causes  $\mathbf{f}$  and  $\mathbf{z}$  to become uncorrelated. Second, if the allele assignments of fragments are obvious  $\mathbf{z}_r$  is effectively determinate, hence uncorrelated with  $\mathbf{f}$ . In the variational Bayesian mean-field formalism we have

$$q(\mathbf{f}) \propto \exp E_{q(\mathbf{z})} [\ln P(\mathbb{R}, \mathbf{z}, \mathbf{f}|\mathbb{A})] \propto \text{Dir}(\mathbf{f}|\boldsymbol{\alpha} + \sum_r \bar{\mathbf{z}}_r) \equiv \text{Dir}(\mathbf{f}|\boldsymbol{\beta}), \quad \boldsymbol{\beta} = \boldsymbol{\alpha} + \sum_r \bar{\mathbf{z}}_r \quad (4)$$

$$q(\mathbf{z}_r) \propto \exp E_{q(\mathbf{f})} [\ln P(\mathbb{R}, \mathbf{z}, \mathbf{f}|\mathbb{A})] \propto \prod_a (\tilde{f}_a \ell_{ra})^{z_{ra}}, \quad (5)$$

where, with  $\psi$  denoting the digamma function, the moments

$$\bar{z}_{ra} = E_{q(\mathbf{z})} [z_{ra}] = \frac{\tilde{f}_a \ell_{ra}}{\sum_{a'} \tilde{f}_{a'} \ell_{ra'}} \quad (6)$$

$$\ln \tilde{f}_a = E_{q(\mathbf{f})} [\ln f_a] = \psi(\beta_a) - \psi(\sum_{a'} \beta_{a'}) \quad (7)$$

are easily obtained from the categorical distribution  $q(\mathbf{z})$  and the Dirichlet distribution  $q(\mathbf{f})$ <sup>2</sup>. We initialize  $\bar{z}_{ra} = 1$  if  $a$  is the most likely allele for fragment  $r$ , 0 otherwise and iterate Equations 4 and 5 until convergence. Having obtained the mean fields of  $q(\mathbf{z})$  and  $q(\mathbf{f})$ , we use the variational approximation (Bishop's Eq 10.3) to the model evidence:

$$\ln P(\mathbb{R}|\mathbb{A}) \approx E_q [\ln P(\mathbb{R}, \mathbf{z}, \mathbf{f}|\mathbb{A})] - E_q [\ln q(\mathbf{z})] - E_q [\ln q(\mathbf{f})]. \quad (8)$$

---

<sup>2</sup> Note that we didn't *impose* this in any way. It simply falls out of the mean field equations.

The terms in Eq 8 all involve the standard moments mentioned above, so after a bit of algebraic cancellation we obtain

$$\ln P(\mathbb{R}|\mathbb{A}) \approx g(\boldsymbol{\alpha}) - g(\boldsymbol{\beta}) + \sum_{ra} \bar{z}_{ra} (\ln \ell_{ra} - \ln \bar{z}_{ra}), \quad (9)$$

where we define  $g$  to be the Dirichlet distribution log normalization:

$$\ln \Gamma(\sum_a \omega_a) - \sum_a \ln \Gamma(\omega_a). \quad (10)$$

We now have the model evidence for allele subset  $\mathbb{A}$ . The TLOD emitted by **Mutect2** for an alt allele is the log evidence ratio of an allele set containing all alleles versus an allele set excluding that allele. That is, it is the log odds that an allele exists. When multiple tumor samples are given, **Mutect2** computes a single TLOD by combining all tumor fragments.

## II. FILTERING

The command line tool **FilterMutectCalls** inputs the unfiltered output of **Mutect2** and emits another vcf containing the same variants, annotated with the filters that they fail, if any.

```
gatk FilterMutectCalls -V unfiltered.vcf \
# FilterMutectCalls may input segmentation for one or more
# tumor samples from CalculateContamination
[--tumor-segmentation segments1.table] \
[--tumor-segmentation segments2.table] \
# FilterMutectCalls may input contamination estimates for one or more
# tumor samples from CalculateContamination
[--contamination-table contamination1.table] \
[--contamination-table contamination2.table] \
# orientation bias priors from LearnReadOrientationModel
[--ob-priors priors.tar.gz] \
-O filtered.vcf
```

Listing 2: FilterMutectCalls command

The optional inputs from the GATK 4 tool **CalculateContamination** are described below.

### A. Filtering Architecture

**FilterMutectCalls** contains a set of filters, each of which computes an error probability for each candidate variant. The filters are divided into three categories: **technical artifacts, non-somatic, and sequencing error**. Roughly, we assume that different types of errors within a category are correlated, while different categories are independent. For example, whether sequencing errors cause several bases to be misread is independent of whether the DNA being read came from a contaminating sample and of whether an error during library preparation caused a base error prior to sequencing. To obtain an overall error probability **FilterMutectCalls** computes the maximum within categories and an independent product between categories. That is:

$$P(\text{error}) = 1 - (1 - \max \text{ artifact error prob})(1 - \max \text{ non-somatic prob})(1 - \text{sequencing error prob}). \quad (11)$$

**FilterMutectCalls** goes over an unfiltered vcf in three passes, two to learn any unknown parameters of the filters' models and to set a threshold on  $P(\text{error})$ , and one to apply the learned filters. This section describes methods for determining the threshold on error probability. One option is to simply set a fixed value  $p$  on the error probability, below which a call is considered a real somatic variant. This non-default behavior can be set via `--threshold-strategy CONSTANT --initial-threshold <double>`.

### 1. F Score Thresholding

**FilterMutectCalls** optimizes the F-score—the harmonic mean of recall and precision—as its default thresholding strategy. The `--f-score-beta <double>` command line argument can be set to change the relative weight of recall to precision. In order to optimize the F-score, **FilterMutectCalls** sorts all variants by the probability that they are errors, from least to greatest and calculates the F-score for thresholds in which the first  $n$  variants pass, starting from  $n = 0$  and ending at  $n = N$ , the total number of candidates. This is a cheap  $O(N)$  computation because initially the expected number of true positive and false positive calls are both zero. When the threshold increases to admit a variant with error probability  $p$ , the expected number of true positive calls increases by  $1 - p$  and the expected number of false positive calls increases by  $p$ . The total expected number of real variants is  $\sum_n (1 - p_n)$ . These quantities suffice to calculate recall and precision for every threshold.

### 2. False Discovery Rate Thresholding

**FilterMutectCalls** can also choose to maximize sensitivity subject to a maximum allowable false discovery rate using the `--threshold-strategy FALSE_DISCOVERY_RATE --false-discovery-rate <double>`. For this calculation **FilterMutectCalls** also sorts the error probabilities  $p_n$  from least to greatest. If we allow the first  $M$  variants to pass the expected false discovery rate is

$$\frac{1}{M} \sum_{n=1}^M p_n \quad (12)$$

This is a non-decreasing function of  $M$  because it is the integral from 0 to  $M$  of the monotonic function  $p$ , hence its derivative with respect to  $M$  is  $p_M$ , which is monotonic. Thus it is easy to choose the highest  $M$  such that the maximum false discovery rate is not exceeded.

## B. Hard Filters

Several filters are hard filters that assign an error probability of 1 whenever some annotation exceeds a threshold. Here we summarize all the hard filters of **FilterMutectCalls**, the command line parameters that set their thresholds, and an explanation of the thresholded quantity.

| Filter                          | Threshold  | Explanation  |
|---------------------------------|--|--|
| <code>clustered_events</code>   | <code>max-events-in-region</code>                  | mutations sharing an assembly region                     |
| <code>duplicate_evidence</code> | <code>unique-alt-read-count</code>                 | unique insert start/end pairs of alt reads               |
| <code>multiallelic</code>       | <code>max-alt-alleles-count</code>                 | passing alt alleles at a site                            |
| <code>base_qual</code>          | <code>min-median-base-quality</code>               | median base quality of alt reads                         |
| <code>map_qual</code>           | <code>min-median-mapping-quality</code>            | median mapping quality of alt reads                      |
| <code>fragment</code>           | <code>max-median-fragment-length-difference</code> | difference of alt and ref reads' median fragment lengths |
| <code>position</code>           | <code>min-median-read-position</code>              | median distance of alt mutations from end of read        |
| <code>panel_of_normals</code>   | <code>panel-of-normals</code>                      | presence in panel of normals                             |

## C. Allele Fraction Clustering Model

Several of **FilterMutectCalls**'s probabilistic filters rely on the following model for the tumor's spectrum of allele fractions. At the top of the model is a binary indicator for whether a variant is either a somatic variant or a sequencing error, but not a technical artifact or a non-somatic mutation.

$$P(\text{consider}) = (1 - \text{max artifact error prob})(1 - \text{max non-somatic prob}). \quad (13)$$

Note that possible sequencing errors, with error probability captured by the independent reads assumption of the **Mutect2** log odds, *are* considered. Next, a candidate SNV has prior probability  $\pi_{\text{real}} = \pi_{\text{SNV}}/3$  to be real and prior

probability  $1 - \pi_{\text{SNV}}/3$  to be a sequencing error, while a candidate indel of length  $L$  (negative for deletions, positive for insertions) has prior probability  $\pi_{\text{real}} = \pi_L$  to be real. Real variants are partitioned according to a mixture model with weights  $\pi_k$ , where  $k = 1, 2$  are high-allele fraction and “background” clusters with allele fractions given by beta distributions and  $k > 2$  are clusters with a single discrete allele fraction. The idea of this partitioning is that there will usually be sharply-defined allele fractions corresponding to the hets of different subclones, which are captured by the discrete clusters, but that homozygosity and copy-number variation produce deviations from this clustering.

The prior probability for each possibility in our model is thus

$$P(\text{Sequencing error}) = P(\text{consider})(1 - \pi_{\text{real}}) \quad (14)$$

$$P(\text{Cluster } k) = P(\text{consider})\pi_{\text{real}}\pi_k \quad (15)$$

Mutect2 emits the likelihood ratio between sequencing error and a real variant assuming a flat prior. However, we can approximately compute the likelihood ratio that a non-flat Beta prior would have produced. We can do this because in Equation 9 a change in the prior allele fraction is unlikely to affect  $\bar{z}$  very much—usually the allele that a read supports is clear regardless of prior. Therefore, only the term  $g(\alpha) - g(\beta)$  changes. Since  $\beta = \alpha + \mathbf{n}$ , where  $\mathbf{n}$  is the vector of observed read counts per allele we can convert the log-likelihood obtained with flat prior  $\alpha_0$  to one reflecting an allele fraction prior  $\alpha$  by adding

$$\Delta \log \text{odds}(\alpha) = \Delta \ln P(\mathbb{R}|\mathbb{A}) = [g(\alpha) - g(\alpha + \mathbf{n})] - [g(\alpha_0) - g(\alpha_0 + \mathbf{n})]. \quad (16)$$

Note that for high-allele fraction and background clusters  $\alpha$  is simply the vector  $(\alpha, \beta)$  of Beta distribution shape parameters, while for the binomial clusters we can treat as an arbitrarily sharp beta distribution with identical mean. Since only *ratios* of likelihoods are meaningful, we may WLOG set the likelihood of reads given sequencing error to be  $P(\text{reads}|\text{Sequencing error}) = 1$  and then set the likelihood of each allele fraction cluster to be  $P(\text{reads}|\text{Cluster } k) = \exp[\text{Mutect2 log odds} + \Delta \log \text{odds}(\alpha)]$ . The likelihood of a new cluster is the average over all possible new allele fractions of the likelihood, which is, by definition, simply the original Mutect2 likelihood that integrated over a flat prior. Thus we have likelihoods for every cluster. Combined with the priors, this gives us the conditional posterior for Gibbs sampling the cluster assignment of a candidate variant.

We can estimate the parameters of the model by maximum likelihood via the EM algorithm. For example, we estimate  $\pi_L$  as the total number of length- $L$  indels in the non-error clusters divided by the total number of callable sites<sup>3</sup>,  $\pi_k$  as the total number of variants assigned to the cluster  $k$  divided by the total number of non-error assignments, etc. The allele fraction of a binomial cluster is estimated as its total alt read count divided by its total read count. Finally, the beta distributions of the high-allele fraction and background clusters are fit by brute force likelihood optimization.

To initialize the model, we begin with a background cluster with a flat Beta distribution. Then we iteratively add clusters by the following approach until the Bayes information criterion (BIC) decreases. To add a cluster, we perform an E step and record the responsibilities of each datum to belong to the background cluster. Next we calculate quantiles of allele fraction weighted by this posterior. For example, the 0.1 quantile is the AF below which is contained 0.1 of the total responsibility for the background cluster. Then, weighted by these background responsibilities, we add up the posterior probabilities for each variant to belong to each allele fraction quantile under a flat prior assumption. The next peak we try to add is the quantile that maximizes this sum. We perform several EM iterations before calculating the BIC.

Once the model converges, the combination of cluster weights and allele-fraction-adjusted likelihoods lets us compute the posterior probability that a candidate variant is a sequencing error, which determines the **weak\_evidence** filter. Since the clusters also define distributions on alt allele counts—the high-AF and background clusters define beta-binomials and the other clusters define binomials—we also have the mixture model distribution  $P(\text{alt count}|\text{total count}, \text{somatic})$ , which the contamination and germline filters use.

#### D. Strand Artifact Model

The strand artifact filter detects sequencing artifacts in which the evidence for the alt allele consists entirely of forward strand reads alone or reverse strand reads alone. We must detect this while taking into account the fact that at some loci, such as near the end of an exome target, *all* reads are biased towards one direction, and therefore a bias towards a particular strand among alt reads is no cause for alarm.

---

<sup>3</sup> This is emitted by Mutect2 and passed to FilterMutectCalls.

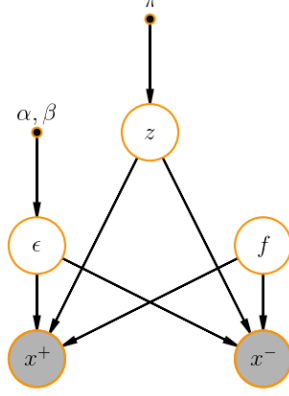


FIG. 1: The probabilistic graphical model for the strand artifact model

Let  $z \in \{z_+, z_-, z_o\}$  be a latent random variable with 1-hot encoding that represents the artifact state of a suspected variant i.e.  $z_+ = 1$  for a forward strand artifact,  $z_- = 1$  for a reverse artifact, and  $z_o = 1$  otherwise. At each locus, let  $x_{\pm}$  be the number of forward (+) or reverse (-) strand alt reads and let  $n_{\pm}$  be the total depth for each strand. By modeling  $x_{\pm}$  relative to  $n_{\pm}$  we account for inherent strand bias due to, for example, reads falling at the end of an exome target and do not confuse it for an artifact. Let  $f$  be the allele fraction of true variation in case  $z_o = 1$ . Let  $\epsilon$  be the strand bias error rate and let  $\theta$  be the non-strand-biased error rate. We will ignore the case in which significant strand bias coincides with real variation, first because this is exceedingly rare and ignoring it has a negligible effect on the parameters of our model, and secondly because such variants should be considered true positives.

The conditional distributions of our model are binomial

$$x_+ | \epsilon, \theta, f, z_+ = 1 \sim \text{Bin}(x_+ | n_+, \epsilon) \quad (17)$$

$$x_+ | \epsilon, \theta, f, z_- = 1 \sim \text{Bin}(x_+ | n_+, \theta) \quad (18)$$

$$x_+ | \epsilon, \theta, f, z_o = 1 \sim \text{Bin}(x_+ | n_+, f), \quad (19)$$

and similarly for  $x_-$ . Putting beta priors on  $\epsilon$ ,  $\theta$ , and  $f$ , with parameters  $(\alpha_{\epsilon}, \beta_{\epsilon})$ ,  $(\alpha_{\theta}, \beta_{\theta})$ , and  $(\alpha_f, \beta_f)$  and marginalizing latent parameters we obtain likelihoods

$$P(x_+, x_- | z_{\pm} = 1) = \text{BetaBinom}(x_{\pm} | n_{\pm}, \alpha_{\epsilon}, \beta_{\epsilon}) \text{BetaBinom}(x_{\mp} | n_{\mp}, \alpha_{\theta}, \beta_{\theta}) \quad (20)$$

$$P(x_+, x_- | z_o = 1) = \int_0^1 \text{Beta}(f | \alpha_f, \beta_f) \text{Binom}(x_+ | n_+, f) \text{Binom}(x_- | n_-, f) df \quad (21)$$

$$= \frac{\binom{n_+}{x_+} \binom{n_-}{x_-}}{\binom{n_+ + n_-}{x_+ + x_-}} \text{BetaBinom}(x_+ + x_- | n_+ + n_-, \alpha_f, \beta_f) \quad (22)$$

Finally, we let  $\pi/2$  be the prior probability of a forward or reverse strand artifact. From the above equations it is straightforward to calculate the posterior probability of  $z$  and to learn  $\pi$  iteratively via the EM algorithm. It is somewhat more complicated to learn  $(\alpha_{\epsilon}, \beta_{\epsilon})$  and  $(\alpha_{\theta}, \beta_{\theta})$ , so we treat these as fixed hyperparameters. We use a flat prior  $\alpha_f = \beta_f = 1$  for the true allele fraction.

### E. Germline Filter

Suppose we have detected an allele such that its (somatic) likelihood in the tumor is  $\ell_t$  and its (diploid) likelihood in the normal is  $\ell_n^4$ . By convention, both of these are relative to a likelihood of 1 for the allele *not* to be found. If

<sup>4</sup> This is the total likelihood for het and hom alt in the normal.

we have no matched normal,  $\ell_n = 1$ . Suppose we also have the population allele frequency  $f$  of this allele. Then the prior probabilities for the normal to be heterozygous and homozygous alt for the allele are  $2f(1-f)$  and  $f^2$  and the prior probability for the normal genotype not to contain the allele is  $(1-f)^2$ . Finally, let the prior for this allele to arise as a somatic variant be  $\pi$ .

If the variant exists in the tumor as a real somatic variant it has likelihood  $\ell_t^S$ , the adjusted **Mutect2** likelihood accounting for allele fraction clustering. If it exists as a germline het in a segment with minor allele fraction  $m$  the likelihood is  $\ell_t(m)$  for the alt minor case and  $\ell_t(1-m)$  for the alt major case, after adjusted the **Mutect2** likelihood to account for pinning the allele fraction to  $m$  or  $1-m$ . We need the unnormalized probabilities of three possibilities:

1. The variant exists in the tumor and the normal as a germline het. This has unnormalized probability  $f(1-f)\ell_n(1-\pi)(\ell_t(m) + \ell_t(1-m))$ .
2. The variant exists in the tumor and the normal as a germline hom alt. This has unnormalized probability  $f^2\ell_n\ell_t(1)(1-\pi)$ .
3. The variant exists in the tumor but not the normal. This has unnormalized probability  $(1-f)^2\ell_t^S\pi$ .

We exclude possibilities in which the variant does not exist in the tumor sample because we really want the conditional probability that the variant is germline given that it would otherwise be called. The normalized sum of the first two possibilities is the germline error probability.

So far we have assumed that the population allele frequency  $f$  is known, which is the case if it is found in our germline resource, such as gnomAD. If  $f$  is not known we must make a reasonable guess as follows. Suppose the prior distribution on  $f$  is  $\text{Beta}(\alpha, \beta)$ . The mean  $\alpha/(\alpha+\beta)$  of this prior is the average human heterozygosity  $\theta \approx 10^{-3}$ , so we have  $\beta \approx \alpha/\theta$ . We need one more constraint to determine  $\alpha$  and  $\beta$ , and since we are concerned with imputing  $f$  when  $f$  is small we use a condition based on rare variants. Specifically, the number of variant alleles  $n$  at some site in a germline resource with  $N/2$  samples, hence  $N$  chromosomes, is given by  $f \sim \text{Beta}(\alpha, \beta), n \sim \text{Binom}(N, f)$ . That is,  $n \sim \text{BetaBinom}(\alpha, \beta, N)$ . The probability of a site being non-variant in every sample is then  $P(n=0) = \text{BetaBinom}(0|\alpha, \beta, N)$ , which we equate to the empirical proportion of non-variant sites in our resource, about 7/8 for exonic sites in gnomAD. Solving, we obtain approximately  $\alpha = 0.01, \beta = 10$  for gnomAD. Now, given that some allele found by **Mutect2** is not in the resource, the posterior on  $f$  is  $\text{Beta}(\alpha, \beta + N)$ , the mean of which is, since  $\beta \ll N$ , about  $\alpha/N$ . By default, **Mutect2** uses this value.

## F. Contamination Filter

Suppose our tumor sample has contamination fraction  $\alpha$  and that at some site we have  $a$  alt reads out of  $d$  total reads. Suppose further that the alt allele has population allele frequency  $f$ . We estimate the probability that these alt reads came from a contaminating sample and not from a true somatic variant. Let  $\pi$  be the prior probability of a somatic variation from the allele fraction clustering model. Then alt read count likelihood  $P(a|d, \text{somatic})$  is also given by this model. The possibility that the reads are due to contamination has prior  $1-\pi$  and we take the maximum likelihood among two models of contamination.

If there are multiple contaminants we approximate each contaminant read as independent so that

$$P(a|d, \text{many contaminant}) = \text{Binom}(a|d, \alpha f). \quad (23)$$

If there is a single contaminating sample it is heterozygous with probability  $2f(1-f)$  and homozygous for the alt with probability  $f^2$ , in which cases fractions  $\alpha/2$  and  $\alpha$  of all reads to be alt contaminants. The contaminant is homozygous for the ref with probability  $(1-f)^2$ , which yields no alt reads. Thus

$$P(a|d, \text{one contaminant}) = 2f(1-f)\text{Binom}(a|d, \alpha/2) + f^2\text{Binom}(a|d, \alpha) + (1-f)^2\mathbb{I}[a=0]. \quad (24)$$

Taking the maximum of these as the likelihood given contamination, we can easily compute the probability of contamination error.

## G. Read Orientation Artifact Filter

The read orientation model is implemented in `LearnReadOrientationModel`, which produces the `--orientation-bias-artifact-priors` input to `FilterMutectCalls`. The raw counts for `codeLearnReadOrientationModel` are generated either by `CollectF1R2Counts` or **Mutect2** itself.



```

gatk Mutect2 -R reference.fasta \
-I tumor.bam \
-O unfiltered.vcf \
[. . . any other arguments] \
--f1r2-tar-gz f1r2.tar.gz

gatk LearnReadOrientationModel \
-I f1r2.tar.gz
-O tumor-artifact-prior.tar.gz

```

Listing 3: LearnReadOrientationModel command

The output `tumor-artifact-prior.tar.gz` is then passed as the parameter to `FilterMutectCalls` with the `--orientation-bias-artifact-priors` argument. Note that all `tar.gz` files shown above contain information for every tumor sample input to `Mutect2`.

The read orientation artifact, also known as the orientation bias artifact, arises due to a chemical change in the nucleotide during library prep that results in, for example, G base-pairing with A. This kind of artifact has a clear signature (e.g. C to A SNP that occurs predominantly for the middle C in the DNA sequence CCG), and it's single-stranded in nature. Downstream, this artifact manifests as low allele fraction SNPs whose evidence for the alt allele consists almost entirely F1R2 reads or F2R1 reads. A read pair is F1R2 (forward 1st, reverse 2nd) if the sequence of bases in Read 1 maps to the forward strand of the reference (F1), and the sequence of Read 2 to the reverse strand of the reference (R2). F2R1 is defined similarly.

Without loss of generality, suppose that the reference context at locus  $i$  is ACT. Let  $\mathbf{z}_i$  denote the genotype at locus  $i$  with the one-hot encoding  $z_{ik} = 1$  iff the genotype of locus  $i$  is  $k$ , where the possible genotypes are

$$\mathbf{z}_i \in \{\text{F1R2}_A, \text{F1R2}_G, \text{F1R2}_T, \text{F2R1}_A, \text{F2R1}_G, \text{F2R1}_T, \text{Hom Ref}, \text{Germline Het}, \text{Somatic Het}, \text{Hom Var}\}$$

$z_i = \text{F1R2}_A$  denotes that at locus  $i$  we have an artifact in which the evidence for alt allele A consists entirely of reads in the F1R2 orientation. The remaining artifact states are defined analogously. Let  $\pi$  denote the prior probabilities of the  $\mathbf{z}_i$  under the reference context ACT. Then we have

$$P(\mathbf{z}_i) = \prod_k \pi_k^{z_{ik}} \quad (25)$$

The number of alt reads at a locus depends on the genotype  $\mathbf{z}_i$ . Let  $n_i$  and  $m_i$  denote the total depth and alt depth at locus  $i$ , respectively. The conditional distribution of  $m_i$  is

$$P(m_i | z_{ik} = 1) = \text{BetaBinomial}(m_i | n_i, \alpha_k, \beta_k) \quad (26)$$

where  $\alpha_k$  and  $\beta_k$  are fixed hyperparameters for genotype  $z_k$ . When the site's genotype indicates in  $m_i$  alt reads we expect a heavily skewed distribution of F1R2 reads. This is captured in the conditional distribution of F1R2 alt reads. Let  $c_i$  denote the number of F1R2 reads among the  $m_i$  alt reads at locus  $i$ . Then we have

$$P(c_i | m_i, z_{ik} = 1) = \text{BetaBinomial}(c_i | m_i, \alpha'_k, \beta'_k) \quad (27)$$

We learn the prior artifact probabilities  $\pi$  based on the observed values of  $n_i$ ,  $m_i$ ,  $c_i$  for each of  $N$  loci using the EM algorithm. In the E-step, we compute the posterior probabilities of  $\mathbf{z}_i$  for  $i = 1 \dots N$ . The joint probabilities of  $\mathbf{z}$  factorizes over  $i$ , thus the posteriors over  $\mathbf{z}$  are independent across loci.

$$P(z_{ik} = 1 | m_i, c_i) \propto P(z_{ik} = 1, m_i, c_i) = \pi_k \text{BetaBinomial}(m_i | n_i, \alpha_k, \beta_k) \text{BetaBinomial}(c_i | m_i, \alpha'_k, \beta'_k) \quad (28)$$

In the M-step we maximize the expectation of the log complete-data likelihood with respect to  $\pi$ . The log complete data likelihood is given as

$$\ln P(\mathbf{z}, \mathbf{m}, \mathbf{c}) = \sum_i \sum_k z_{ik} (\ln \pi_k + \ln \text{BetaBinomial}(m_i | n_i, \alpha_k, \beta_k) + \ln \text{BetaBinomial}(c_i | m_i, \alpha'_k, \beta'_k)) \quad (29)$$

Maximizing the log likelihood under the constraint  $\sum_k \pi_k = 1$  gives us

$$\pi_k = \frac{N_k}{N} \quad (30)$$

where  $N_k = \sum_i P(z_{ik} | m_i, c_i)$  is the effective count of loci with genotype  $k$ . We alternate E-step and M-step until convergence. We then use the learned prior genotype probabilities to compute the posterior artifact probabilities of variants in a vcf. The filtering threshold is set such that the false discovery rate doesn't exceed a specified value, as described below.

## H. Polymerase Slippage

For indels in short tandem repeats (STRs) `FilterMutectCalls` uses a simple model for the possibility that alt reads are due to polymerase slippage. The prior  $\pi_L$  for a real variant of length  $L$  comes from the allele fraction clustering model. `FilterMutectCalls` assumes that polymerase slippage only occurs in STRs of 8 bases or more and only results in insertions or deletions of a single repeat unit. The likelihood of  $a$  alt reads out of  $d$  total reads in the case of a real somatic variant is given by the allele fraction clustering model. The likelihood in the case of polymerase slippage is the marginal of binomial likelihoods over a slippage rate with a uniform prior from 0 to 0.1, which is a regularized Beta function. Given priors and likelihoods, the error probability follows.

## I. Normal Artifacts

A matched normal is useful not only for detecting germline variants but also for distinguishing technical artifacts for somatic mutations. Even a large panel of normals may not include, for example, mapping errors due to a rare SNP in a centromere, while a matched normal will exhibit such errors. `Mutect2` emits a normal artifact log odds annotation by applying the somatic likelihoods model to the normal. We combine this likelihood of the reads given that an artifact appears in the normal with a prior probability for normal artifacts to obtain a posterior error probability. Since this prior is unknown, we use the rate of technical artifacts in the tumor detected by `FilterMutectCalls` as a proxy.

## J. Bad Haplotypes

`Mutect2` emits phasing information for calls in the same assembly region. We assign a “bad haplotype” probability equal to the greatest technical artifact probability of any in-phas call within a certain distance, by default 100 bases.

## III. RELATED GATK TOOLS

The Broad somatics SNVs and indels pipeline involves several GATK tools besides `Mutect2` and `FilterMutectCalls`. We describe them here.

### A. Creating a Panel of Normals

Currently, a panel of normals is simply a vcf of blacklisted sites flagged as recurrent artifacts<sup>5</sup>. One generates it by running `Mutect2` in tumor-only mode on a large number of normal samples, then running the following workflow:

```
gatk GenomicsDBImport -R reference.fasta -L intervals.interval_list \
  --genomicsdb-workspace-path pon_db \
  -V normal1.vcf \
  -V normal2.vcf \
  -V normal3.vcf . . .

gatk CreateSomaticPanelOfNormals -R reference.fasta -V gendb://pon_db -O pon.vcf
```

Listing 4: `CreateSomaticPanelOfNormals` command

### B. Calculating Contamination

To calculate the cross-sample calculation of a tumor sample, one runs the GATK tools `GetPileupSummaries` and `CalculateContamination` as follows:

---

<sup>5</sup> `CreateSomaticPanelOfNormals` emits more information than this, but `Mutect2` does not yet use it.

```

gatk GetPileupSummaries -I tumor.bam \
  -V common-biallelic.vcf \
  -L common-biallelic.vcf \
  -O tumor.pileups

# if a normal is present, it is helpful to obtain its pileup summaries
gatk GetPileupSummaries -I normal.bam \
  -V common-biallelic.vcf \
  -L common-biallelic.vcf \
  -O tumor.pileups

gatk CalculateContamination -I tumor.pileups \
  # the normal pileups are useful but optional
  [-matched normal.pileups] \
  -O contamination.table \
  # it is highly recommended to produce segments for FilterMutectCalls
  [-tumor-segmentation segments.table]

```

Listing 5: CalculateContamination command

**CalculateContamination** is the GATK’s fast, simple, and accurate method for calculating the contamination of a sample. This method does not require a matched normal, makes no assumptions about the number of contaminating samples, and remains accurate even when the sample has a lot of copy number variation.

The inputs are a bam file and a vcf of common variants—for example ExAC, gnomAD, or 1000 Genomes—with their allele frequencies. The basic idea, which comes from ContEst<sup>6</sup> by Kristian Cibulskis and others in the Broad Institute Cancer Genome Analysis group, is simply to count ref reads at hom alt sites and subtract the number of ref reads expected from sequencing error to obtain the number of ref reads contaminating these hom alt sites. Finally, we use the allele frequencies to account for the fact that some contaminating reads have the alt allele. The only subtlety is in distinguishing hom alt sites from loss of heterozygosity events, which we describe below.

Suppose we have a set  $\mathbb{H}$  of SNPs at which our sample is homozygous for the alternate allele. Let  $N_{\text{ref}}$  be the total number of ref reads at these sites. We can decompose  $N_{\text{ref}}$  as follows:

$$N_{\text{ref}} = N_{\text{ref}}^{\text{error}} + N_{\text{ref}}^{\text{contamination}}, \quad (31)$$

where  $N_{\text{ref}}^{\text{error}}$  and  $N_{\text{ref}}^{\text{contamination}}$  are as the number of ref reads due to error and contamination, respectively. We can obtain  $N_{\text{ref}}$  by counting reads, and we estimate  $N_{\text{ref}}^{\text{error}}$  as follows. Suppose, WLOG, that the ref allele is A and the alt is C. Then, assuming that all substitution errors are equally likely,  $N_{\text{ref}}^{\text{error}}$  is approximately half the number of Gs and Ts. This is, of course, not a perfect assumption for any one site, but on average over all the sites in  $\mathbb{H}$  it is very good.

Next we take the expectation of both sides of Equation 31 to obtain

$$\langle N_{\text{ref}} - N_{\text{ref}}^{\text{error}} \rangle = \left\langle \sum_{s \in \mathbb{H}} \text{number of contaminant ref reads at } s \right\rangle \quad (32)$$

$$= \sum_{s \in \mathbb{H}} \langle \text{number of contaminant ref reads at } s \rangle \quad (33)$$

$$= \sum_{s \in \mathbb{H}} \langle \text{number of contaminant reads at } s \times \text{ref fraction of contaminant reads at } s \rangle \quad (34)$$

$$= \sum_{s \in \mathbb{H}} \langle \text{number of contaminant reads at } s \rangle \times \langle \text{ref fraction of contaminant reads at } s \rangle \quad (35)$$

where we have used the linearity of the expectation and the independence of the total number of contaminant reads with the fraction of contaminant reads that are ref. The expectation of the total number of contaminant reads is the depth  $d_s$  at site  $s$  times the contamination, which we denote by  $\chi$ . The expected fraction of contaminant reads that

---

<sup>6</sup> ContEst: estimating cross-contamination of human samples in next-generation sequencing data, *Bioinformatics* **27**, 2601 (2011)

are ref is one minus the alt allele frequency  $f_s$ . Crucially, this fact is independent of how many contaminating samples there are. Thus we have

$$\langle N_{\text{ref}} - N_{\text{ref}}^{\text{error}} \rangle = \chi \sum_{s \in \mathbb{H}} d_s (1 - f_s) \quad (36)$$

and obtain the estimate

$$\hat{\chi} \approx \frac{N_{\text{ref}} - N_{\text{ref}}^{\text{error}}}{\sum_{s \in \mathbb{H}} d_s (1 - f_s)} \quad (37)$$

Let us now roughly estimate the error bars on this result. The main source of randomness is the stochasticity in the number of contaminating ref reads. Although the nature of this randomness depends on the number of contaminants, the most variable case, hence an upper bound, is that of a single haploid contaminant, since at each site the only possibilities are the extremes of all contaminant reads being ref or all being alt. In this case, the contribution to the numerator of Eq. 37 from site  $s$  is the random variable  $X_s Z_s$ , where  $X_s \sim \text{Binom}(d_s, \chi)$  is the number of contaminant reads at  $s$  and  $Z_s$  is a binary indicator for whether the contaminant reads are ref, with  $P(Z_s = 1) = 1 - f_s$ .  $X$  and  $Z$  are independent, so we can work out the variance of  $XZ$  as:

$$\text{var}(XZ) = E[X^2 Z^2] - E[XZ]^2 \quad (38)$$

$$= (1 - f_s) E[X^2] - (1 - f_s)^2 E[X]^2 \quad (39)$$

$$= (1 - f_s) (\text{var}(X) + E[X]^2) - (1 - f_s)^2 E[X]^2 \quad (40)$$

$$= (1 - f_s) d_s \chi (1 - \chi) + f_s (1 - f_s) d_s^2 \chi^2 \quad (41)$$

And therefore the standard error on  $\hat{\chi}$  comes out to the square root of the sum of these per-site variances, divided by the denominator of Eq. 37, that is,

$$\text{std}(\chi) = \frac{\sqrt{\sum_s [(1 - f_s) d_s \chi (1 - \chi) + f_s (1 - f_s) d_s^2 \chi^2]}}{\sum_s d_s (1 - f_s)} \quad (42)$$

However,  $\chi$  is not the same as its estimator  $\hat{\chi}$ . We can define the maximum of a confidence interval as the largest  $\chi$  such  $\chi - \text{std}(\chi) < \hat{\chi}$ . We can define the minimum similarly. Note that in the case where contamination is small and we estimate  $\hat{\chi} = 0$  the variance grows linearly with  $\chi$  for small  $\chi$ , hence the standard deviation grows faster than linear

It remains to describe how we determine which sites are hom alt. The fundamental challenge here is that in cancer samples loss of heterozygosity may cause het sites to look like hom alt sites. Our strategy is to partition the genome into allelic copy-number segments, then infer the minor allele fraction of those segments. We segment the genome just as in GATK CNV, using a kernel segmenter with a Gaussian kernel computed on the alt fraction. A nonlinear kernel is important because each segment is multimodal, with peaks for hom ref, alt minor het, alt major het, and hom alt.

We then perform maximum likelihood estimation MLE on a model with learned parameters  $\mu$ , the local minor allele fraction for each segment,  $\chi$ , the contamination, and a constant base error rate parameter  $\epsilon$  determined by counting reads that are neither the ref nor primary alt base at biallelic SNPs as described above. The model likelihood is

$$P(\{a\}|\{f\}, \chi, \{d\}) = \prod_{\text{segments}} \prod_{n \text{ sites}} \sum_{s \text{ genotype } g} P(g|f_s) \text{Binom}(a_s|d_s, (1 - \chi)\phi(g, \mu_n, \epsilon) + \chi f_s) \quad (43)$$

where  $a_s$  and  $d_s$  are the alt and total read counts at site  $s$ , allelic CNV genotypes  $g$  run over hom ref, alt minor, alt major, and hom alt with priors  $P(\text{hom ref}) = (1 - f_s)^2$ ,  $P(\text{alt minor}) = P(\text{alt major}) = f_s(1 - f_s)$ , and  $P(\text{homalt}) = f_s^2$ .  $\phi(g, \mu, \epsilon)$  is the alt allele fraction of the uncontaminated sample:  $\phi(\text{hom ref}) = \epsilon$ ,  $\phi(\text{alt minor}) = \mu$ ,  $\phi(\text{alt major}) = 1 - \mu$ ,  $\phi(\text{hom alt}) = 1 - \epsilon$ . The binomial is the weighted average of the uncontaminated sample and sample reads drawn independently from allele frequency  $f$ . This is inconsistent with a single diploid contaminant sample, or indeed with any finite number of contaminants, which is why we do not the the MLE estimate in the final output of the tool. The model also assumes that the uncontaminated and contaminating samples have the same overall depth distribution at each site, which is inconsistent with any differences in copy-number. We perform the MLE by brute force, alternately maximizing with respect to  $\chi$  with  $\mu$  fixed and vice versa. In order to make the solution more robust, we exclude segments with low  $\mu$  from the maximization over  $\chi$  by taking the highest possible threshold (up to 0.5, of course) for  $\mu$  that retains at least 1/4 of all sites.

Once we have learned the parameters of this model, we can easily infer the posterior probabilities of hom alt genotypes. We take every site with a posterior probability greater than 0.5. In order to make the result more reliable

against CNVs, we again impose a threshold on segment minor allele fraction and apply the above formula only to hom alt sites in these segments. This time, however, we choose the highest possible threshold such that the estimated relative error is less than 0.2.

Finally, we note that the same calculation can be reversed by using alt reads in hom ref sites as the signal and replacing  $f$  by  $1 - f$  everywhere above. We use the estimate from hom refs as a backup when the hom alt estimate has too great an error, as can occur in the case of targeted panels with few sites. We do not use this as our primary estimate because it is much more affected by uncertainty in the population allele frequencies and is thus susceptible to systematic bias.

### C. Filtering Alignment Artifacts

`FilterMutectCalls` does a fairly good job detecting all sorts of errors, including mapping artifacts. When more precision is needed and a slight loss of sensitivity is acceptable, one can run the output of `FilterMutectCalls` through `FilterAlignmentArtifacts`. This tool assembles alt-supporting reads together with their mates using the GATK assembly engine in order to generate a local alt allele haplotype. It then maps this alt haplotype to the best reference available<sup>7</sup> using an embedded BWA-mem with parameters optimized for detecting multimapping. The tool requires a BWA mem reference image index file, which is available from the GATK resource bucket. The first step below, in which this image is generated, is usually not necessary.

```
gatk BwaMemIndexImageCreator -I reference.fasta -O reference.fasta.img

gatk FilterAlignmentArtifacts \
  # the reference argument is the bam's original alignment
  -R original-reference.fasta \
  # filtered calls from FilterMutectCalls
  -V filtered.vcf \
  -I tumor.bam \
  --bwa-mem-index-image reference.fasta.img \
  -O realignment-filtered.vcf
```

Listing 6: `FilterAlignmentArtifacts` command

### D. Proposed tumor in normal estimation tool

Note: the following notes are just a proposal for which no GATK tool yet exists. A popular tool is DeTiN<sup>8</sup> by Amaro Taylor-Weiner and others at the Broad Institute Cancer Genome Analysis group.

Similar to the spirit of `CalculateContamination`, the fraction of tumor reads in the normal bam is a single number with a large amount of evidence and is probably well-estimated by simple descriptive statistics rather than a full-fledged probabilistic model. It shouldn't be much more complicated than finding somatic variants and comparing their signal in the normal sample to that in the tumor.

We propose the following steps to obtain our input of confident somatic SNVs:

- Run `Mutect2` with the `--genotype-germline-sites` argument to obtain a preliminary list of somatic SNVs, including those that look like germline variants due to tumor-in-normal contamination. For the sake of speed, we could implement a pileup-based mode in which we skip reassembly and equate read likelihoods with base qualities. This would allow us to obtain variant annotations using the existing architecture of `Mutect2` and therefore to filter calls with no new code.
- Run `FilterMutectCalls`. With default settings this eliminates the great majority of sequencing artifacts. To eliminate even more we could increase the log odds threshold slightly, essentially requiring a slightly larger alt allele count. Normally we don't do this because it sacrifices some sensitivity, but for our purposes here a 10 or 20 percent loss of sensitivity is perfectly acceptable as long as we are left with enough SNVs for our estimate.

<sup>7</sup> That is, if the reads come from an hg19-aligned bam, they will still be realigned to hg38 if that is provided.

<sup>8</sup> DeTiN: overcoming tumor-in-normal contamination, *Nature Methods* **15**, 531 (2018)

It would also make sense to be especially stringent with variants that have any significant population allele frequency in gnomAD, as well as possible mapping errors. Thus we might also run `FilterAlignmentArtifacts` with strict parameters.

- Reconsider all variants that are filtered only by the germline and/or normal artifact filters. Those that have enough read counts in the normal that we conclude they are germline variants, as opposed to tumor in normal contamination, should remain filtered.

The above steps are all very reliable, so at this point we can assume we have a collection of confident biallelic somatic SNVs that are hom ref in the germline. Similar to `CalculateContamination`, we can now estimate the number of alt reads in the normal at these sites:

$$\text{alt in normal} \approx \sum_{\text{sites}} (\text{depth in normal}) \times (\text{alt fraction in tumor}) \times (\text{tumor in normal fraction}) \quad (44)$$

Hence we estimate

$$\text{tumor in normal fraction} \approx \frac{\text{total number of alt reads in normal at somatic SNV sites}}{\sum_{\text{somatic SNV sites}} (\text{depth in normal}) \times (\text{alt fraction in tumor})} \quad (45)$$

We could also iterate this process as in `CalculateContamination`: first get an initial guess of the tumor-in-normal fraction, then use the initial guess to improve our “un-filtering” in the third step above.