

# Notes on CNV Methods

Mehrtash Babadi,<sup>\*</sup> David Benjamin,<sup>†</sup> and Samuel K. Lee<sup>‡</sup>

*Broad Institute, 75 Ames Street, Cambridge, MA 02142*

(Dated: February 8, 2017)

Some notes on current and proposed methods used in the GATK CNV and ACNV workflows.

## I. INTRODUCTION

The GATK uses two types of information from sequencing data to detect copy number variations (CNVs). First, targets (usually exons, but in principle any genomic locus) with abnormally high or low coverage suggest amplifications or deletions, respectively. Second, sites that are heterozygous in a normal sample and have allele ratios significantly different from 1:1 in the matched tumor sample imply a CNV event involving one or both alleles. The workflow is correspondingly split into two major portions:

1. GATK CNV: Using coverage data that has been normalized against a panel of normals to remove sequencing noise, targets are partitioned into continuous segments that represent the same copy-number event. The segmentation is performed by a circular-binary-segmentation (CBS) algorithm described by Olshen et al. 2004 that was originally developed to segment noisy array copy-number data.<sup>1</sup> Amplifications, deletions, and copy-neutral regions are called from the segmentation.
2. GATK ACNV: Heterozygous sites are identified in the normal case sample and segmented, again using CBS, according to their ref:alt allele ratios in the tumor sample. These allele-fraction segments are combined with the copy-ratio segments found by GATK CNV to form a common set of segments. Modeling of both the copy ratio and minor allele fraction of each segment is alternated with the merging adjacent segments that are sufficiently similar according to this model, until convergence.

## II. STEPS IN THE GATK CNV AND ACNV WORKFLOWS

### A. Coverage collection

[[SL: Details of coverage collection go here.](#)] This is implemented by the GATK command-line tool `CalculateTargetCoverage`.

### B. Creation of a panel of normals

We cannot simply divide the coverage of each target by the average sequencing depth to obtain an estimate of its copy ratio. This is because the coverage of different targets is heavily-biased by factors including the efficiency of their baits, GC content, and mappability. In order to detect CNVs, we must determine these systematic effects on the coverage of each target in the absence of CNVs, which requires a panel of normal samples (PoN) that are representative of the sequencing conditions of the case sample. PoN samples must also be created using the same baits as the case sample.

The steps for creating a panel of normals are:

1. Obtain the coverage (total number of overlapping reads) of every target and sample.
2. Calculate the median coverage of each target over all samples.

---

<sup>\*</sup>Electronic address: [mbabadi@broadinstitute.org](mailto:mbabadi@broadinstitute.org)

<sup>†</sup>Electronic address: [davidben@broadinstitute.org](mailto:davidben@broadinstitute.org)

<sup>‡</sup>Electronic address: [slee@broadinstitute.org](mailto:slee@broadinstitute.org)

<sup>1</sup> Specifically, the CBS implementation provided by the R package `DNACopy` is used.

3. Filter out targets whose median coverage is below a given percentile (by default 25%) of target medians.
4. Divide all coverages by their corresponding target medians.
5. Filter out samples with too great a proportion of zero-coverage targets (by default 5%).
6. Filter out targets with zero coverage in too great a proportion of samples (by default 2%).
7. Filter out samples whose median coverage is above or below certain percentiles (by default 2.5% and 97.5%) of sample medians.
8. Replace all remaining zero coverages with their corresponding target median.
9. Calculate the range of coverage from percentile  $p\%$  to  $(100 - p)\%$  for each target and truncate coverages at each target to lie within these ranges. By default  $p = 0.1$ .
10. Divide each coverage by its sample median.
11. Take the  $\log_2$  of each coverage.
12. Calculate the median of each sample and take the median of these over all targets. Subtract this median of medians from each coverage.
13. Perform a singular value decomposition (SVD) of the resulting matrix and calculate its pseudo-inverse truncated to the space spanned by the  $k$  right eigenvectors with largest singular values. Choose  $k$  using Jolliffe’s heuristic of retaining singular values greater than 0.7 times the mean singular value.

This procedure is implemented by the GATK tool `CreatePanelOfNormals`. The output is: a  $N \times k$  matrix  $P$ , the columns of which are the retained right eigenvectors (eigensamples), and its pseudoinverse  $P^+$ ; and the target medians (before any transformations). Here  $N$  denotes the number of targets.

### C. Segmentation by tangent-normalized coverage

We first divide the integer coverage of the case sample at each target by the corresponding target median from the PoN and take the  $\log_2$  transformation to obtain an  $N \times 1$  column matrix  $\mathbf{x}$ . We then calculate the “tangent-normalized” coverage:  $\mathbf{x} - PP^+\mathbf{x}$ . The meaning of this is as follows:  $PP^+$  is an operator that projects onto the column space of  $P$ . That is, it projects onto the space spanned by the  $k$  most significant eigensamples representing the (non-CNV-related) variability of the coverage. Subtracting the projection  $PP^+\mathbf{x}$  therefore isolates the CNV signal and removes noise due to fluctuations in sequencing bias. This is implemented by the GATK tool `NormalizeSomaticReadCounts`.

Finally, the tangent-normalized coverage vector is passed to CBS to obtain coverage segments. This is implemented by the GATK tool `PerformSegmentation`.

### D. Calling of events from coverage segments

[[SL: Description of caller goes here.](#)] This is performed by the GATK tool `CallSegments`, which is the final step in the GATK CNV portion of the case-sample workflow.

### E. Collection of allele counts at het sites

The first step in the GATK ACNV portion of the case-sample workflow is to gather the necessary allele-count data. This procedure is implemented by the GATK tool `GetHetCoverage`.

Given a large database of common SNPs, we search the normal control sample for heterozygous sites. To determine whether a site with  $r$  ref reads and  $a$  alt reads is heterozygous, we calculate the two-sided  $p$ -value under the null hypothesis that the number of alt reads follows a binomial distribution:  $a \sim \text{Binom}(a + r, 1/2)$ . If the  $p$ -value is not too small we consider the site heterozygous. Ref and alt counts are then obtained at these sites in the tumor case sample.

Alternatively, the GATK tool `GetBayesianHetCoverage`, which instead performs Bayesian hypothesis testing to identify het sites, can be used to perform this step; see Sec. II K for details.

The results of this step are combined with the coverage segments generated by GATK CNV and passed to the GATK tool `AllelicCNV`, which performs the rest of the steps in the GATK ACNV workflow.

## F. Segmentation by minor allele fraction

To obtain initial minor-allele-fraction segments, we estimate the minor allele fraction for each het site under the allele-fraction model discussed in Sec. III B. Specifically, we begin by taking the maximum-likelihood estimates given by Equation 16 with reference bias ignored (i.e.,  $\lambda_j = 1$ ) and segmenting them with CBS. Using this initial segmentation, the maximum-likelihood values for the reference-bias hyperparameters  $\alpha$  and  $\beta$  (see Sec. III B) are found. The minor allele fraction for each het site is then re-estimated as before, but now assuming  $\lambda_j = \alpha/\beta$  (i.e., the reference bias at each site is taken to be the mean reference bias across all sites<sup>2</sup>), and again segmented upon. This procedure is iterated until the segmentation converges or the number of iterations reaches a specified limit; if a cycle occurs, the first repeated segmentation is taken.

## G. Union of copy-ratio and minor-allele-fraction segments

At this point, per-target estimates of copy ratio (i.e., tangent-normalized coverage) and per-het estimates of minor allele fraction have been segmented separately by CBS. We now perform a segment-union step to combine both segmentations into a single one, the idea being that there may be breakpoints present in the copy-ratio segmentation that were missed in the minor-allele-fraction segmentation and vice versa.<sup>3</sup>

Such a segment union could be created naively by simply taking all segments created by the union of both sets of breakpoints. However, this typically results in oversegmentation, for two reasons: 1) the copy-ratio segmentation itself is typically oversegmented, due to residual systematic noise that is not removed normalization against the panel of normals<sup>4</sup>, and 2) the genomic locations of the copy-ratio and minor-allele-fraction breakpoints corresponding to a single underlying event do not exactly overlap (instead, the breakpoints are given by the minimum spanning intervals containing the targets and hets in that event, respectively, and these intervals are not necessarily identical). We thus perform some heuristic steps to reduce the resulting number of segments.

First, if the copy-ratio segments have been called amplified, deleted, or copy neutral (e.g., by the caller described in Sec. II D), we can use the calls to merge adjacent copy-neutral segments, which partially addresses the first issue. Note, however, that oversegmentation can still remain in adjacent amplified or deleted segments.

Second, we can improve upon a naive union of breakpoints by merging segments that are spuriously created due to the inexact overlap of copy-ratio and minor-allele-fraction breakpoints. Not allowing the formation of segments by the addition of minor-allele-fraction breakpoints near the starts and ends of segments that originate from the copy-ratio segmentation partially addresses this issue. Likewise, merging those segments formed by the addition of minor-allele-fraction breakpoints to the middle of segments that originate from the copy-ratio segmentation also reduces oversegmentation.<sup>5</sup> Here, whether such a segment is merged with the adjacent segment to the left or with the adjacent segment to the right is decided by nonparametric tests of similarity between the tangent-normalized coverages and the allele counts in the segments and the genomic distances between the segments. These tests are also used for small-segment merging, which we describe next.

<sup>2</sup> An issue is filed to relax this assumption, which is only made so that maximum-likelihood estimates of the minor allele fraction can be easily cached as a function of the ref and alt counts. In future releases, we may also incorporate the distribution across samples of reference bias at each site, which can be learned from an allelic panel of normals (discussed in Sec. V A), into the het-segmentation step.

<sup>3</sup> Note that there are several issues filed concerning this step and that it will most likely be modified in the near future.

<sup>4</sup> This can be somewhat alleviated by using the `undo.splits` parameter of CBS, with varying results. With `undo.splits="sdundo"`, CBS will merge segments within a certain number of standard deviations of each other (analogous to the heuristic procedure we use to perform similar-segment merging, which is discussed in Sec. II J). With `undo.splits="prune"`, CBS instead merges segments that result in a proportional increase of the squared error that is less than a specified threshold; however, note that enabling this option often causes CBS to hang indefinitely on samples with a large number ( $\gtrsim 200$ -300) of segments. Issues are filed to investigate alternatives to segmentation algorithms that could offer the possibility of controlling the amount of segmentation in a more principled way than those offered by CBS.

<sup>5</sup> Both of these procedures treat the copy-ratio segments as the primary segmentation, to which the minor-allele-fraction breakpoints are added. This is done for legacy reasons, even though, as previously mentioned, the copy-ratio segmentation often contains residual segmentation from systematic noise. Furthermore, both corrections are only necessary because a naive union of breakpoints only approximates the true procedure we would like to perform here—that is, using changepoints in one series of data to identify changepoints in a second series of data, in the case where the locations of the data points from both series do not strictly overlap. Perhaps a better procedure would be: 1) perform multiple changepoint detection (e.g., using CBS) on both series, 2) use single changepoint detection to look for additional changepoints in the first series that fall in between those from the second series, and vice versa.

## H. Small-segment merging

Using CBS to segment the targets in GATK CNV results in segments that contain at least a specified minimum number of targets  $n_t$  (by default,  $n_t = 3$ ). However, after taking the union of target and SNP segments, small segments with less than  $n_t$  targets may be introduced. To be consistent with CBS and CNV, ACONV treats these small segments as spurious, and removes them by merging them with adjacent segments.

For each small segment, the question we want to answer is: which of the two adjacent segments (i.e., those to the left or right of the small segment in the center) is most similar to the small segment, and hence, should be merged with it? We would like to determine this based on the tangent-normalized  $\log_2$  coverages, allele counts, and genomic locations of each of the three segments. Note that each segment may be missing either coverages or allele counts, but not both. However, this question is not statistically well defined, so we will use the following heuristic procedure:

We first examine the allele counts (as opposed to the coverages—this is because the center segment has a small number of targets, by construction, and so the ability to determine which adjacent segment is more similar to it using the coverages is inherently limited). In particular, we examine the alternate-allele fractions in each of the segments; these should have a bimodal (unimodal) distribution for unbalanced (balanced) segments. The Kolmogorov-Smirnov test statistic, which measures the similarity of two data sets, is used to construct two distances between the alternate-allele fractions in the left and center segments and those in the right and center segments, respectively. Including corrections that account for the sizes of the data sets, the distances are used to construct a pair of scores for merging with the left and right segments, respectively. The adjacent segment with the higher score has alternate-allele fractions that are more similar, and hence should be merged with the center small segment.

However, if the Kolmogorov-Smirnov distances are not sufficiently dissimilar, if they are both close to unity (i.e., if the alternate-allele fractions in neither the left nor the right segment overlap significantly with those in the center segment), or if there are not enough hets ( $>2$  data points in each data set) to calculate the Kolmogorov-Smirnov distances, we instead use the inverse minor-allele fractions in each of the three segments. These, ideally, have a distribution that is roughly unimodal in each segment. Two distances between the two pairs of data sets are constructed using the Hodges-Lehmann estimator, which gives a measure of the difference in the location parameters of two data sets, and these distances are used to construct a pair of scores as above.

If the scores generated from the allele counts are too similar or if any of the segments is missing hets, then we attempt to use the coverages instead. Here, we simply use the Hodges-Lehmann estimator to construct two distances and a corresponding pair of scores as above.

If the scores generated from the coverages are too similar or if any of the segments is missing targets, we then simply use genomic distance (defined between adjacent breakpoints) to decide which adjacent segment is closer. For consistency, we also convert the two genomic distances into a pair of scores that sum to unity.

In the unlikely event that all of the above scores are equal, we randomly choose one of the adjacent segments and merge it with the center small segment.

Although this procedure is admittedly quite ad-hoc, it performs reasonably well on simulated data. Furthermore, it is unlikely that incorrect merging of small segments has a significant adverse effect on the subsequent model-fitting step, which we discuss next.

## I. Model fitting

At this point, a common segmentation of the genome has been derived from both the tangent-normalized coverage and the het allele counts. Each segment contains at least  $n_t$  coverages and may or may not also contain hets.

We now proceed to separately fit a copy-ratio model to the coverages and a minor-allele-fraction model to the allele counts. Both models contain local, segment-level parameters which represent the  $\log_2$  copy ratio and minor allele fraction, respectively, as well as global parameters that attempt to model systematic noise and biases arising from sequencing. These models are discussed in detail in Sec. III.

We use Markov Chain Monte Carlo (MCMC) to generate a specified number of samples from the posteriors of each of the parameters. These samples are used to generate posterior summary statistics; the posterior mode, the 95% highest posterior density credible interval, and deciles are reported for the segment-level parameters.

## J. Similar-segment merging

We next perform a smoothing step on the fitted copy-ratio and minor-allele-fraction models to further reduce the segmentation.<sup>6</sup> Proceeding across the genome from left to right, we examine the posterior summaries for adjacent pairs of segments. If both the credible intervals for the local copy-ratio and minor-allele-fraction parameters overlap by a specified amount, the segments are merged. The posterior summaries for the newly created segment are determined from those of the original two segments by approximating all posteriors as Gaussian (i.e., using inverse-variance weighting). The new segment is then repeatedly checked for similarity against the adjacent segment to the right and merged until it is no longer similar, at which point we proceed to the next pair of adjacent segments. After one complete traversal of the genome, we optionally refit both models using MCMC.

This procedure is iterated until the segmentation converges or the number of iterations reaches a specified limit. Both models are then refit using MCMC if necessary, resulting in the final output of GATK ACNV: posterior summaries for both  $\log_2$  copy ratio and minor allele fraction in each segment.

## K. Detection of het sites using a Bayesian model

Here, we describe a procedure for calling heterozygous (**Het**) sites that (1) takes into account the base read alignment and sequencing qualities, and (2) works for both normal and tumor data. This procedure is implemented by the GATK command-line tool `GetBayesianHetCoverage`.

Provisioning situations that only the tumor data is available to us ("tumor-only"), in addition to the presently considered situation of paired normal-tumor data ("paired normal-tumor"), we need to modify our criterion for calling a **Het** site. Conceptually, since reads from tumor samples are not pure (contaminated with subclones, normals, etc), a statistical test that rejects the **Het** hypothesis based on the premise of having equal probability of Ref and Alt reads is bound to reject **Het** cases when applied to tumor reads. Here, we propose a more sensible Bayesian model.

**Notation:** Let us first focus on a single site  $j$ , with  $R_{kj} \in \{A, C, T, G\}$  denoting the mapped base at site  $j$  from read  $k$ , and  $\varepsilon_{kj}^B$  and  $\varepsilon_k^M$  denoting the err probability of base calling and mapping. Also, let  $\text{Ref}_j$  and  $\text{Alt}_j$  denote the Ref and Alt alleles at this site.

**Definition of error:** In case of a base error event, the base could be read as any other three bases with equal probability. In case of a mapping error event, we assume equal probability for all four bases.

**Rareness of somatic SNP events:** In order to proceed with the model, we assume that somatic SNPs are rare events such that Hom/Het sites retain their germline identity.

**Likelihood of Hom<sub>j</sub>:** Assuming that site  $j$  is homozygous (**Hom**), we find the likelihood of the reads by conditioning over the allele and error events. We easily find:

$$P(R_{kj}|\text{Hom}_j) = P(\text{Ref}_j|\text{Hom}_j) \prod_{k=1}^{N_j} \left[ \frac{\varepsilon_{kj}^B}{3} + \frac{\varepsilon_k^M}{4} + \left( 1 - \frac{4}{3} \varepsilon_{kj}^B - \varepsilon_k^M \right) \delta_{R_{kj}, \text{Ref}_j} \right] + \\ P(\text{Alt}_j|\text{Hom}_j) \prod_{k=1}^{N_j} \left[ \frac{\varepsilon_{kj}^B}{3} + \frac{\varepsilon_k^M}{4} + \left( 1 - \frac{4}{3} \varepsilon_{kj}^B - \varepsilon_k^M \right) \delta_{R_{kj}, \text{Alt}_j} \right]. \quad (1)$$

We need to know the two priors  $P(\text{Ref}_j|\text{Hom}_j)$  and  $P(\text{Alt}_j|\text{Hom}_j)$ , both of which can be estimation from the statistics of the population to which the sample belongs. If this data is not available, we may use the flat prior  $P(\text{Ref}_j|\text{Hom}_j) = P(\text{Alt}_j|\text{Hom}_j) = 1/2$  with little harm.

**Likelihood of Het<sub>j</sub>:** Assuming that site  $j$  is **Het**, and that the the probability of the Ref allele in the sample is  $f_{j,R}$ ,

---

<sup>6</sup> This step is analogous to the aforementioned `undo.splits="sdundo"` procedure optionally performed by CBS.

we have:

$$\begin{aligned}
p_{kj,R} &\equiv P(R_{kj} = \text{Ref}_j | \text{Het}_j, f_{j,R}) = (1 - \varepsilon_{kj}^B - \varepsilon_k^M) f_{j,R} + \frac{\varepsilon_{kj}^B}{3} (1 - f_{j,R}) + \varepsilon_k^M / 4, \\
p_{kj,A} &\equiv P(R_{kj} = \text{Alt}_j | \text{Het}_j, f_{j,R}) = (1 - \varepsilon_{kj}^B - \varepsilon_k^M) (1 - f_{j,R}) + \frac{\varepsilon_{kj}^B}{3} f_{j,R} + \varepsilon_k^M / 4, \\
p_{kj,\circ} &\equiv P(R_{kj} \neq \text{Ref}_j, \text{Alt}_j | \text{Het}_j, f_{j,R}) = \frac{\varepsilon_{kj}^B}{3} + \frac{\varepsilon_k^M}{4}.
\end{aligned} \tag{2}$$

Therefore, the likelihood reads:

$$P(\{R_{kj}\} | \text{Het}_j) = \int_0^1 df_{j,R} P(f_{j,R} | \text{Het}_j) \prod_{k=1}^{N_j} p_{kj,R}^{I(R_{kj}=\text{Ref}_j)} p_{kj,A}^{I(R_{kj}=\text{Alt}_j)} p_{kj,\circ}^{I(R_{kj} \neq \text{Ref}_j, \text{Alt}_j)}. \tag{3}$$

The integration over  $f_{j,R}$  is not as trivial as before since (1) the error probabilities differs from site to site, and (2) the prior is not necessary conjugate to  $\text{Het}$  likelihood. For the uniformity of notation, we define:

$$\begin{aligned}
R_{kj} = \text{Ref}_j &\Rightarrow \alpha_{kj} \equiv \frac{\varepsilon_{kj}^B}{3} + \frac{\varepsilon_k^M}{4}, \quad \beta_{kj} = 1 - \frac{4\varepsilon_{kj}^B}{3} - \frac{\varepsilon_k^M}{4}, \\
R_{kj} = \text{Alt}_j &\Rightarrow \alpha_{kj} \equiv 1 - \varepsilon_{kj}^B - \frac{3\varepsilon_k^M}{4}, \quad \beta_{kj} = -1 + \frac{4\varepsilon_{kj}^B}{3} + \varepsilon_k^M.
\end{aligned} \tag{4}$$

such that:

$$P(\{R_{kj}\} | \text{Het}_j) = \left[ \prod_{k \in \mathcal{I}_\circ} \frac{\varepsilon_{kj}^B}{3} \right] \left[ \int_0^1 df P(f | \text{Het}) \prod_{k \in \mathcal{I}_{RA}} (\alpha_{kj} + \beta_{kj} f) \right], \tag{5}$$

where  $\mathcal{I}_\circ$  are the indices of reads that are neither Ref or Alt at site  $j$ , and  $\mathcal{I}_{RA}$  are indices of reads that either Ref or Alt. Furthermore,  $P(f | \text{Het})$  is the common prior for Ref allele fraction. For a given prior, we calculate the  $f$ -integral numerically with a fixed-order quadrature. Since the integrand is polynomial of  $f$ , a Gaussian quadrature is well-suited to approximate the integral provided that the prior is also smooth.

**Caveats:** (1) sensitivity to error underestimation: if the read/alignment qualities are overestimated, even a single deviation from the  $\text{Hom}_j$  hypothesis can dramatically reduce the likelihood. (2) Loss of heterozygosity can manifest itself as homozygosity; in practice, it should not be an issue since the samples are not pure and germline heterozygosity should yield sufficient evidence to reject the  $\text{Hom}$  hypothesis. (3) Heterozygosity in a sizable subclone resulting from a somatic SNP may manifest itself as germline heterozygosity. This is also expected not to be a major issue since somatic SNPs are rare.

**A model prior for allele fraction at Het sites:** In this section, we construct a simple prior for the Ref allele fraction at Het sites. To this end, we assume (1) a minimum (maximum) fraction  $\rho_{\min}$  ( $\rho_{\max}$ ) of the cells in the sample may have events that change the allele fraction with respect to germline (large copy number events, CNLOH, etc). Furthermore, we assume that the maximum copy number is bounded from above by  $N_c$ . Otherwise, we assume flat priors over both the copy number and non-germline fraction. Under these assumptions, the distribution of the Ref allele fraction is given by:

$$P(f | \text{Het}) = \frac{1}{(N_c + 1)^2} \sum_{n,m=0}^{N_c} \int_{\rho_{\min}}^{\rho_{\max}} \frac{d\rho}{\rho_{\max} - \rho_{\min}} \delta\left(f - \frac{(1-\rho) + \rho m}{2(1-\rho) + \rho(m+n)}\right). \tag{6}$$

Since the prior will be symmetric under the transformation  $f \rightarrow 1 - f$ , we will assume  $f < 1/2$  hereafter. The  $\rho$  integration is trivially performed and we find:

$$\begin{aligned}
P(f | \text{Het}) &= \frac{1}{(N_c + 1)^2} \sum_{n,m=0}^{N_c} \frac{1}{\rho_{\max} - \rho_{\min}} \frac{|n-m|}{[1-m+f(n+m-2)]^2} \theta\left(f - \frac{(1-\rho_{\max}) + \rho_{\max}m}{2(1-\rho_{\max}) + \rho_{\max}(m+n)}\right) \\
&\quad \times \theta\left(\frac{(1-\rho_{\min}) + \rho_{\min}m}{2(1-\rho_{\min}) + \rho_{\min}(m+n)} - f\right).
\end{aligned} \tag{7}$$

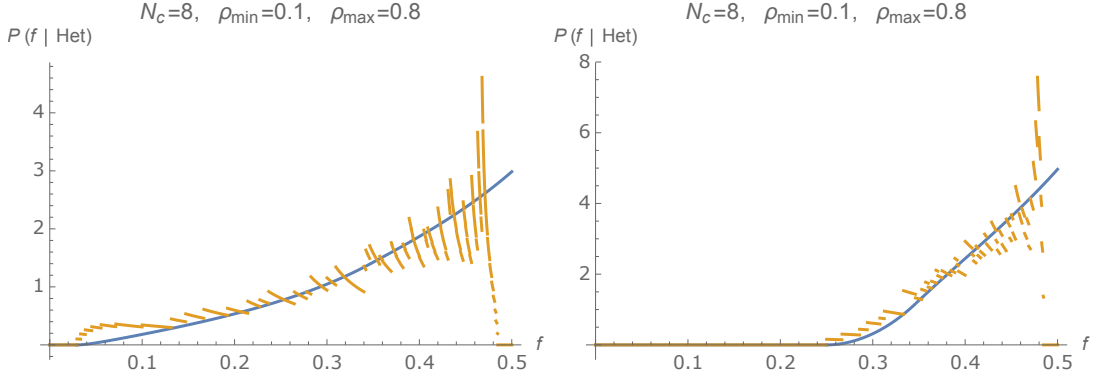


FIG. 1: Two examples of the Ref allele fraction prior  $P(f|\text{Het})$  at Het sites based on minimum/maximum non-germline cells and maximum copy number. The blue lines denote the continuous approximation given in Eq. (8), The discontinuous orange lines denote the result with discrete copy number summation given in Eq. (the delta function peak at  $f = 1/2$  is not shown).

The summand is ambiguous for  $n = m = 1$  since  $f$  evaluates to  $1/2$  independent of  $\rho$ . The correct prescription is to replace it with  $\delta(f - 1/2)/(\rho_{\max} - \rho_{\min})$ .

The discrete summation over the copy numbers  $(n, m)$  result in a discontinuous prior. It is convenient to approximate the discrete summations with integrals over  $n$  and  $m$ . This approximation preserves the main features of the prior while converges to the discrete result for large  $N_c$ . The double integral over  $(n, m)$  must be performed with diligence since the Heaviside functions restrict the integration region depending on the value of  $f$ . We leave out the details and just quote the final result:

$$P(f|\text{Het}) = \begin{cases} P_{<}(f) & f_{\text{th}} \leq f \leq f^*, \\ P_{>}(f) & f^* < f \leq \frac{1}{2}, \end{cases} \quad (8)$$

where:

$$\begin{aligned} f_{\text{th}} &= \frac{1 - \rho_{\max}}{N_c \rho_{\max} + 2(1 - \rho_{\max})}, \\ f^* &= \frac{1 - \rho_{\min}}{N_c \rho_{\min} + 2(1 - \rho_{\min})}, \\ P_{<}(f) &= \frac{(\rho_{\max}(fN_c - 1) - 1)(f((N_c - 2)\rho_{\max} + 2) + \rho_{\max} - 1) + 2\rho_{\max}(f(fN_c - 2) + 1) \log\left(\frac{\rho_{\max}(f(N_c - 2) + 1)}{1 - 2f}\right)}{2(f - 1)^2 f^2 N_c^2 \rho_{\max} (\rho_{\max} - \rho_{\min})}, \\ P_{>}(f) &= \frac{(\rho_{\max} - \rho_{\min})(\rho_{\max}\rho_{\min}(f(N_c - 2) + 1)(fN_c - 1) + 2f - 1) + 2\rho_{\max}\rho_{\min}(f(fN_c - 2) + 1) \log\left(\frac{\rho_{\max}}{\rho_{\min}}\right)}{2(f - 1)^2 f^2 N_c^2 \rho_{\max} \rho_{\min} (\rho_{\max} - \rho_{\min})}. \end{aligned} \quad (9)$$

Fig. 1 shows two examples of this prior along with the version with discrete copy number summations.

**The Bayesian decision rule:** Using the Bayes' theorem, the log odds of heterozygosity is found as:

$$\log \text{odds}(\text{Het}_j) = \log P(\{R_{kj}\}|\text{Het}_j) + \log P(\text{Het}_j) - \log P(\{R_{kj}\}|\text{Hom}_j) - \log P(\text{Hom}_j). \quad (10)$$

In order to evaluate the right hand side, we need to have knowledge of the prior  $P(\text{Het}_j)$ . This can be worked out from population statistics. Otherwise, we may use the flat prior  $P(\text{Het}_j) = 1/2$ .

Having the log odds, the decision rule is simple: we call a **Het** site if its odds exceeds a given threshold:

$$\text{Call Het}_j \Leftrightarrow \log \text{odds}(\text{Het}_j) > \log \frac{1 - 10^{-s_{\text{Het}}}}{10^{-s_{\text{Het}}}} = \log(10^{s_{\text{Het}}} - 1), \quad (11)$$

where we have defined the **Het** calling stringency parameter  $s_{\text{Het}}$  as a convenient parametrization of the decision boundary. Finally, we note that the log likelihoods scale linearly with the read depth  $N_j$  (each read results in an additional multiplicative term). Therefore, the statistic  $\log \text{odds}(\text{Het}_j)$  linearly deviates from the decision threshold

$\log(10^{s_{\text{Het}}} - 1) \propto s_{\text{Het}}$  as the read depth increases.

**Increasing power using haplotype information:** Todo. The basic idea is to utilize SNP correlations to test multiple correlated sites simultaneously for heterozygosity (1) to increase power, and (2) to improve the prior on  $\text{Ref}_j/\text{Alt}_j$ . A good starting point to run HaplotypeCaller on a few normal/tumor reads and check the strength/range/size of correlations between SNP constellations.

### III. GATK CNV/ACNV MODELS

#### A. Copy-ratio model

We fit a simple, heuristic copy-ratio model to the segmented, tangent-normalized  $\log_2$  coverages. Our primary goal is to estimate accurate mean  $\log_2$  copy ratios within each segment for use in downstream tools, so a simple model suffices assuming that the tangent-normalization step has removed most of the systematic noise.

The model assumes that the tangent-normalized  $\log_2$  coverages in each segment are distributed as a mixture of: 1) a Gaussian distribution, with mean given by a segment-level  $\log_2$  copy-ratio parameter and a variance that is common to all the segments<sup>7</sup>, and 2) a uniform distribution, which is meant to model outlier  $\log_2$  coverages and is truncated at the minimum and maximum observed values.

This model can be expressed in terms of local target-level outlier indicators and segment-level parameters for the mean  $\log_2$  copy ratio, as well as global parameters for the variance and outlier-distribution mixture fraction. We generate posterior samples by Gibbs sampling the conditional distributions for each parameter in succession; slice sampling is used for continuous parameters.

#### B. Allelic model

We want a generative model for allelic fractions that infers its parameters from the data. We observe alt and ref read counts for each het site and wish to infer the minor allelic fraction of every segment. Let's consider what other hidden variables belong in the model. Read counts obey an overdispersed binomial distribution in which the probability of an alt read is a site-dependent random variable. Letting  $\theta_j$  be the probability that a mapped read at het  $j$  is an alt we have

$$P(a_j, r_j | \theta_j) = \binom{a_j + r_j}{a_j} \theta_j^{a_j} (1 - \theta_j)^{r_j} = \binom{n_j}{a_j} \theta_j^{a_j} (1 - \theta_j)^{r_j}, \quad (12)$$

where  $a_j$  and  $r_j$  are alt and ref read counts and  $n_j = a_j + r_j$  is the total read count at site  $j$ . Now we consider  $\theta_j$ . Suppose site  $j$  belongs to a segment with minor allelic fraction  $f$  and is alt minor, such that  $P(\text{alt}) = f$  and  $P(\text{ref}) = 1 - f$  are the probabilities that a random DNA fragment will contain the alt and ref alleles. Let  $x_j^{\text{alt(ref)}} = P(\text{mapped}|\text{alt(ref)})$  be the probabilities that an alt (ref) DNA fragment at site  $j$  eventually gets sequenced and mapped. Then  $\theta_j$  is the conditional probability that a mapped read comes from an alt fragment:

$$\theta_j = P(\text{alt}|\text{mapped}) = \frac{P(\text{alt})P(\text{mapped}|\text{alt})}{P(\text{alt})P(\text{mapped}|\text{alt}) + P(\text{ref})P(\text{mapped}|\text{ref})} \quad (13)$$

$$= \frac{f x_j^{\text{alt}}}{f x_j^{\text{alt}} + (1 - f) x_j^{\text{ref}}} = \frac{f}{f + (1 - f) \lambda_j}, \quad (14)$$

where  $\lambda_j = x_j^{\text{ref}}/x_j^{\text{alt}}$  is the “bias ratio” of ref to alt sequenceability and mappability at site  $j$ . A similar result for ref minor sites follows from substituting  $f \leftrightarrow 1 - f$ . In addition to the bias ratio  $\lambda_j$  we need an indicator variables

---

<sup>7</sup> Note that this differs from AllelicCapSeg, which assumed that non-outlier, tangent-normalized non- $\log_2$  coverages were Gaussian distributed with variance proportional to the mean in each segment. However, examination of many samples shows that this proportionality is only weakly exhibited, on average, in the non- $\log_2$  coverages—and even less so in the  $\log_2$  coverages, which exhibit a variance than is, on average, constant with mean  $\log_2$  copy ratio. It is true that allowing the variance in each segment to be drawn from a global distribution, rather than fixed to a single number, would provide a better fit to the data; however, the estimates of the mean  $\log_2$  copy ratio are not likely to be strongly biased under the simple model. Furthermore, this copy-ratio model will soon be superseded by the generative coverage model proposed in Sec. VB.



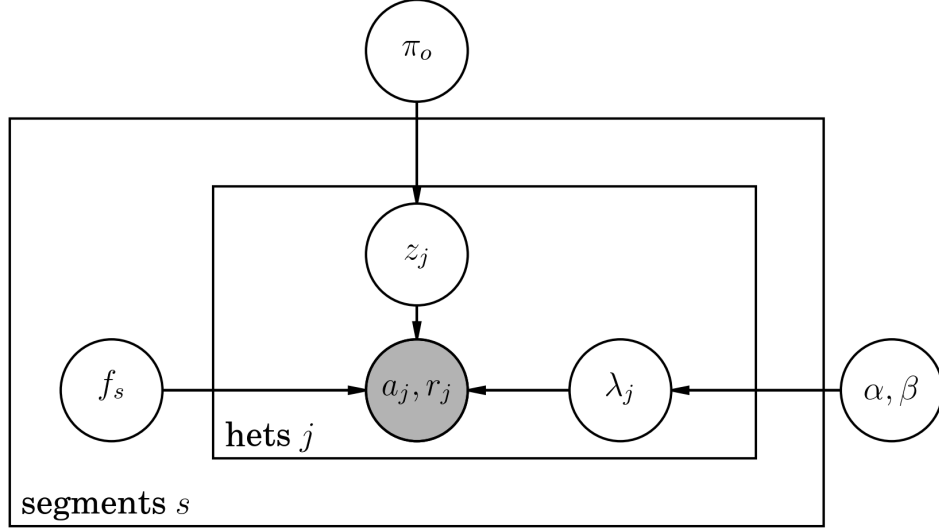


FIG. 2: Graphical model for ACNV allelic model

$z_j$  with three states, alt minor, ref minor, and an outlier state that gives robustness to anomalous events. For this outlier state we average the binomial likelihood over all  $\theta$  to get:

$$P(a_j, r_j | \text{outlier}) = \binom{n_j}{a_j} \int_0^1 \theta_j^{a_j} (1 - \theta_j)^{r_j} d\theta_j = \binom{n_j}{a_j} \frac{a_j! r_j!}{(n_j + 1)!} \quad (15)$$

For notational convenience we give  $z_j$  a one-of- $K$  encoding  $z_j = (z_{ja}, z_{jr}, z_{jo})$  in which one component equals 1 and the rest 0.

The contribution of site  $j$  to the likelihood is

$$P(a_j, r_j | f_j, \lambda_j, z_j) = \binom{n_j}{a_j} \left[ \frac{f_j^{a_j} (1 - f_j)^{r_j} \lambda_j^{r_j}}{(f_j + (1 - f_j) \lambda_j)^{n_j}} \right]^{z_{ja}} \left[ \frac{(1 - f_j)^{a_j} f_j^{r_j} \lambda_j^{r_j}}{(1 - f_j + f_j \lambda_j)^{n_j}} \right]^{z_{jr}} \left[ \frac{a_j! r_j!}{(n_j + 1)!} \right]^{z_{jo}} \quad (16)$$

where  $f_s$  is the minor allele fraction of the segment containing site  $j$ . We will consider  $f$  to be drawn from a uniform distribution on  $[0, 1/2]$  – that is, we give it a flat prior – but in the future we can obtain some sort of clustering behavior, representing the fact that events in the same subclone that exhibit the same integer copy numbers will have the same minor allelic fractions, by drawing  $f_s$  from a Dirichlet process.

We assume that the bias ratios come from a common Gamma distribution with parameters  $\alpha, \beta$ :

$$P(\lambda_j | \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_j^{\alpha-1} e^{-\beta \lambda_j} \quad (17)$$

Note that bias ratios tend to be near 1.0 and so the choice of distribution is not too important as long as it has adjustable mean and standard deviation. We choose the Gamma distribution because it is the simplest such distribution on  $\mathbb{R}^+$ . We will give the parameters  $\alpha$  and  $\beta$  a flat prior  $P(\alpha, \beta) \propto 1$ .

Finally, the indicator  $z_j$  is a multinomial random variable distributed according to parameter vector  $\pi$ :

$$P(z_{ja(r,o)} = 1 | \pi) = \pi_{a(r,o)} \quad (18)$$

We set the alt and ref minor probabilities equal so that the only free parameter is  $\pi = \pi_o$ , with  $\pi_{a(r)} = (1 - \pi)/2$ . The Bayesian network corresponding to this model is shown in Figure III B.

As with the other parameters, we put a flat prior on  $\pi$ . Putting all the pieces together the likelihood is

$$\mathbb{L} = \prod_j \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_j^{\alpha-1} e^{-\beta \lambda_j} \left[ \frac{(1 - \pi) f_j^{a_j} (1 - f_j)^{r_j} \lambda_j^{r_j}}{(f_j + (1 - f_j) \lambda_j)^{n_j}} \right]^{z_{ja}} \left[ \frac{(1 - \pi) (1 - f_j)^{a_j} f_j^{r_j} \lambda_j^{r_j}}{(1 - f_j + f_j \lambda_j)^{n_j}} \right]^{z_{jr}} \left[ \frac{2\pi a_j! r_j!}{(n_j + 1)!} \right]^{z_{jo}}. \quad (19)$$

The dependence on  $\lambda$  for alt minor sites is

$$g(\lambda_j, \alpha, \beta, f_j, a_j, r_j) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{f_j^{a_j} (1 - f_j)^{r_j} \lambda_j^{\alpha + r_j - 1} e^{-\beta \lambda_j}}{(f_j + (1 - f_j) \lambda_j)^{n_j}}. \quad (20)$$

For ref minor sites the dependence is the same but with  $f \leftrightarrow 1 - f$ . We show in Appendix A that this function can be integrated analytically, and thus we can marginalize  $\lambda$  out of the model to obtain the likelihood

$$\prod_j \left[ \frac{1 - \pi}{2} \phi(\alpha, \beta, f_j, a_j, r_j) \right]^{z_{ja}} \left[ \frac{1 - \pi}{2} \phi(\alpha, \beta, 1 - f_j, a_j, r_j) \right]^{z_{jr}} \left[ \frac{\pi a_j! r_j!}{(n_j + 1)!} \right]^{z_{jo}}, \quad (21)$$

where  $\phi(\alpha, \beta, f_j, a_j, r_j) = \int_0^\infty g(\lambda, \alpha, \beta, f, a, r) d\lambda$ . Pseudocode for computing  $\phi$  is presented in Appendix A. Furthermore, marginalizing out  $z$  is trivial – simply sum each term over its three possible states. We then have a collapsed likelihood

$$p(f, \alpha, \beta, \pi) \propto \prod_j \left[ \frac{1 - \pi}{2} \phi(\alpha, \beta, f_j, a_j, r_j) + \frac{1 - \pi}{2} \phi(\alpha, \beta, 1 - f_j, a_j, r_j) + \frac{\pi a_j! r_j!}{(n_j + 1)!} \right] \quad (22)$$

Integrating out the latent variables removes the strongest correlations from the model – intuitively,  $f$  should not be too sensitive to  $\alpha$  and  $\beta$ , for example – and significantly improves mixing. The exception is  $\alpha$  and  $\beta$ , since adjusting one with the other fixed changes the mean of the prior on  $\lambda$ . Thus we reparameterize in terms of  $\mu$  and  $\sigma^2$ , the mean and variance of the common gamma distribution of biases, where  $\alpha = \mu^2/\sigma^2$  and  $\beta = \mu/\sigma^2$ . Due to the weak correlations our MCMC method does not need to be very sophisticated. We choose to sample each variable with one-dimensional adaptive Metropolis, tuning the proposal step size to achieve some reasonable acceptance rate like 0.4 or so. Thus we have completely specified an MCMC scheme for this model, given by Algorithm 1:

---

**Algorithm 1** MCMC algorithm for ACNV allelic model

---

- 1: Initialize all parameters to a maximum likelihood initial guess (see below).
  - 2: **repeat**
  - 3:   Sample each  $f_s$  with adaptive Metropolis
  - 4:   Sample  $\pi$  with adaptive Metropolis
  - 5:   Sample  $\mu$  with adaptive Metropolis
  - 6:   Sample  $\beta$  with adaptive Metropolis
  - 7: **until** convergence
- 

We initialize the model by finding the mode of likelihood. This significantly reduces burn-in time of our MCMC sampling. Also, it allows us to give the adaptive Metropolis samplers better initial guesses for their step sizes. Since in practice there is a single global maximum of the likelihood it is easy to find. After initializing the initialization with rough guesses for the parameters, we successively find one-dimensional maxima adjusting one parameter at a time until the likelihood converges. One could use multidimensional optimization to obtain faster convergence, but after marginalizing out latent parameters the remaining correlations are weak and thus this simple approach performs quite well. Since we may delegate one-dimensional maximization to mathematical libraries, the only thing left to describe is our initial coarse guess.

In the initial guess we set the outlier probability  $\pi_o = 0.01$ ,  $\mu = 1.0$ , and  $\sigma^2 = 0.1$ . With the exception of  $\sigma^2$  these are all reasonable guesses. We choose  $\sigma^2$  to be larger than what we actually believe because  $\mu$  converges more slowly from a bad initial guess if  $\sigma^2$  is too small. The only non-trivial part of the initial guess is the minor allele fractions. For each segment, we wish to set the minor allele fraction to the number of reads from minor alleles divided by total number of reads – this is an unbiased estimator if allelic bias is absent. The problem is that we have counts of alt and ref reads, not minor and major reads. Our solution is to weight the alt and ref read counts on each het by probabilities that the het is alt and ref minor, respectively. That is, we set

$$f_S \approx \frac{\sum_{j \in S} a_j P(z_{ja} = 1) + r_j P(z_{jr} = 1)}{\sum_{j \in S} (a_j + r_j) (P(z_{ja} = 1) + P(z_{jr} = 1))} \quad (23)$$

For this coarse guess we ignore the possibility of outliers, so that  $P(z_{ja} = 1) + P(z_{jr} = 1) = 1$ . Ignoring bias and outliers the alt minor likelihood of het  $j$  is proportional to  $f_j^{a_j} (1 - f_j)^{r_j}$ . Since we don't know  $f$  yet, we integrate this (including the normalization) from  $f = 0$  to  $f = 1/2$  in order to get  $P(z_{ja} = 1)$ . This quantity is called the incomplete regularized beta function  $I$ . Thus we have

$$P(z_{ja} = 1) \approx I(1/2, a_j + 1, r_j + 1), \quad P(z_{jr} = 1) = 1 - P(z_{ja} = 1). \quad (24)$$

### C. Calling segments after allelic CNV workflow

After running the allelic fraction and copy ratio model, we have a list of segments  $s$ , each with its own posterior pdfs  $f_s^{\text{CR}}$  and  $f_s^{\text{MAF}}$  of the copy ratio and minor allele fraction<sup>8</sup>. That is,  $f_s^{\text{MAF}}(x)$  is the posterior probability density from ACNV that segment  $s$  has minor allele fraction  $x$ . We assume that for each segment some fraction  $\rho$  of sequenced cells carry  $m$  and  $n$  copies of the original homologs, while the remaining  $1 - \rho$  cells are diploid. This assumption is compatible with both normal contamination and tumor heterogeneity but not with distinct subclones containing different CNVs at overlapping segments. It *can* express distinct subclones that inherit a CNV from a common ancestor, as well as a single subclone that incurs overlapping CNVs as long as both are fixed (in the population genetics sense) in that subclone.

Each distinct value of  $\rho$  therefore corresponds to a node in the tumor's phylogenetic tree, its value being the proportion of sequenced cells belonging to subclones descended from that node. We therefore expect its values to be drawn from a discrete multinomial distribution, on which we place a symmetric and sparse Dirichlet prior. That is, let  $\rho$  take on values  $\rho_1, \rho_2 \dots \rho_K$  and let  $z_s$  be a binary-valued indicator vector such that  $z_{sk} = 1$  if the CNV on segment  $s$  occurs in fraction  $\rho_k$  of sequenced cells. Then

$$P(\pi|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_k \pi_k^{\alpha/K-1} \quad (25)$$

$$P(z_s|\pi) = \prod_k \pi_k^{z_{sk}} \quad (26)$$

Here  $\alpha$  is the concentration parameter such that the smallness of  $\alpha/K$  enforces sparseness<sup>9</sup>, i.e. most cluster components will not be used. The  $K \rightarrow \infty$  limit is a Dirichlet process and for finite  $K$  to work well,  $K$  must be larger than the number of components needed; in practice making  $K$  twice as large as the number of components works well. The expected number of clusters found in data of size  $N$  (here, the number of segments) is roughly  $\alpha \ln N$ , so we place a vague prior on  $\alpha$  that corresponds to roughly a single- or double-digit number of clusters. For example, a broad gamma prior with mean 1:

$$P(\alpha) = \text{Gamma}(\alpha|1, 1) \quad (27)$$

We have little prior knowledge on tumor's phylogeny, so we put a uniform prior on the values of  $\rho$ :  $P(\rho_k) = 1$ .

Next we relate copy ratio and minor allele fraction to  $(m, n, \rho)$ . The total copy number is a weighted sum of  $(1 - \rho)$  diploid cells and  $\rho$  cells with copy number  $m + n$ .

$$\text{cr}(m, n, \rho) \equiv (2(1 - \rho) + \rho(m + n)) / 2. \quad (28)$$

Similarly, the minor allele fraction is a weighted sum of  $1 - \rho$  diploid cells with a single copy of the minor allele and  $\rho$  cells with  $\min(m, n)$  copies, divided by the total:

$$\text{maf}(m, n, \rho) \equiv \frac{(1 - \rho) + \rho \min(m, n)}{2(1 - \rho) + \rho(m + n)} \quad (29)$$

It is convenient to represent the latent state  $(m, n)$  via binary indicator variables  $v$  and  $w$  with e.g.  $v_{sm} = 1, w_{sn} = 1$  if segment  $s$  has  $m$  and  $n$  copies of the original homologs.

Finally, we place a simple factorized multinomial prior on  $(m, n)$ :  $P(m, n) = P(m)P(n) = \phi_m \phi_n$ , which we can do if we set of maximum copy number of, say,  $m, n < 4$ . The factorization assumption realistic regarding the origin of CNVs but not necessarily regarding their *viability*. For example, a homozygous deletion could be lethal when a heterozygous deletion is not. However, we expect this effect to be less dramatic for small segments, which have less phenotypic impact. Large segments ought to have sufficient statistical power that the prior is less important. Taking into account the copy ratio and minor allele fraction posteriors from ACNV as well as the multinomial prior, the model likelihood is

$$P(z_s, v_s, w_s, \pi, \phi, \rho, \alpha) = P(\alpha) \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_k \pi_k^{\alpha/K-1} \prod_{s,k,n,m} [\pi_k \phi_m \phi_n f_s^{\text{CR}}(\text{cr}(m, n, \rho_k)) f_s^{\text{MAF}}(\text{maf}(m, n, \rho_k))]^{z_{sk} v_{sm} w_{sn}} \quad (30)$$

<sup>8</sup> ACNV obtains MCMC samples from these posteriors; we assume that a reasonable distribution has been fit to these posterior samples.

<sup>9</sup> If  $\alpha < K$  the prior is singular as  $\pi_k \rightarrow 0$  for any  $k$ .

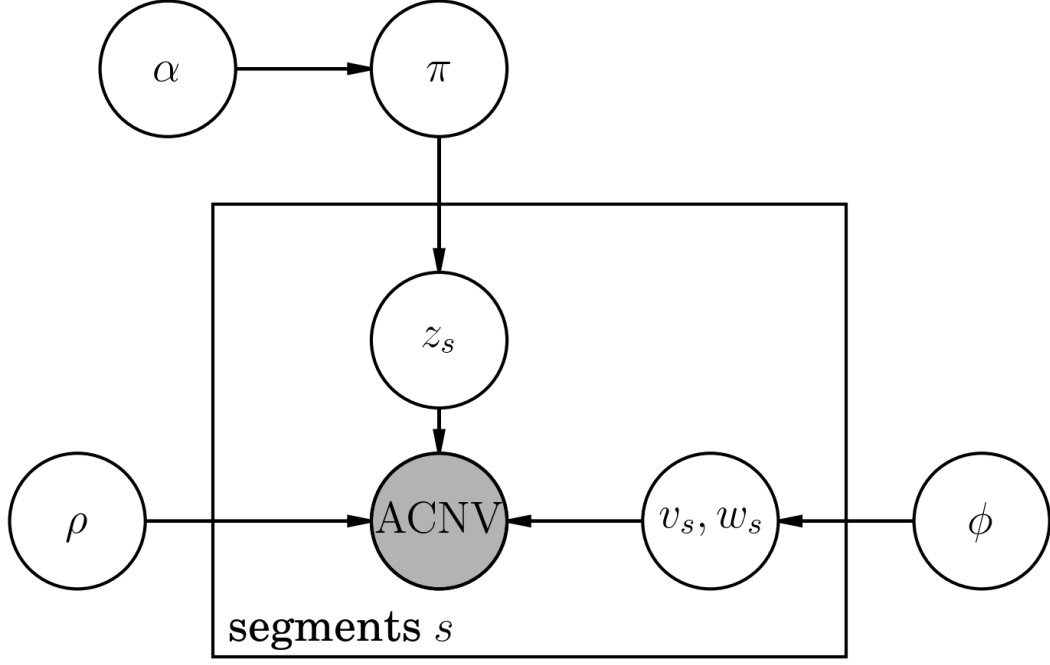


FIG. 3: Graphical model for ACNV caller. “ACNV” represents posterior probability output of ACNV;  $v, w$  are indicators of homolog integer copy numbers;  $\rho$  is the set of values of purity  $\times$  cancer cell fraction;  $z$  is the corresponding indicator;  $\phi$  is the multinomial prior on homolog counts;  $\pi$  is the multinomial prior on  $z$ ;  $\alpha$  is a Dirichlet concentration parameter encouraging a sparse set of  $\rho$  values.

Note that we have simply multiplied of contributions of copy number and minor allele fraction. This is justified because we inferred the former only from total read counts, while the inference for the latter was *conditioned* on the total read depth of each het. Thus there is no double-counting of evidence. This argument is somewhat heuristic because ACNV infers copy number from *target* read counts and minor allele fraction from *SNP* allele counts, but is valid to the extent that total depth at het sites is correlated with depth and the targets they belong to. For off-target SNPs it is not heuristic at all.

The graphical model is shown in Figure III C.

We will obtain maximum likelihood estimates of  $\rho$  and  $\phi$  and give the remaining variables the variational factorized distribution  $p(\alpha, \pi, z, v, w) \rightarrow q(\alpha)q(\pi)q(z, v, w)$ . We now proceed to carry out the standard recipe of the EM and variational Bayes algorithms. Denoting one variable or group of variables by  $X$ , all other variables by  $Z$ , and the joint probability by  $p(X, Z)$ , the mean-field posterior on  $X$  is

$$\ln q(X) = E_{q(Z)}[\ln p(X, Z)] + \text{const} \quad (31)$$

For those variables  $X$  for which we seek a point estimate and not a full posterior we employ a similar formula

$$X = \arg \max [E_{q(Z)}[\ln p(X, Z)]] \quad (32)$$

We will henceforth drop the subscript  $q(Z)$  from the expectation  $E_{q(Z)}$  – all expectations are with respect to the factorized distribution. Following this prescription, we find that the posterior on  $\alpha$  is

$$q(\alpha) \propto \frac{P(\alpha)\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \exp \left( \frac{\alpha}{K} \sum_k E[\ln \pi_k] \right) \quad (33)$$

The posteriors on  $\pi$  and  $\phi$  are

$$q(\pi) \propto \prod_k \pi_k^{E[\alpha]/K - 1 + \sum_s E[z_{sk}]}, \quad q(\phi) \propto \prod_j \phi_j^{\sum_s E[v_{sj} + w_{sj}]} \quad (34)$$

The maximization objective for  $\rho$  is

$$\rho_k = \arg \max \sum_{s,k,n,m} E[z_{sk} v_{sm} w_{sn}] [\ln f_s^{\text{CR}}(\text{cr}(m, n, \rho_k) + \ln f_s^{\text{MAF}}(\text{maf}(m, n, \rho_k))] \quad (35)$$

Lastly,  $q(z, v, w)$  is a categorical distribution which we evaluate by plugging in values:

$$E[z_{sk} v_{sm} w_{sn}] = \frac{\phi_m \phi_n e^{E[\ln \pi_k]} f_s^{\text{CR}}(\text{cr}(m, n, \rho_k) f_s^{\text{MAF}}(\text{maf}(m, n, \rho_k))}{\sum_{k,m,n} \text{“ ”}} \quad (36)$$

Equations 33 – 36 require the expectations  $E[\alpha]$ ,  $E[\ln \pi]$ ,  $E[z_{sk}]$ ,  $E[v_{sj}]$ ,  $E[w_{sj}]$ , and  $E[z_{sk} v_{sm} w_{sn}]$ . The last of these is the E step Equation 36. Three more follow directly from marginalization:

$$E[z_{sk}] = \sum_{m,n} E[z_{sk} v_{sm} w_{sn}], \quad E[v_{sj}] = E[w_{sj}] = \sum_{k,n} E[z_{sk} v_{sj} w_{sn}] \quad (37)$$

By inspection, the Dirichlet posterior  $q(\pi)$  of Equation 34 yields the following logarithmic moments:

$$E[\ln \pi_k] = \psi \left( E[\alpha]/K + \sum_s E[z_{sk}] \right) - \psi \left( E[\alpha] + \sum_{s,k} E[z_{sk}] \right), \quad (38)$$

where  $\psi$  is the digamma function. Likewise,  $q(\phi)$  is Dirichlet and is maximized with

$$\phi_j = \frac{\sum_s E[v_{sj} + w_{sj}]}{\sum_{s,i} E[v_{si} + w_{si}]} \quad (39)$$

$E[\alpha]$  is not analytic but requires only a single numerical integral per iteration:

$$E[\alpha] = \frac{\int \alpha q(\alpha) d\alpha}{\int q(\alpha) d\alpha} \quad (40)$$

We therefore have a self-contained iteration scheme in terms of expectations only, Algorithm 2.

---

**Algorithm 2** calling allele counts of ACNV segments

---

- 1: Initialize  $E[\alpha] = 1$
  - 2: Initialize  $(\rho_1, \rho_2, \dots, \rho_K) = (1/K, 2/K, \dots, 1)$
  - 3: Initialize  $E[\ln \pi_j] = \ln(1/K)$  for all  $j$ .
  - 4: Initialize  $\phi$  in some reasonable way, i.e.  $\phi_1 > \phi_2 > \phi_0 > \phi_3$ .
  - 5: **repeat**
  - 6:   Update  $E[z_{sk} v_{sm} w_{sn}]$  via Equation 36.
  - 7:   Update  $E[z_{sk}]$ ,  $E[v_{sj}]$ ,  $E[w_{sj}]$  via Equation 37.
  - 8:   Update  $E[\ln \pi_k]$  via Equation 38
  - 9:   Update  $\phi$  via Equation 39
  - 10:   Update  $E[\alpha]$  via Equation 40
  - 11:   Update  $\rho$  via Equation 35
  - 12: **until** convergence
- 

Once this converges, the main objects of interest are the posterior probabilities of different allele counts,  $P(v_{sm} = 1, w_{sn} = 1) = \sum_k E[z_{sk} v_{sm} w_{sn}]$ . For the purposes of guessing phylogeny the fractions  $\rho_k$  are also interesting.

#### IV. GERMLINE EXOME CNVS

The GATK treats germline CNVs differently from somatic CNVs. This is partly due to fundamental differences, such as the absence of subclones in the germline setting. However, many arbitrary inconsistencies are historic in nature, arising from the germline algorithm’s origins in the XHMM method. It is important to keep this in mind when reading these notes. The two most significant differences between the GATK’s germline and somatic workflows is are the neglect of allelic information (i.e. alt and ref read counts at het sites) in the germline workflow and the use of an HMM for simultaneous segmentation and calling in the germline workflow.

We will treat the HMM as a black box. Although the GATK has its own implementation, the functionality is standard. Thus we will only describe how we define its states, initial probabilities, transition probabilities, and emission distributions. Besides that, it suffices to describe what is done to raw coverage data before it is fed into the HMM.

### A. Normalization of raw germline data

The germline model does not separate the creation of a panel of normals from a case workflow. Rather, it calls CNVs simultaneously for all samples in a cohort. Its starting point is an  $S \times T$  matrix of raw coverage, where  $S$  is the number of samples and  $T$  is the number of targets. We then normalize by each sample's average coverage to get an  $S \times T$  *proportional coverage* matrix  $P$ :

$$P_{st} = \frac{(\text{raw coverage})_{st}}{\text{average depth of sample } s} \quad (41)$$

Next, as in the somatic workflow, we perform principal components analysis (PCA) on the proportional coverage in order to remove noise due to laboratory conditions etc. from the coverage, leaving (we hope) only a CNV signal and a small amount of residual noise. For purely historical reasons PCA is expressed here in slightly different terms from the somatic case. PCA yields a length- $T$  mean proportional coverage vector  $\mu$  and set of  $M$  principal vectors  $\mathbf{v}_k$ , also of length  $T$ , such that the proportional coverage of each sample is approximated by the mean coverage  $\mu$  plus a linear combination of the principal components:

$$P_s \approx \mu + \sum_{k=1}^M \beta_{sk} \mathbf{v}_k \quad (42)$$

Because the principal components capture the shared variation among all samples, we expect them *not* to capture individual variation due to CNVs. There is necessarily some contamination because the samples we call are the same samples used to decide the principal components – there is no separate PoN. Nonetheless, this effect should be small if there are enough samples. Therefore, the next step is to produce the tangent-normalized coverage  $X$ , which is again an  $S \times T$  matrix:

$$X_s = P_s - \mu - \sum_{k=1}^M \beta_{sk} \mathbf{v}_k. \quad (43)$$

(Here a single subscript for a matrix denotes an entire row).

Finally, the tangent-normalized coverage is converted to a Z-score coverage in which each target is mean-centered (tangent-normalization should yield a mean of zero for each target over all samples, so this part is trivial) and divided by the standard deviation of tangent-normalized coverage of that target over all samples:

$$Z_{st} = X_{st} / \text{std}(X_{.t}) \quad (44)$$

The codebase also allows for filtering at each stage of coverage based on target GC and repeat fraction and various coverage descriptive statistics such as mean, standard deviation and interquartile range of targets across samples and vice versa. However, we do not yet have a sense of best practices for these. Furthermore, what constitutes best practices will change as we improve the model.

### B. Germline HMM

Each sample's Z-score coverage is segmented and called separately via the Viterbi algorithm, which finds the maximum-likelihood solution of an HMM. The hidden states are neutral, deletion, and duplication – the XHMM model does not take into account homozygous deletions or multiple duplications.

The HMM's transition matrix is guided by the principle (an approximation, of course) that there is some underlying biological HMM on a *per-base* level and that *per-target* transitions are simply the realization of this underlying HMM on a coarser scale. The per-base transition matrix is defined by two parameters. The first is the probability  $p$  to make a transition from a neutral state to a CNV state. Equivalently,  $1/p$  is, roughly, the average separation between CNVs. The second is the probability  $1/D$  that a CNV state ends. Equivalently,  $D$  is the average CNV length in base pairs. The probability for a CNV to terminate between two consecutive targets a distance  $d$  apart is  $1 - e^{-d/D}$ .

Letting  $f = e^{-1/D}$  the transition matrix  $T$  between two adjacent bases is

$$T = \begin{array}{c|ccc} & \text{from} \backslash \text{to} & - & 0 & + \\ \hline - & f & 1-f & 0 \\ 0 & p & 1-2p & p \\ + & 0 & 1-f & f \end{array} \quad (45)$$

We neglect transitions between different types of CNVs at consecutive bases, which are extremely rare. Note that this in no way precludes CNVs of different types occurring at adjacent targets. The transition matrix for two targets separated by  $d$  bases is  $T^d$ . We can compute this very cheaply by first diagonalizing  $T$  as  $T = U\Sigma V^T$ , where  $\Lambda$  is a diagonal matrix. Then  $T^d = U^T \Lambda^d U$ . For numerical stability one usually works with log transition probabilities, so we have:

$$\log (T^d)_{ij} = \log \sum_k U_i^T k \Lambda_{kk}^d U_{kj} \quad (46)$$

$$= \log \sum_k \Lambda_{kk}^d U_{kj} U_{ki} \quad (47)$$

$$= \log \sum_k \exp (d \log \Lambda_{kk} + \log U_{kj} + \log U_{ki}) \quad (48)$$

In this form we can work entirely in log space and exploit the log-sum-exp trick for stability.

The emission model is as follows. Each hidden state emits a normally-distributed Z-score. The means are  $-M$ ,  $0$ , and  $+M$  for deletion, neutral, and duplication states, respectively, where  $M$  is a user-specified parameter whose default is 3. Each emission distribution is given unit variance. This model is quite wrong. Consider a duplication. The tangent-normalized coverage ought to be roughly 0.5 times the proportional coverage – the raw coverage is  $3/2$  that of a diploid target, leaving  $1/2$  remaining after (ideal) tangent-normalization. Then division by the target standard deviation to get a Z-score yields who-knows-what. Since different targets have different average proportional coverage, the global parameter  $M$  is misguided. Basically, the current model is not a model at all, but a heuristic.

## V. PROPOSED METHODS

### A. Using Panel of Normals for Allelic Fraction Model

The GATK ACNV allelic model learns a global distribution on allelic biases and uses it as a shared prior for the allelic biases of SNPs. While better than nothing, it would be much more powerful to use prior knowledge of the allelic bias at each SNP individually. We can learn these per-SNP biases from a panel of normals using the allelic model, but with two simplifications. First, minor allele fractions are always  $1/2$  since normal samples are diploid and do not exhibit subclonality. Second, we do not account for outliers; that is, we set the outlier probability  $\pi = 0$ . The reason for this is that the panel of normals reflects typical distributions of allelic biases and censoring data via an outlier classification could render these distributions artificially tight. If the allelic bias at some SNP site varies a lot we want to know about it. The overall likelihood is

$$\prod_j \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_j^{\alpha-1} e^{-\beta \lambda_j} \prod_{s \in \mathcal{H}_j} \frac{\lambda_j^{r_{sj}}}{(1 + \lambda_j)^{n_{sj}}} \quad (49)$$

$$= \prod_j \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta \lambda_j} \frac{\lambda_j^{\alpha+r_{\cdot j}-1}}{(1 + \lambda_j)^{n_{\cdot j}}} \quad (50)$$

where  $\lambda_j$  is the allelic bias ratio of SNP  $j$  (for samples sequenced and mapped using the same technology as the panel of normals),  $\mathcal{H}_j$  is the set of samples in the panel of normals that are heterozygous at SNP  $j$ ,  $r_{\cdot j} = \sum_{s \in \mathcal{H}_j} r_{sj}$ , and  $n_{\cdot j} = \sum_{s \in \mathcal{H}_j} n_{sj}$ . As before, the biases are assumed to come from a common distribution  $\text{Gamma}(\alpha, \beta)$ , but due to the large number of samples in the panel of normals the data will yield a posterior distribution on each  $\lambda_j$  that may be quite different from the global prior. It is these posteriors that we will use as input to ACNV. Although they are the object of interest, however, we will first marginalize them out of the likelihood in order to obtain maximum likelihood estimates of  $\alpha$  and  $\beta$ . We have in fact already performed this marginalization – Equation 50 is the special case  $f = 1/2$ ,  $\pi = 0$  of the allelic-model likelihood, Equation 16, and thus its marginalization over latent variables is obtained by substituting  $f = 1/2$ ,  $\pi = 0$  into Equation 22, which yields

$$p(\alpha, \beta) = \prod_j \phi(\alpha, \beta, f = 1/2, n_{\cdot j} - r_{\cdot j}, r_{\cdot j}). \quad (51)$$

This likelihood is easily maximized numerically to obtain MLE values of  $\alpha$  and  $\beta$ . Having done this, we can then approximate the posterior on each  $\lambda_j$  as a gamma distribution using the method of Appendix A. As shown there, the posterior on  $\lambda_j$  is  $\text{Gamma}(\rho_j, \tau_j)$  where  $\rho_j$  and  $\tau_j$  are computed in Algorithm 4, with  $a \rightarrow n_{\cdot j} - r_{\cdot j}$  and  $r \rightarrow r_{\cdot j}$ .

Once we have the posteriors on each  $\lambda_j$  from the panel of normals, they are used as priors for  $\lambda_j$  in the ACNV allelic model. This obviates the hyperparameters  $\alpha$  and  $\beta$ , and Equation 22 becomes

$$p(f, \pi) \propto \prod_j \left[ \frac{1-\pi}{2} \phi(\rho_j, \tau_j, f_j, a_j, r_j) + \frac{1-\pi}{2} \phi(\rho_j, \tau_j, 1-f_j, a_j, r_j) + \frac{\pi a_j! r_j!}{(n_j+1)!} \right] \quad (52)$$

where  $f$  and  $\pi$  may once again be sampled via adaptive Metropolis.

## B. Generative Model for Read Counts

In this section we address several goals:

1. Connect copy ratio (or copy number) and raw read counts in a single probabilistic model without transforming the data.
2. Take into account the Poisson nature of coverage depth, thereby giving less weight to low-coverage targets and separating the inherent variance due to Poisson statistics from experimental noise. We want to use the panel of normals to subtract only the latter.
3. Choose the number of principal components to use in an automatic and principled manner.
4. Use an algorithm that does not waste time calculating all principal components when we only want the few most significant ones.
5. Correct for CNV events that occur in the panel of normals.

Suppose we have vectors of read counts over a set of  $T$  targets for  $S$  samples,  $\mathbf{n}_s$ ,  $s = 1 \dots S$  where  $n_{st}$  is the coverage of sample  $s$  at target  $t$ . We imagine that laboratory conditions for a particular sample yielding an underlying bias vector  $\mathbf{b}_s$ , where  $e^{b_{st}}$  is the propensity of target  $t$  to be captured, sequenced, and mapped in the preparation of sample  $s$ . Suppose also that sample  $s$  has an average depth  $d_s$  and a vector of copy numbers  $\mathbf{c}_s$ , where the latent variable  $c_{st}$  is the copy number of sample  $s$  at target  $t$ . Our model for coverage is<sup>10</sup>:

$$n_{st} \sim \text{Poisson}(d_s c_{st} e^{b_{st}}) \quad (53)$$

We can achieve many of the goals listed above by performing probabilistic PCA not on  $\mathbf{n}$  directly but rather on  $\mathbf{b}$ . This is one reason why we choose a Poisson parameter proportional to  $e^b$  and not simply  $b$  – probabilistic PCA involves a normal distribution on  $\mathbf{b}_s$ , whereas the Poisson parameter must be positive. Furthermore, since PCA is a linear model with additive effects of different latent features, using  $e^b$  represents multiplicative effects, which is more plausible. Our model is

$$\mathbf{z} \sim \text{N}(\mathbf{0}, \mathbf{I}) \quad (54)$$

$$\mathbf{b} \sim \text{N}(\mathbf{W}\mathbf{z} + \mathbf{m}, \mathbf{\Psi}), \quad (55)$$

where  $\mathbf{z} \in \mathbb{R}^D$  is a low-dimensional latent vector of laboratory conditions,  $\mathbf{W} \in \mathbb{R}^{T \times D}$  is a linear map from latent space to target space,  $\mathbf{m} \in \mathbb{R}^T$  is the vector of mean biases, and  $\mathbf{\Psi} \in \mathbb{R}^{T \times T}$  is a diagonal matrix of residual variance not explained by the latent features. We approximate the Poisson as a Gaussian centered at its mode (as a function of  $b_{st}$ ):

$$\prod_t \text{Poisson}(n_{st} | d_s c_{st} e^{b_{st}}) \approx \text{N}(\mathbf{b}_s | \mathbf{m}_s, \mathbf{\Sigma}_s), \quad m_{st} = \ln(n_{st} / (c_{st} d_s)), \quad \Sigma_{st} = 1/n_{st} \quad (56)$$

to obtain the joint likelihood

$$P(\mathbf{n}, \mathbf{b}, \mathbf{z} | \mathbf{c}, \mathbf{W}, \mathbf{m}, \mathbf{\Psi}, \mathbf{d}) = \prod_s \text{N}(\mathbf{z}_s | \mathbf{0}, \mathbf{I}) \text{N}(\mathbf{b}_s | \mathbf{W}\mathbf{z}_s + \mathbf{m}, \mathbf{\Psi}) \text{N}(\mathbf{b}_s | \mathbf{m}_s, \mathbf{\Sigma}_s) \quad (57)$$

---

<sup>10</sup> In Equation 53 the quantity  $d_s$  is a hypothetical quantity representing average coverage in the absence of bias. Since this is impossible to know we use average coverage instead. This doesn't matter since any constant factor will be absorbed into  $e^b$  via the parameter  $\mathbf{m}$ .



We can integrate out  $\mathbf{b}$  via the identity  $\int N(x|\mu_1, \sigma_1^2)N(x|\mu_2, \sigma_2^2) dx = N(\mu_1|\mu_2, \sigma_1^2 + \sigma_2^2)$  to obtain

$$P(\mathbf{n}, \mathbf{z}|\mathbf{c}, \mathbf{W}, \mathbf{m}, \Psi, \mathbf{d}, \mathbf{n}) = \prod_s N(\mathbf{z}_s|\mathbf{0}, \mathbf{I})N(\mathbf{m}_s|\mathbf{W}\mathbf{z}_s + \mathbf{m}, \Psi + \Sigma_s) \quad (58)$$

While we will learn  $\Psi$  and  $\mathbf{W}$  via the EM algorithm, we may determine  $\mathbf{m}$  by integrating out  $\mathbf{z}$  followed by direct maximization. We marginalize  $\mathbf{z}_s$  from Equation 58 using a standard result for linear Gaussian models (Bishop Eq. 2.115) to obtain

$$P(\mathbf{n}|\mathbf{c}, \mathbf{W}, \mathbf{m}, \Psi, \mathbf{d}, \mathbf{n}) = \prod_s N(\mathbf{m}_s|\mathbf{m}, \Psi + \Sigma_s + \mathbf{W}\mathbf{W}^T) \quad (59)$$

$$= \exp\left(-\frac{1}{2} \sum_s (\mathbf{m} - \mathbf{m}_s)^T (\Psi + \Sigma_s + \mathbf{W}\mathbf{W}^T)^{-1} (\mathbf{m} - \mathbf{m}_s)\right) \quad (60)$$

If  $\Sigma_s$  were the same for each  $s$ , the likelihood in Equation 60 would be maximized by  $\mathbf{m} = \frac{1}{S} \sum_s \mathbf{m}_s$ . As we now argue, this intuitively-appealing approximate is accurate and justified. First, note that  $\Psi + \mathbf{W}\mathbf{W}^T$  is the covariance of log-coverage due to experimental conditions, both those included explicitly in  $\mathbf{W}$  and the residual, unexplained covariance  $\Psi$ . The quantities  $\Psi + \Sigma_s + \mathbf{W}\mathbf{W}^T$  are essentially weighting factors that penalize lower-coverage samples (with larger  $\Sigma$ ). If variations in  $\Sigma$  tend to be small compared to  $\Psi + \mathbf{W}\mathbf{W}^T$ , then these weights will be very similar between samples and we may use the simple formula that assigns each sample the same weight. It is an empirical fact that the noise before normalization is much greater than noise after normalization, which implies that  $\Sigma$  is small compared to  $\Psi + \mathbf{W}\mathbf{W}^T$ . Variations in  $\Sigma$  are, of course, even smaller, especially in the typical case that all PoN samples have similar depth. Furthermore, if the depths of PoN samples are not correlated with their positions in latent space (as is reasonable) our approximation is an unbiased estimator, because for any given position in latent space our failure to use the exact weights will on average wash out. Finally, suppose, contrary to empirical fact, that  $\Sigma$  tended to be large compared to  $\Psi + \mathbf{W}\mathbf{W}^T$ . In this case either laboratory noise is small and we don't need this fancy model in the first place or PoN depth of coverage is very poor, in which case accurate normalization is hopeless<sup>11</sup>. Thus we will use

$$\mathbf{m} = \frac{1}{S} \sum_s \mathbf{m}_s \quad (\text{an excellent approximation}) \quad (61)$$

The approximation that led to this can be interpreted as the replacement  $\Sigma_s \rightarrow \bar{\Sigma} \equiv \frac{1}{S} \sum_s \Sigma_s$  and we will consider it again below when we discuss EM updates of  $\mathbf{W}$  and  $\Psi$ .

For fixed  $\mathbf{c}$  we may obtain maximum likelihood estimates of  $\mathbf{W}$  and  $\Sigma$  by an iterative EM approach in which we alternate between computing the Gaussian posteriors of each  $\mathbf{z}_s$  (E step) and maximizing the log-likelihood with respect to  $\mathbf{W}$ ,  $\Sigma$ , and  $\mathbf{m}$  (M step). This is similar to the factor analysis model discussed in Chapter 12 of Bishop. We note calling  $\mathbf{c}$  at each iteration allows us to correct for CNV events that may be present in the panel of normals<sup>12</sup>.

The E step follows from substituting  $\Psi \rightarrow \Psi_s \equiv \Psi + \Sigma_s$  in Bishop's Equations 12.66-67 to get

$$\mathbf{G}_s = (\mathbf{I} + \mathbf{W}^T \Psi_s^{-1} \mathbf{W})^{-1} \quad (62)$$

$$E[\mathbf{z}_s] = \mathbf{G}_s \mathbf{W}^T \Psi_s^{-1} (\mathbf{m}_s - \mathbf{m}) \quad (63)$$

$$E[\mathbf{z}_s \mathbf{z}_s^T] = \mathbf{G}_s + E[\mathbf{z}_s] E[\mathbf{z}_s]^T \quad (64)$$

For the M steps, we must maximize with respect to  $E[\sum_s \ln N(\mathbf{m}_s|\mathbf{W}\mathbf{z}_s + \mathbf{m}, \Psi + \Sigma_s)]$ , which reduces to

$$-\frac{1}{2} \sum_{st} [\ln(\Psi_t + \Sigma_{st}) + ((\mathbf{W}E[\mathbf{z}_s \mathbf{z}_s^T] \mathbf{W}^T)_{tt} + 2(m_t - m_{st})(\mathbf{W}E[\mathbf{z}_s])_t + (m_t - m_{st})^2) \Psi_{st}^{-1}] \quad (65)$$

<sup>11</sup> We have also derived exact EM updates for  $\mathbf{m}$  and compared simulated results to the approximation. The difference is basically non-existent.

<sup>12</sup> If our copy-ratio calling algorithm is a maximum-likelihood method, for example calling via the Viterbi algorithm for an HMM, then provided that our likelihood is averaged with respect to the E step for  $\mathbf{z}$  the entire process including optimization of  $\mathbf{c}$ ,  $\mathbf{m}$ ,  $\mathbf{W}$ , and  $\Psi$  is an EM algorithm and thus will increase likelihood at each iteration and converge.

If we do not use the approximation for  $\mathbf{m}$ , its M step is

$$\mathbf{m} = \left( \sum_s \Psi_s^{-1} \right)^{-1} \sum_s [\Psi_s^{-1} (\mathbf{m}_s - \mathbf{W} E[\mathbf{z}_s])] . \quad (\text{exact}) \quad (66)$$

The M step for  $\Psi_t$  requires solving numerically for each  $\Psi_t$  the following equations:

$$\sum_s \left[ \frac{1}{\Psi_t + \Sigma_{st}} - \frac{\beta_{st}}{(\Psi_t + \Sigma_{st})^2} \right] = 0, \beta_{st} = (\mathbf{W} E[\mathbf{z}_s \mathbf{z}_s^T] \mathbf{W}^T)_{tt} + 2(m_t - m_{st}) (\mathbf{W} E[\mathbf{z}_s])_t + (m_t - m_{st})^2 \quad (67)$$

Alternatively, the aforementioned approximation  $\Sigma_s \rightarrow \bar{\Sigma}$  gives

$$\Psi_{tt} = \frac{1}{S} \sum_s \beta_{st} - \bar{\Sigma}_{tt}, \text{ (simplified)} \quad (68)$$

In the M step equation for  $\mathbf{W}$  we incorporate an automatic relevance determination (ARD) prior:

$$P(\mathbf{W}) = \prod_k \left( \frac{\alpha_\mu}{2\pi} \right)^{T/2} \exp \left( -\frac{1}{2} \alpha_\mu \sum_t W_{t\mu}^2 \right). \quad (69)$$

If  $\alpha_\mu \rightarrow \infty$  latent feature  $\mu$  is effectively turned off. Thus we can initially choose a liberal estimate of  $D$  and the model will automatically become more parsimonious. The M step log likelihood times the ARD prior depend on the  $t$ -th row  $\mathbf{W}_{t\cdot}$  of  $\mathbf{W}$  as

$$-\frac{1}{2} \left( -\sum_\mu \ln \alpha_\mu + \mathbf{W}_{t\cdot} (\mathbf{A} + \mathbf{Q}_t) \mathbf{W}_{t\cdot}^T - 2\mathbf{W}_{t\cdot} \mathbf{v}_t \right) \quad (70)$$

where  $\mathbf{A} \equiv \text{Diag}(\alpha_1, \alpha_2 \dots \alpha_D)$  and we have defined

$$\mathbf{Q}_t = \sum_s \Psi_{st}^{-1} E[\mathbf{z}_s \mathbf{z}_s^T], \mathbf{v}_t = \sum_s \Psi_{st}^{-1} (m_{st} - m_t) E[\mathbf{z}_s] \quad (71)$$

The maximum a posterior result for  $\mathbf{W}_{t\cdot}$  is

$$\mathbf{W}_{t\cdot} = (\mathbf{A} + \mathbf{Q}_t)^{-1} \mathbf{v}_t \quad (72)$$

In the approximation  $\Psi_s \rightarrow \bar{\Psi}$  this formula is unchanged but  $\mathbf{Q}_t$  is  $S$  times as fast to calculate. The other M steps and the E step are not affected by the ARD prior.

To determine  $\alpha_\mu$ , we re-exponentiate Equation 70<sup>13</sup> and integrate out  $\mathbf{W}$  to obtain the evidence for  $\mathbf{A}$ :

$$P(\mathbf{n}|\mathbf{A}) \propto \prod_k \alpha_k^{T/2} \prod_t \int e^{-\frac{1}{2} \mathbf{W}_{t\cdot} (\mathbf{A} + \mathbf{Q}_t) \mathbf{W}_{t\cdot}^T - \mathbf{W}_{t\cdot} \mathbf{v}_t} d\mathbf{W}_{t\cdot} \quad (73)$$

The ARD coefficients  $\alpha$  are determined by maximizing the log evidence. That is, we set

$$\frac{\partial}{\partial \alpha_k} \ln P(\mathbf{n}|\mathbf{A}) = 0 \Rightarrow \frac{1}{2} \left( \frac{T}{\alpha_\mu} - \sum_t \langle W_{t\mu}^2 \rangle \right) = 0 \Rightarrow \alpha_\mu = \frac{T}{\sum_t \langle W_{t\mu}^2 \rangle}, \quad (74)$$

where the average  $\langle W_{t\mu}^2 \rangle$  is taken with respect to the density  $q(\mathbf{W}_{t\cdot}) \propto e^{-\frac{1}{2} \mathbf{W}_{t\cdot} (\mathbf{A} + \mathbf{Q}_t) \mathbf{W}_{t\cdot}^T - \mathbf{W}_{t\cdot} \mathbf{v}_t}$ . Completing the square, we find that  $q(\mathbf{W}_{t\cdot})$  is Gaussian with covariance  $(\mathbf{A} + \mathbf{Q}_t)^{-1}$  and mean  $(\mathbf{A} + \mathbf{Q}_t)^{-1} \mathbf{v}_t$ . Note that this mean is precisely the M step value for  $q(\mathbf{W}_{t\cdot})$ , as we would hope! Thus we get

$$\langle W_{t\mu}^2 \rangle = W_{t\mu}^2 + (\mathbf{A} + \mathbf{Q}_t)_{\mu\mu}^{-1} \quad (75)$$

Let us now summarize these steps. First, in exact mode we have

---

<sup>13</sup> This is the distribution on  $\mathbf{W}$  that we would obtain from a mean-field variational factorization  $q(\mathbf{z})q(\mathbf{W})$ .

- E step:  $\mathbf{G}_s = (\mathbf{I} + \mathbf{W}^T \Psi_s^{-1} \mathbf{W})^{-1}$ ,  $E[\mathbf{z}_s] = \mathbf{G}_s \mathbf{W}^T \Psi_s^{-1} (\mathbf{m}_s - \mathbf{m})$ ,  $E[\mathbf{z}_s \mathbf{z}_s^T] = \mathbf{G}_s + E[\mathbf{z}_s] E[\mathbf{z}_s]^T$   
 $\mathbf{G}_s$  and all the  $E[\mathbf{z}_s]$  are each  $O(D^2 TS)$ .  $E[\mathbf{z}_s \mathbf{z}_s^T]$  is  $O(D^2 S)$ . The E step overall is  $O(D^2 TS)$ .
- $\mathbf{m} = (\sum_s \Psi_s^{-1})^{-1} \sum_s [\Psi_s^{-1} (\mathbf{m}_s - \mathbf{W} E[\mathbf{z}_s])]$  is  $O(DTS)$ .
- $\beta_{st} = (\mathbf{W} E[\mathbf{z}_s \mathbf{z}_s^T] \mathbf{W}^T)_{tt} + 2(m_t - m_{st}) (\mathbf{W} E[\mathbf{z}_s])_t + (m_t - m_{st})^2$  is  $O(D^2 TS)$
- Solving  $\sum_s \left[ \frac{1}{\Psi_t + \Sigma_{st}} - \frac{\beta_{st}}{(\Psi_t + \Sigma_{st})^2} \right] = 0$  for each  $t$  is  $O(TS)$  with a prefactor equal to the number of evaluations required to find a root. As long as this number is less than  $D^2$  this step is subleading.
- $\mathbf{Q}_t = \sum_s \Psi_{st}^{-1} E[\mathbf{z}_s \mathbf{z}_s^T]$  is  $O(D^2 TS)$ .
- $\mathbf{v}_t = \sum_s \Psi_{st}^{-1} (m_{st} - m_t) E[\mathbf{z}_s]$  is  $O(DTS)$ .
- $\mathbf{W}_{t\cdot} = (\mathbf{A} + \mathbf{Q}_t)^{-1} \mathbf{v}_t$  is  $O(D^3 T)$ .
- $\langle W_{t\mu}^2 \rangle = W_{t\mu}^2 + (\mathbf{A} + \mathbf{Q}_t)_{\mu\mu}^{-1}$  is  $O(D^3 T)$ .
- $\alpha_\mu = T / \sum_t \langle W_{t\mu}^2 \rangle$  is  $O(DT)$ .

In approximate mode

- E step:  $\mathbf{G} = (\mathbf{I} + \mathbf{W}^T \bar{\Psi}^{-1} \mathbf{W})^{-1}$ ,  $E[\mathbf{z}_s] = \mathbf{G} \mathbf{W}^T \bar{\Psi}^{-1} (\mathbf{m}_s - \mathbf{m})$ ,  $E[\mathbf{z}_s \mathbf{z}_s^T] = \mathbf{G} + E[\mathbf{z}_s] E[\mathbf{z}_s]^T$   
 $\mathbf{G}$  is  $O(D^2 T)$ .  $E[\mathbf{z}_s \mathbf{z}_s^T]$  is  $O(D^2 S)$ .  $\mathbf{G}_s \mathbf{W}^T \Psi_s^{-1}$ , which is common to all  $E[\mathbf{z}_s]$ , is  $O(D^2 T)$  after which  $E[\mathbf{z}_s]$  is  $O(DTS)$ .  $E[\mathbf{z}_s \mathbf{z}_s^T]$  is  $O(D^2 S)$ . The E step overall is  $O(DTS)$ .
- $\mathbf{m} = \frac{1}{S} \sum_s \mathbf{m}_s$  is a one-time cost of  $O(TS)$ .
- $\sum_s \beta_{st} = (\mathbf{W} (\sum_s E[\mathbf{z}_s \mathbf{z}_s^T]) \mathbf{W}^T)_{tt} + 2(\mathbf{W} (\sum_s (m_t - m_{st}) E[\mathbf{z}_s]))_t + \sum_s (m_t - m_{st})^2$  is  $O(D^2 T)$ .
- Given  $\sum_s \beta_{st}$ ,  $\Psi_{tt} = \frac{1}{S} \sum_s \beta_{st} - \bar{\Sigma}_{tt}$  is  $O(T)$ .
- $\mathbf{Q}_t = \bar{\Psi}_t^{-1} \sum_s E[\mathbf{z}_s \mathbf{z}_s^T]$  is  $O(D^2(S + T))$ .
- $\mathbf{v}_t = \sum_s \bar{\Psi}_t^{-1} (m_t - m_{st}) E[\mathbf{z}_s]$  is  $O(DTS)$ .
- $\mathbf{W}_{t\cdot} = (\mathbf{A} + \mathbf{Q})^{-1} \mathbf{v}_t$  is  $O(D^2 T)$ .
- $\langle W_{t\mu}^2 \rangle = W_{t\mu}^2 + (\mathbf{A} + \mathbf{Q})_{\mu\mu}^{-1}$  is  $O(DT)$ .
- $\alpha_\mu = T / \sum_t \langle W_{t\mu}^2 \rangle$  is  $O(DT)$ .

In exact mode the leading cost is a few terms of  $O(D^2 TS)$  flops, each with small prefactors, per iteration. In approximate mode it is  $O(DTS)$  flops. Assuming a total prefactor of 10 and  $T = 2 \times 10^5$ ,  $D = 10$ ,  $S = 500$  a full EM iteration costs  $10^{11}$  flops in exact mode and  $10^{10}$  in approximate mode. On a single 1 GHz core ( $10^9$  flops per second) this comes out to roughly 100 or 10 seconds. Even in exact mode we will probably get most of the way toward convergence with approximate iterations, followed by several exact iterations. Very conservatively, 100 approximate iterations and 10 exact ones should suffice, which comes out to less than half an hour.

We can apply the parameters learned from the panel of normals to single-sample calling, which requires the likelihood as a function of the copy numbers  $\mathbf{c}$ . Applying the same E step as above, the likelihood is

$$P(\text{data}|\mathbf{c}) \propto \prod_t \exp \left[ -\frac{1}{2} ((\mathbf{m}_t - \mathbf{m}_{st})^2 + 2(\mathbf{m}_t - \mathbf{m}_{st}) \mathbf{W}_t E[\mathbf{z}_s]) \Psi_{st}^{-1} \right], \quad (76)$$

which depends on  $\mathbf{c}$  via  $m_{st} = \ln(n_{st}/(c_{st} d_s))$ . Note that this is factorized into independent likelihood terms for each target and is thus suitable for the emission model of an HMM. This likelihood is not Gaussian in  $\mathbf{c}$ , but it does not need to be for the Viterbi and forward-backward algorithms.

We can easily integrate sample-specific GC bias into this model. Let  $f_s(\text{GC})$  be the GC bias of GC content GC for sample  $s$ . Then this enters into the model as an additional multiplier to the Poisson parameter. That is, we replace  $d_s c_{st} \rightarrow d_s c_{st} f_s(\text{GC}_t)$ , which affects the model learning and inference only via the definition of  $m_{st}$ . We can

iteratively re-estimate the GC bias function  $f_s$  by regressing the bias not explained by the latent factors. That is, for each target the Poisson parameter is, ignoring GC effects,

$$c_{st}d_s \exp(\mathbf{W}_t E[\mathbf{z}_s] + m_t) \quad (77)$$

and thus the ratio  $n_{st}/[c_{st}d_s \exp(\mathbf{W}_t E[\mathbf{z}_s] + m_t)]$  (with error bars of size  $1/\sqrt{n}$  if we want to do a weighted regression) is an estimate of  $f_s(\text{GC}_t)$  that we can feed into our favorite regression model. This is more sophisticated than the standard approach of simply regressing  $n$  versus GC in that it seeks to explain with GC only the bias that cannot be explained by linear latent features.

### C. HMM-based segmentation of somatic CNVs

In CNV segmentation, the hidden states are defined at genomic loci – targets or SNPs. The transition matrix of an HMM for segmentation will be a function of the distances between consecutive loci, similar to what already exists in our germline code. The emission likelihoods of the HMM will be given by probabilistic models for allele counts at het sites (which we already have) and total coverage of targets (another proposed method). Broadly speaking, all that remains is to define the hidden states and to tweak the transition model for the somatic case. For concreteness we will begin the discussion in terms of het allele fraction segmentation.

We already have a probabilistic model  $P(a_i, r_i | f_i)$ , where  $a_i$  and  $r_i$  are observed alt and ref allele counts at het site  $i$  and  $f_i$  is the underlying minor allele fraction of the segment to which this het belongs. This is the emission likelihood of our HMM. The hidden states are therefore a discrete set of minor allele fractions  $f$ . The discreteness of these states follows from the discreteness of the tumor's phylogeny. We learn a finite set of minor allele fractions via a Chinese Restaurant Process to obtain a set of  $K$  hidden states. A simple transition model assumes a length scale  $D$ , so that over  $d$  bases the CNV state is unchanged with probability  $e^{-d/D}$ , and with probability  $1 - e^{-d/D}$  a new CNV state is chosen with uniform probability. Thus every site  $n$  has an associated binary indicator  $s_n$  with prior probability  $P(s_n = 0) = e^{-d_n/D}$ , where  $d_n$  is the distance from the previous het to the  $n$ th het. This corresponds to the following transition probabilities for the  $n$ th site:

$$P(f_j \rightarrow f_k) = \left[ e^{-d_n/D} \delta_{jk} \right]^{1-s_n} \left[ (1 - e^{-d_n/D})/K \right]^{s_n} \quad (78)$$

Letting  $z_{nk}$  be a binary indicator for hidden state  $f_k$  at het site  $n$ , the model likelihood is

$$\left( \prod_{nk} P(a_n, r_n | f_k)^{z_{nk}} \right) \left( \prod_{nj} \left( \left[ e^{-d_n/D} \delta_{jk} \right]^{1-s_n} \left[ (1 - e^{-d_n/D})/K \right]^{s_n} \right)^{z_{(n-1)j} z_{nk}} \right) \quad (79)$$

Supposing that we can obtain the posterior joint probability of the hidden states  $s$  and  $z$ , we can take the logarithm of Equation 79 and read off the M step equations. The M step for  $D$  is

$$D = \operatorname{argmax}_D \sum_n \left( E[s_n] \ln(1 - e^{-d_n/D}) - \frac{d_n}{D} E[(1 - s_n)] \right) \quad (80)$$

The M step for  $f_k$  is

$$f_k = \operatorname{argmax}_{f_k} \sum_{nk} E[z_{nk}] \ln P(a_n, r_n | f_k) \quad (81)$$

The above model contains a hidden chain of states  $z_n$  along with an additional hidden node  $s_n$  for each  $n$  that is a parent of  $z_n$  in the DAG<sup>14</sup>. This model cannot be fed into a black-box HMM due to the extra nodes. It is trivial to marginalize out the  $s_n$  to obtain a simple HMM with transition matrix

$$P(f_j \rightarrow f_k) = e^{-d_n/D} \delta_{jk} + (1 - e^{-d_n/D})/K, \quad (82)$$

<sup>14</sup> Note that the  $s_n$  have no links with each other and are not a chain.

however using such a model the forward-backward algorithm will only output the E step posteriors  $E[z_{nk}]$  and  $E[z_{(n-1)j}z_{nk}]$ , while we need  $E[s_n z_{(n-1)j}z_{nk}]$  and  $E[s_n]$ . Fortunately, these are actually easy to obtain from Bayes' Rule.

$$E[s_n z_{(n-1)j}z_{nk}] = P(s_n = 1, z_{(n-1)j} = 1, z_{nk} = 1) \quad (83)$$

$$= P(z_{(n-1)j} = 1, z_{nk} = 1) P(s_n = 1 | z_{(n-1)j} = 1, z_{nk} = 1) \quad (84)$$

$$= E[z_{(n-1)j}z_{nk}] \frac{P(z_{nk} = 1 | z_{(n-1)j} = 1, s_n = 1) P(s_n = 1)}{P(z_{nk} = 1 | z_{(n-1)j} = 1)} \quad (85)$$

$$= E[z_{(n-1)j}z_{nk}] \frac{(1 - e^{-d_n/D})/K}{e^{-d_n/D}\delta_{jk} + (1 - e^{-d_n/D})/K}, \quad (86)$$

after which we easily obtain  $E[s_n] = \sum_{jk} E[s_n z_{(n-1)j}z_{nk}]$  and  $E[1 - s_n] = 1 - E[s_n]$ . Thus we can perform the desired E step using only the forward-backward algorithm of the marginalized model.

The M step amounts to  $K + 1$  numerical optimizations (for  $D$  and  $\{f_k\}$ ) of objectives containing  $T$  terms each along with several very cheap operations. The E step's time complexity is the cost  $O(K^2T)$  of the forward-backward algorithm. We note that since the allele-fraction model requires segmentation information, during EM iteration we should also perform segmentations via the Viterbi algorithm and relearn the parameters of the allele-fraction model. The Viterbi algorithm also has cost  $O(K^2T)$ . The steps described above are summarized in Algorithm 3.

---

**Algorithm 3** HMM segmentation of somatic allele fraction

---

- 1: Initialize  $D$  with a reasonable guess of an average CNV size.
  - 2: Initialize  $(f_1, f_2, \dots, f_K) = (1/(2K), 2/(2K), \dots, 1/2)$
  - 3: Initialize  $\bar{\pi}_k = 1/K$  for all  $k$ .
  - 4: **repeat**
  - 5:   Run the forward-backward algorithm on the HMM given by transitions Eq. 82 to obtain  $E[z_{nk}]$  and  $E[z_{(n-1)j}z_{nk}]$ .
  - 6:   Calculate  $E[s_n z_{(n-1)j}z_{nk}]$  from Eq. 86.
  - 7:   Calculate  $E[s_n] = \sum_{jk} E[s_n z_{(n-1)j}z_{nk}]$  and  $E[1 - s_n] = 1 - E[s_n]$ .
  - 8:   Update  $D$  via Eq. 80.
  - 9:   Update  $f_k$  via Eq. 81.
  - 10:   Calculate  $N_k$  via Eq. ??.
  - 11:   Calculate  $\bar{\pi}_k$  via Eq. ??.
  - 12:   Calculate  $\bar{\alpha}$  from Eq. ??.
  - 13: **until** convergence
- 

The framework we described above works equally well for hidden copy ratio states, provided we have a generative model for copy ratio with an associated likelihood. Furthermore, by using two-dimensional hidden states containing minor allele fraction and copy ratio we may segment both simultaneously. The HMM machinery has no difficulty with heterogeneous emission likelihoods.

## Appendix A: Marginalizing out latent variables of the allelic model

We wish to evaluate

$$\phi(\alpha, \beta, f, a, r) = \int_0^\infty g(\lambda, \alpha, \beta, f, a, r) d\lambda \quad (A1)$$

where

$$g(\lambda, \alpha, \beta, f, a, r) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{f_j^a (1-f)^r \lambda^{\alpha+r-1} e^{-\beta\lambda}}{(f + (1-f)\lambda)^{a+r}} \quad (A2)$$

An extremely good approximation for all values of  $f$ ,  $\alpha$ ,  $\beta$ , and  $a$ ,  $r$  is

$$g(\lambda, \alpha, \beta, f, a, r) \approx c \lambda^{\rho-1} e^{-\tau\lambda}. \quad (A3)$$

where  $\rho$  and  $\tau$  are chosen to reproduce the mode of  $g(\lambda, \alpha, \beta, f, a, r)$  and the curvature at its mode. Having approximated our integrand as a gamma distribution's pdf on  $\lambda$ , we integrate it analytically

$$\phi(\alpha, \beta, f, a, r) = c \int_0^\infty \lambda^{\rho-1} e^{-\tau\lambda} d\lambda = c \frac{\Gamma(\rho)}{\tau^\rho} \quad (A4)$$

The mode  $\lambda_0$  is found by setting logarithmic derivatives to zero:

$$\frac{d}{d\lambda} [(\alpha + r - 1) \ln \lambda - \beta \lambda - n \ln (f + (1 - f)\lambda)]_{\lambda_0} = 0 \quad (\text{A5})$$

$$\frac{\alpha + r - 1}{\lambda_0} - \beta - \frac{n(1 - f)}{f + (1 - f)\lambda_0} = 0 \quad (\text{A6})$$

Multiplying out the denominators yields a quadratic equation. Taking the positive root gives

$$\lambda_0 = \frac{\sqrt{w^2 + 4\beta f(1 - f)(r + \alpha - 1 - w)}}{2\beta(1 - f)}, \quad w = (1 - f)(a - \alpha + 1) + \beta f. \quad (\text{A7})$$

The second derivative of  $\ln f$  at  $\lambda_0$  is

$$\kappa = -\frac{r + \alpha - 1}{\lambda_0^2} + \frac{n(1 - f)^2}{(f + (1 - f)\lambda_0)^2} \quad (\text{A8})$$

The mode of the approximating gamma distribution is  $(\rho - 1)/\tau$  and the log second derivative is  $-(\rho - 1)/\lambda_0^2$ . Equating these, we obtain

$$\rho = 1 - \kappa\lambda_0^2, \quad \tau = -\kappa\lambda_0 \quad (\text{A9})$$

Finally, we choose  $c$  so that the values of  $\ln f$  and the approximation match at  $\lambda_0$ :

$$\ln c = \alpha \ln \beta - \ln \Gamma(\alpha) + a \ln f + r \ln(1 - f) + (r + \alpha - \rho) \ln \lambda_0 + (\tau - \beta)\lambda_0 - n \ln (f + (1 - f)\lambda_0) \quad (\text{A10})$$

Algorithm 4 shows the entire computation.

---

**Algorithm 4** Calculating  $\phi(\alpha, \beta, f, a, r)$

---

```

1:  $n = a + r$ 
2:  $w = (1 - f)(a - \alpha + 1) + \beta f$ 
3:  $\lambda_0 = \left( \sqrt{w^2 + 4\beta f(1 - f)(r + \alpha - 1 - w)} \right) / (2\beta(1 - f))$ 
4:  $\kappa = \left( n(1 - f)^2 \right) / (f + (1 - f)\lambda_0)^2 - (r + \alpha - 1) / \lambda_0^2$ 
5:  $\rho = 1 - \kappa\lambda_0^2$ 
6:  $\tau = -\kappa\lambda_0$ 
7:  $\ln c = \alpha \ln \beta - \ln \Gamma(\alpha) + a \ln f + r \ln(1 - f) + (r + \alpha - \rho) \ln \lambda_0 + (\tau - \beta)\lambda_0 - n \ln (f + (1 - f)\lambda_0)$ 
8: return  $c\Gamma(\rho)/\tau^\rho$ 
```

---