Notes on CNV Methods

David Benjamin* and Samuel K. Lee†
Broad Institute, 75 Ames Street, Cambridge, MA 02142

(Dated: April 13, 2016)

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 ration significantly different from 1:1 in the matched tumor sample imply a CNV event involving one or both alleles. The workflow is as follows:

1. Partition targets into continuous segments that represent the same copy-number event using coverage data. 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.[1]

2. Find heterozygous sites in the normal case sample and segment these, again using CBS, according to their ref:alt allele ratios in the tumor sample.

3. Combine the two sets of segments in a liberal manner that tends to produce too many segments.

4. Alternate between modeling the copy ratio and minor allele fraction of each segment with merging adjacent segments that are sufficiently similar according to this model.

II. SEGMENTATION BY COVERAGE AND MINOR ALLELE FRACTION

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. 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 model 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 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.

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.

*Electronic address: davidben@broadinstitute.org
†Electronic address: slee@broadinstitute.org

[1] Specifically, the CBS implementation provided by the R package DNACopy is used.

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 Jollife's heuristic of retaining singular values greater than 0.7 times the mean singular value.

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

B. 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.

Finally, the tangent-normalized coverage vector is passed to CBS to obtain coverage segments.

C. Het coverage and segmentation by minor allele fraction

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: $a \sim \mathrm{Binom}(a + r, 1/2)$. 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. To obtain initial minor-allele-fraction segments, we estimate the minor allele fraction for each het site by taking the maximum-likelihood estimate given by Equation 5 with allelic bias ignored (i.e., $\lambda_j = 1$) and pass the resulting list to CBS.

D. Target/SNP segment union

[SL: SL will fill this in.]

E. Small-segment merging

[SL: SL to update this to the new method.]

Using CBS to segment the targets in GATK CNV results in segments that are larger than a specified minimum number of targets n_t (by default, $n_t = 2$). 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, ACNV treats these small segments as spurious, and removes them by merging them with adjacent segments.

III. GATK CNV/ACNV MODELS

A. Copy-ratio model

[SL: SL will fill this in.]

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 θ_j be the probability that a mapped read at het j is an alt we have

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

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 θ_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 θ_j is the conditional probability that a mapped read comes from an alt fragment:

$$\theta_j = P(\text{alt}|\text{mapped}) = \frac{P(\text{alt})P(\text{mapped}|\text{alt})}{P(\text{alt})P(\text{mapped}|\text{alt}) + P(\text{ref})P(\text{mapped}|\text{ref})} \tag{2}$$

$$= \frac{f x_j^{\text{alt}}}{f x^{\text{alt}} + (1 - f) x_j^{\text{ref}}} = \frac{f}{f + (1 - f)\lambda_j}, \tag{3}$$

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 λ_j we need an indicator variables 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 θ to get:

$$P(a_j, r_j|\text{outlier}) = \binom{n_j}{a_j} \int_0^1 \theta_j^{a_j} (1 - \theta_j)^{r_j} \, d\theta_j = \binom{n_j}{a_j} \frac{a_j! r_j!}{(n_j + 1)!} \tag{4}$$

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

$$P(a_j, r_j|f_j, \lambda_j, z_j) = \binom{n_j}{a_j} \left[\frac{f_j^{a_j} (1 - f_j)^{r_j} \lambda_j^{r_j}}{(f_j + (1 - f_j)\lambda_j)^{n_j}} \right]^{z_{ja}} \left[\frac{(1 - f_j)^{a_j} f_j^{r_j} \lambda_j^{r_j}}{(1 - f_j + f_j \lambda_j)^{n_j}} \right]^{z_{jr}} \left[\frac{a_j! r_j!}{(n_j + 1)!} \right]^{z_{jo}} \tag{5}$$

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 α, β:

$$P(\lambda_j|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_j^{\alpha - 1} e^{-\beta \lambda_j} \tag{6}$$

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 α and β a flat prior $P(\alpha, \beta) \propto 1$.

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

$$P(z_{ja(r,o)} = 1|\pi) = \pi_{a(r,o)} \tag{7}$$

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 π. Putting all the pieces together the likelihood is

$$\mathbb{L} = \prod_j \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_j^{\alpha - 1} e^{-\beta \lambda_j} \left[\frac{(1 - \pi) f_j^{a_j} (1 - f_j)^{r_j} \lambda_j^{r_j}}{(f_j + (1 - f_j)\lambda_j)^{n_j}} \right]^{z_{ja}} \left[\frac{(1 - \pi)(1 - f_j)^{a_j} f_j^{r_j} \lambda_j^{r_j}}{(1 - f_j + f_j \lambda_j)^{n_j}} \right]^{z_{jr}} \left[\frac{2\pi a_j! r_j!}{(n_j + 1)!} \right]^{z_{jo}}. \tag{8}$$

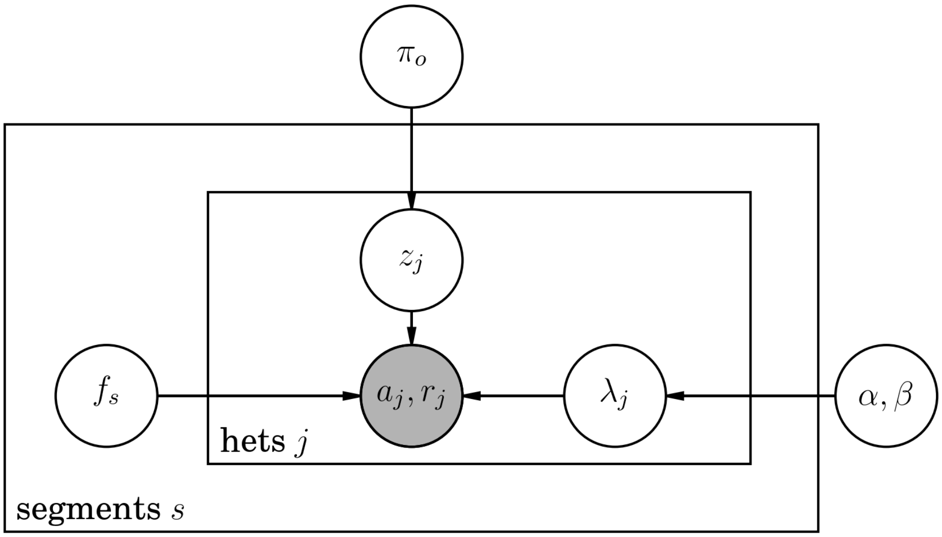

FIG. 1: Graphical model for ACNV allelic model

The dependence on λ for alt minor sites is

$$g(\lambda_j, \alpha, \beta, f_j, a_j, r_j) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{f_j^{a_j} (1-f_j)^{r_j} \lambda_j^{\alpha + r_j - 1} e^{-\beta \lambda_j}}{(f_j + (1-f_j)\lambda_j)^{n_j}}. \tag{9}$$

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

$$\prod_j \left[\frac{1-\pi}{2} \phi(\alpha, \beta, f_j, a_j, r_j) \right]^{z_{ja}} \left[\frac{1-\pi}{2} \phi(\alpha, \beta, 1-f_j, a_j, r_j) \right]^{z_{jr}} \left[\frac{\pi a_j! r_j!}{(n_j+1)!} \right]^{z_{jo}}, \tag{10}$$

where $\phi(\alpha, \beta, f_j, a_j, r_j) = \int_0^\infty g(\lambda, \alpha, \beta, f, a, r) \, d\lambda_j$. Pseudocode for computing ϕ 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

$$p(f, \alpha, \beta, \pi) \propto \prod_j \left[\frac{1-\pi}{2} \phi(\alpha, \beta, f_j, a_j, r_j) + \frac{1-\pi}{2} \phi(\alpha, \beta, 1-f_j, a_j, r_j) + \frac{\pi a_j! r_j!}{(n_j+1)!} \right] \tag{11}$$

Integrating out the latent variables removes the strongest correlations from the model – intuitively, f should not be too sensitive to α and β, for example – and significantly improves mixing. The exception is α and β, since adjusting one with the other fixed changes the mean of the prior on λ. Thus we reparameterize in terms of μ and σ^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:

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.

Algorithm 1 MCMC algorithm for ACNV allelic model

```
Initialize all parameters to a maximum likelihood initial guess (see below).
repeat
    Sample each $f_s$ with adaptive Metropolis
    Sample $\pi$ with adaptive Metropolis
    Sample $\mu$ with adaptive Metropolis
    Sample $\beta$ with adaptive Metropolis
until convergence
```

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 σ^2 these are all reasonable guesses. We choose σ^2 to be larger than what we actually believe because μ converges more slowly from a bad initial guess if σ^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 to 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

$$f_S \approx \frac{\sum_{j \in S} a_j P(z_{ja} = 1) + r_j P(z_{jr} = 1)}{\sum_{j \in S} (a_j + r_j)(P(z_{ja} = 1) + P(z_{jr} = 1))} \tag{12}$$

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

$$P(z_{ja} = 1) \approx I(1/2, a_j + 1, r_j + 1), \quad P(z_{jr} = 1) = 1 - P(z_{ja} = 1). \tag{13}$$

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^{CR} and f_s^{MAF} of the copy ratio and minor allele fraction[2]. That is, $f_s^{MAF}(x)$ is the posterior probability density from ACNV that segment s has minor allele fraction x. We assume that for each segment some fraction ρ 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 ρ 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 ρ take on values $\rho_1, \rho_2 \ldots \rho_K$ and let z_s be a binary-valued indicator vector such that $z_{sk} = 1$ if the CNV on segment s occurs in fraction ρ_k of sequenced cells. Then

$$P(\pi|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_k \pi_k^{\alpha/K-1} \tag{14}$$

$$P(z_s|\pi) = \prod_k \pi_k^{z_{sk}} \tag{15}$$

Here α is the concentration parameter such that the smallness of α/K enforces sparseness[3], i.e. most cluster components will not be used. The $K \to \infty$ 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

[2] ACNV obtains MCMC samples from these posteriors; we assume that a reasonable distribution has been fit to these posterior samples.

[3] If $\alpha < K$ the prior is singular as $\pi_k \to 0$ for any k.

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 α that corresponds to roughly a single- or double-digit number of clusters. For example, a broad gamma prior with mean 1:

$$P(\alpha) = \mathrm{Gamma}(\alpha|1,1) \tag{16}$$

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

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

$$\mathrm{cr}(m, p, \rho) \equiv \left(2(1-\rho) + \rho(m+n)\right)/2. \tag{17}$$

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

$$\mathrm{maf}(m, n, \rho) \equiv \frac{(1-\rho) + \rho\,\min(m,n)}{2(1-\rho) + \rho(m+n)} \tag{18}$$

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 mulitnomial prior, the model likelihood is

$$P(z_s, v_s, w_s, \pi, \phi, \rho, \alpha) = P(\alpha)\frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K}\prod_k \pi_k^{\alpha/K-1} \prod_{s,k,n,m} \left[\pi_k \phi_m \phi_n f_s^{\mathrm{CR}}(\mathrm{cr}(m,n,\rho_k) f_s^{\mathrm{MAF}}(\mathrm{maf}(m,n,\rho_k)\right]^{z_{sk}v_{sm}w_{sn}} \tag{19}$$

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 ρ and ϕ and give the remaining variables the variational factorized distribution $p(\alpha, \pi, z, v, w) \to q(\alpha)q(\pi)q(z, v, w)$. 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

$$\ln q(X) = E_{q(Z)}[\ln p(X, Z)] + \mathrm{const} \tag{20}$$

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

$$X = \arg\max\left[E_{q(Z)}[\ln p(X, Z)]\right] \tag{21}$$

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 α is

$$q(\alpha) \propto \frac{P(\alpha)\Gamma(\alpha)}{\Gamma(\alpha/K)^K}\exp\left(\frac{\alpha}{K}\sum_k E\left[\ln \pi_k\right]\right) \tag{22}$$

The posteriors on π and ϕ are

$$q(\pi) \propto \prod_k \pi_k^{E[\alpha]/K-1+\sum_s E[z_{sk}]},\; q(\phi) \propto \prod_j \phi_j^{\sum_s E[v_{sj}+w_{sj}]} \tag{23}$$

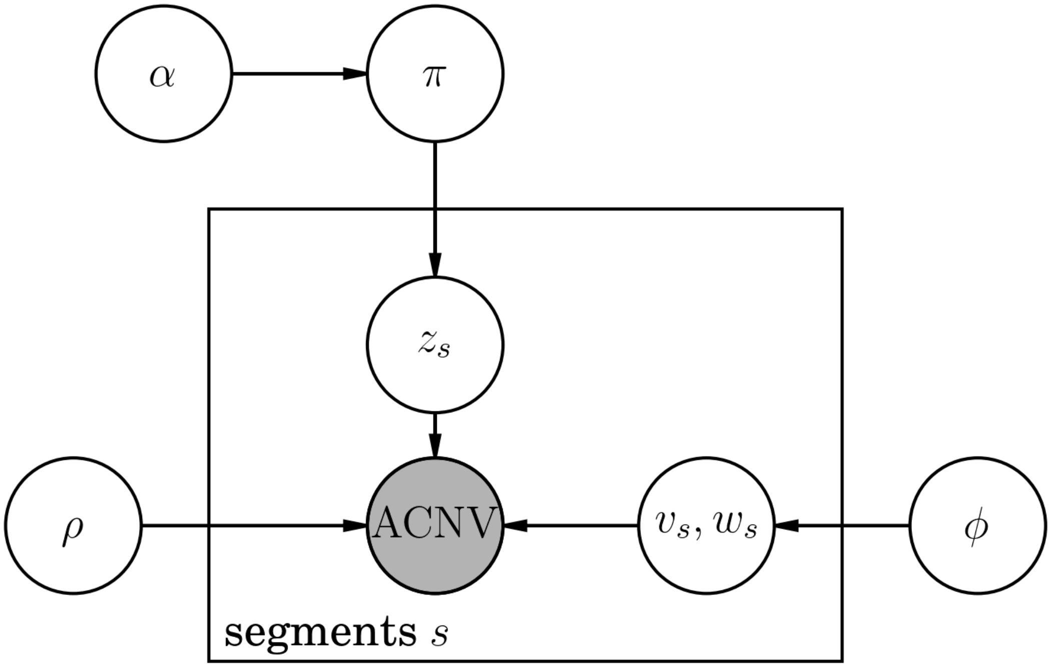

FIG. 2: Graphical model for ACNV caller. “ACNV” represents posterior probability output of ACNV; v, w are indicators of homolog integer copy numbers; ρ is the set of values of purity × cancer cell fraction; z is the corresponding indicator; ϕ is the multinomial prior on homolog counts; π is the multinomial prior on z; α is a Dirichlet concentration parameter encouraging a sparse set of ρ values.

The maximization objective for ρ is

$$\rho_k = \arg\max \sum_{s,k,n,m} E[z_{sk} v_{sm} w_{sn}] \left[\ln f_s^{\mathrm{CR}}(\mathrm{cr}(m,n,\rho_k) + \ln f_s^{\mathrm{MAF}}(\mathrm{maf}(m,n,\rho_k)\right] \tag{24}$$

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

$$E\left[z_{sk} v_{sm} w_{sn}\right] = \frac{\phi_m \phi_n e^{E[\ln \pi_k]} f_s^{\mathrm{CR}}(\mathrm{cr}(m,n,\rho_k) f_s^{\mathrm{MAF}}(\mathrm{maf}(m,n,\rho_k)}{\sum_{k,m,n} \text{“ ”}} \tag{25}$$

Equations 22 – 25 require the expectations $E[\alpha]$, $E[\ln \pi]$, $E\left[z_{sk}\right]$, $E[v_{sj}]$, $E[w_{sj}]$, and $E[z_{sk} v_{sm} w_{sn}]$. The last of these is the E step Equation 25. Three more follow directly from marginalization:

$$E\left[z_{sk}\right] = \sum_{m,n} E[z_{sk} v_{sm} w_{sn}],\ E[v_{sj}] = E[w_{sj}] = \sum_{k,n} E[z_{sk} v_{sj} w_{sn}] \tag{26}$$

By inspection, the Dirichlet posterior $q(\pi)$ of Equation 23 yields the following logarithmic moments:

$$E[\ln \pi_k] = \psi\left(E[\alpha]/K + \sum_s E\left[z_{sk}\right]\right) - \psi\left(E[\alpha] + \sum_{s,k} E\left[z_{sk}\right]\right), \tag{27}$$

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

$$\phi_j = \frac{\sum_s E\left[v_{sj} + w_{sj}\right]}{\sum_{s,i} E\left[v_{si} + w_{si}\right]} \tag{28}$$

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

$$E[\alpha] = \frac{\int \alpha\ q(\alpha)\ d\alpha}{\int q(\alpha)\ d\alpha} \tag{29}$$

Algorithm 2 calling allele counts of ACNV segments

```
Initialize $E[\alpha] = 1$
Initialize $(\rho_1, \rho_2, \ldots \rho_K) = (1/K, 2/K, \ldots 1)$
Initialize $E[\ln \pi_j] = \ln(1/K)$ for all $j$.
Initialize $\phi$ in some reasonable way, i.e. $\phi_1 > \phi_2 > \phi_0 > \phi_3$.
repeat
    Update $E[z_{sk} v_{sm} w_{sn}]$ via Equation 25.
    Update $E\left[z_{sk}\right]$, $E[v_{sj}]$, $E[w_{sj}]$ via Equation 26.
    Update $E[\ln \pi_k]$ via Equation 27
    Update $\phi$ via Equation 28
    Update $E[\alpha]$ via Equation 29
    Update $\rho$ via Equation 24
until convergence
```

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

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

$$P_{st} = \frac{(\text{raw coverage})_{st}}{\text{average depth of sample } s} \tag{30}$$

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 μ 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 μ plus a linear combination of the principal components:

$$P_s \approx \mu + \sum_{k=1}^{M} \beta_{sk} \mathbf{v}_k \tag{31}$$

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:

$$X_s = P_s - \mu - \sum_{k=1}^{M} \beta_{sk} \mathbf{v}_k. \tag{32}$$

(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:

$$Z_{st} = X_{st} / \mathrm{std}(X_{\cdot t}) \tag{33}$$

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 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

$$T = \begin{array}{cc} \text{from\textbackslash to} & \begin{array}{ccc} - & 0 & + \end{array} \\ \begin{array}{c} - \\ 0 \\ + \end{array} & \begin{pmatrix} f & 1-f & 0 \\ p & 1-2p & p \\ 0 & 1-f & f \end{pmatrix} \end{array} \tag{34}$$

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 Λ 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:

$$\begin{aligned} \log \left(T^d \right)_{ij} &= \log \sum_k U_i^T k \Lambda_{kk}^d U_{kj} && (35) \\ &= \log \sum_k \Lambda_{kk}^d U_{kj} U_{ki} && (36) \\ &= \log \sum_k \exp \left(d \log \Lambda_{kk} + \log U_{kj} + \log U_{ki} \right) && (37) \end{aligned}$$

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 s 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 be 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

$$\prod_j \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_j^{\alpha-1} e^{-\beta\lambda_j} \prod_{s \in \mathcal{H}_j} \frac{\lambda_j^{r_{sj}}}{(1+\lambda_j)^{n_{sj}}} \tag{38}$$

$$= \prod_j \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta\lambda_j} \frac{\lambda_j^{\alpha + r_{\cdot j} - 1}}{(1+\lambda_j)^{n_{\cdot j}}} \tag{39}$$

where λ_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 Gamma(α, β), but due to the large number of samples in the panel of normals the data will yield a posterior distribution on each λ_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 α and β. We have in fact already performed this marginalization – Equation 39 is the special case $f = 1/2$, $\pi = 0$ of the allelic-model likelihood, Equation 5, and thus its marginalization over latent variables is obtained by substituting $f = 1/2$, $\pi = 0$ into Equation 11, which yields

$$p(\alpha, \beta) = \prod_j \phi(\alpha, \beta, f = 1/2, n_{\cdot j} - r_{\cdot j}, r_{\cdot j}). \tag{40}$$

This likelihood is easily maximized numerically to obtain MLE values of α and β. Having done this, we can then approximate the posterior on each λ_j as a gamma distribution using the method of Appendix A. As shown there, the posterior on λ_j is Gamma(ρ_j, τ_j) where ρ_j and τ_j are computed in Algorithm 3, with $a \to n_{\cdot j} - r_{\cdot j}$ and $r \to r_{\cdot j}$.

Once we have the posteriors on each λ_j from the panel of normals, they are used as priors for λ_j in the ACNV allelic model. This obviates the hyperparameters α and β, and Equation 11 becomes

$$p(f, \pi) \propto \prod_j \left[\frac{1-\pi}{2} \phi(\rho_j, \tau_j, f_j, a_j, r_j) + \frac{1-\pi}{2} \phi(\rho_j, \tau_j, 1 - f_j, a_j, r_j) + \frac{\pi a_j! r_j!}{(n_j + 1)!} \right] \tag{41}$$

where f and π 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, $\mathbf{n}_s$, $s = 1 \ldots S$ 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:

$$n_{st} \sim \text{Poisson}(d_s c_{st} e^{b_{st}}) \tag{42}$$

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

$$\mathbf{z}_s \sim \text{N}(\mathbf{0}, \mathbf{I}) \tag{43}$$

$$\mathbf{b}_s \sim \text{N}(\mathbf{W}\mathbf{z}_s + \mathbf{m}, \mathbf{\Psi}), \tag{44}$$

where $\mathbf{z}_s \in \mathbb{R}^D$ 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}):

$$\prod_t \text{Poisson}(n_{st} | d_s c_{st} \lambda_{st}) \approx \text{N}(\mathbf{b}_s | \mathbf{m}_s, \mathbf{\Sigma}_s), \; m_{st} = \ln(n_{st}/(c_{st} d_s)), \; \mathbf{\Sigma}_{st} = 1/n_{st} \tag{45}$$

to obtain the joint likelihood

$$P(\mathbf{n}, \mathbf{b}, \mathbf{z} | \mathbf{c}, \mathbf{W}, \mathbf{m}, \mathbf{\Psi}, d) = \prod_s \text{N}(\mathbf{z}_s | \mathbf{0}, \mathbf{I}) \text{N}(\mathbf{b}_s | \mathbf{W}\mathbf{z}_s + \mathbf{m}, \mathbf{\Psi}) \text{N}(\mathbf{b}_s | \mathbf{m}_s, \mathbf{\Sigma}_s) \tag{46}$$

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

$$P(\mathbf{n}, \mathbf{z} | \mathbf{c}, \mathbf{W}, \mathbf{m}, \mathbf{\Psi}, d, \mathbf{n}) = \prod_s \text{N}(\mathbf{z}_s | \mathbf{0}, \mathbf{I}) \text{N}(\mathbf{m}_s | \mathbf{W}\mathbf{z}_s + \mathbf{m}, \mathbf{\Psi} + \mathbf{\Sigma}_s) \tag{47}$$

For fixed $\mathbf{c}$ we may obtain maximum likelihood estimates of $\mathbf{W}$, $\mathbf{\Sigma}$, and $\mathbf{m}$ 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 $\mathbf{W}$, $\mathbf{\Sigma}$, and $\mathbf{m}$ (M step). This is similar to the factor analysis model discussed in Chapter 12 of Bishop. We note that a second level of iteration, on $\mathbf{c}$, allows us to correct for CNV events that may be present in the panel of normals: alternate between 1) learning $\mathbf{W}$, $\mathbf{\Sigma}$, and $\mathbf{m}$ and 2) calling $\mathbf{c}_s$ for each sample in the panel.

The E step follows from substituting $\mathbf{\Psi} \to \mathbf{\Psi}_s \equiv \mathbf{\Psi} + \mathbf{\Sigma}_s$ in Bishop's Equations 12.66-67 to get

$$\mathbf{G}_s = \left(\mathbf{I} + \mathbf{W}^T \mathbf{\Psi}_s^{-1} \mathbf{W} \right)^{-1} \tag{48}$$

$$E\left[\mathbf{z}_s\right] = \mathbf{G}_s \mathbf{W}^T \mathbf{\Psi}_s^{-1} (\mathbf{m}_s - \mathbf{m}) \tag{49}$$

$$E\left[\mathbf{z}_s \mathbf{z}_s^T\right] = \mathbf{G}_s + E[\mathbf{z}_s] E[\mathbf{z}_s]^T \tag{50}$$

For the M steps, we must maximize with respect to

$$E\left[\sum_s \ln \text{N}(\mathbf{m}_s | \mathbf{W}\mathbf{z}_s + \mathbf{m}, \mathbf{\Psi} + \mathbf{\Sigma}_s) \right] = -\frac{1}{2} \sum_{st} \left[\ln \Psi_{st} + E[(\mathbf{W}\mathbf{z}_s + \mathbf{m} - \mathbf{m}_s)_t^2] (\Psi_{st})^{-1} \right] \tag{51}$$

which reduces to

$$-\frac{1}{2} \sum_{st} \left[\ln(\mathbf{\Psi}_t + \Sigma_{st}) + \left(\mathbf{W}_t E[\mathbf{z}_s \mathbf{z}_s^T] \mathbf{W}_t^T + (\mathbf{m}_t - \mathbf{m}_{st})^2 + 2(\mathbf{m}_t - \mathbf{m}_{st}) \mathbf{W}_t E[\mathbf{z}_s] \right) \Psi_{st}^{-1} \right] \tag{52}$$

The M step for $\mathbf{m}$ is routine and comes out to:

$$\mathbf{m} = \left(\sum_s \mathbf{\Psi}_s^{-1}\right)^{-1} \sum_s \left[\mathbf{\Psi}_s^{-1}(\mathbf{m}_s - (\mathbf{W}E[\mathbf{z}_s])\right] \tag{53}$$

Although this is simple and not too expensive to compute ($O(TDS)$ operations), we now introduce a reasonable approximation that we may use as an initial reference implementation or as an optional speed-up. The approximation is $\mathbf{\Sigma}_s \approx \bar{\mathbf{\Sigma}} \equiv \frac{1}{S}\sum_s \mathbf{\Sigma}_s$, in which case $\mathbf{\Psi}_s = \bar{\mathbf{\Psi}}_s = \mathbf{\Psi} + \bar{\mathbf{\Sigma}}$ for each s and cancellations lead to

$$\mathbf{m} = \frac{1}{S}\sum_s (\mathbf{m}_s - \mathbf{W}E[\mathbf{z}_s]) \text{ (simplified)} \tag{54}$$

The M step for $\mathbf{\Psi}_t$ is

$$\sum_s \left[\frac{1}{\mathbf{\Psi}_t + \Sigma_{st}} - \frac{\beta_{st}}{(\mathbf{\Psi}_t + \Sigma_{st})^2}\right] = 0,\ \beta_{st} = \mathbf{W}_t E[\mathbf{z}_s\mathbf{z}_s^T]\mathbf{W}_t^T + (\mathbf{m}_t - \mathbf{m}_{st})^2 + 2(\mathbf{m}_t - \mathbf{m}_{st})\mathbf{W}_t E[\mathbf{z}_s] \tag{55}$$

which is straightforward to solve numerically. Alternatively, the aforementioned approximation gives

$$\Psi_{tt} = \frac{1}{S}\sum_s \beta_{st} - \bar{\Sigma}_{tt}, \text{ (simplified)} \tag{56}$$

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

$$P(\mathbf{W}) = \prod_k \left(\frac{\alpha_k}{2\pi}\right)^{T/2} \exp\left(-\frac{1}{2}\alpha_k \sum_t W_{tk}^2\right). \tag{57}$$

If $\alpha_k \to \infty$ the kth column / feature of $\mathbf{W}$ is effectively turned off. Thus we can initially choose a liberal estimate of K and the model will automatically become more parsimonious. The maximum a posterior result for the row $\mathbf{W}_t$ is routine and gives

$$\mathbf{W}_t^T = \left(\text{Diag}(\alpha_1, \alpha_2 \ldots \alpha_K) + \sum_s \Psi_{st}^{-1} E[\mathbf{z}_s\mathbf{z}_s^T]\right)^{-1} \sum_s \Psi_{st}^{-1}(\mathbf{m}_{st} - \mathbf{m}_t)E[\mathbf{z}_s] \tag{58}$$

In the approximation $\mathbf{\Psi}_s \to \bar{\mathbf{\Psi}}$ this becomes

$$\mathbf{W}_t^T = \left(\bar{\Psi}_{tt}\text{Diag}(\alpha_1, \alpha_2 \ldots \alpha_K) + \sum_s E[\mathbf{z}_s\mathbf{z}_s^T]\right)^{-1} \sum_s (\mathbf{m}_{st} - \mathbf{m}_t)E[\mathbf{z}_s] \tag{59}$$

The other M steps and the E step are not affected by the ARD prior.

The ARD coefficients α are determined via the evidence approximation in which their marginal likelihood is maximized. That is, we set

$$\begin{aligned} \frac{\partial}{\partial \alpha_k} \int P(\mathbf{W}|\alpha)\text{likelihood}(\mathbf{W})\, d\mathbf{W} &= 0 \qquad (60) \\ \frac{1}{2}\int \left(\frac{T}{\alpha_k} - \sum_t W_{tk}^2\right) \text{posterior}(\mathbf{W})\, d\mathbf{W} &= 0 \qquad (61) \\ & \qquad (62) \end{aligned}$$

If we posterior is tightly peaked about the MLE estimate $\mathbf{W}$, this has the solution

$$\alpha_k = \frac{T}{\sum_t W_{tk}^2} \tag{63}$$

We now consider the computational expense of these steps. The slowest part of the E step is computing $\mathbf{G}_s$, which requires $O(D^2T)$ arithmetic operations per sample, $O(D^2TS)$ in total. If we use the approximation $\mathbf{\Psi}_s \to \bar{\mathbf{\Psi}}$ there is only a single $\mathbf{G}$ and the cost is again $O(D^2T)$. Each calculation of $E[\mathbf{z}_s]$ costs $O(DT)$ operations for a total of

$O(DTS)$. For any reasonable panel of normals $S > D$, so the E step cost is $O(D^2TS)$ or $O(DTS)$ depending on the approximation. The cost of the M step for $\mathbf{m}$ is $O(DTS)$, regardless. Calculating β_{st} costs $O(D^2)$ operations for a total cost of $O(D^2TS)$ over all s and t. Then for each Ψ_t the M step optimization objective costs $O(S)$ to compute. Assuming an $O(1)$ cost of numerical optimization, solving for all Ψ_t numerically costs $O(TS)$, which is subleading compared to the cost of calculating all β_{st}. In the approximate version we only need the *sum* $\sum_s \beta_{st}$, which is cheap to compute because, for example, $\sum_s \mathbf{W}_t E[\mathbf{z}_s \mathbf{z}_s^T]\mathbf{W}_t^T = \mathbf{W}_t \left(\sum_s E[\mathbf{z}_s \mathbf{z}_s^T]\right) \mathbf{W}_t^T$. Calculating this for all t while caching the value of $\sum_s E[\mathbf{z}_s \mathbf{z}_s^T]$ costs $O(D^2T)$. The M step for $\mathbf{W}$ costs $O(D^2ST)$ or $O(DST)$ depending on the approximation. The dropped prefactors for all these are small – 1 or 2, usually – so taking everything together the costs of one full EM iteration is roughly $10D^2TS$ without the approximation and $10DTS$ with it. Assuming $T = 2 \times 10^5$, $D = 10$, $S = 500$ these come out to 10^{10} and 10^{11} flops, respectively. On a single 1 GHz core (10^9 flops per second) this comes out to roughly 10 or 100 seconds. Very conservatively, the EM algorithm will converge well within iterations, so the worst-case scenario is about two hours for the unapproximated version. Given that a panel of normals only has to be learned once, no approximations will be needed in production. However, we may find the approximate version very useful for development.

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

$$P(\text{data}|\mathbf{c}) \propto \prod_t \exp\left[-\frac{1}{2}\left((\mathbf{m}_t - \mathbf{m}_{st})^2 + 2(\mathbf{m}_t - \mathbf{m}_{st})\mathbf{W}_t E[\mathbf{z}_s]\right)\Psi_{st}^{-1}\right], \tag{64}$$

which depends on $\mathbf{c}$ via $m_{st} = \ln\left(n_{st}/(c_{st} d_s)\right)$. 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 $d_s c_{st} \to d_s c_{st} f_s(\text{GC}_t)$, 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,

$$c_{st} d_s \exp\left(\mathbf{W}_t E[\mathbf{z}_s] + m_t\right) \tag{65}$$

and thus the ratio $n_{st}/\left[c_{st} d_s \exp\left(\mathbf{W}_t E[\mathbf{z}_s] + m_t\right)\right]$ (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.

Appendix A: Marginalizing out latent variables of the allelic model

We wish to evaluate

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

where

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

An extremely good approximation for all values of f, α, β, and a, r is

$$g(\lambda, \alpha, \beta, f, a, r) = \frac{\lambda^{\alpha+r-1} e^{-\beta\lambda_j}}{\left(f + (1-f)\lambda\right)^{a+r}} \approx c\lambda^{\rho-1} e^{-\tau\lambda}. \tag{A3}$$

where ρ and τ 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 λ, we integrate it analytically

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

The mode λ_0 is found by setting logarithmic derivatives to zero:

$$\begin{aligned} \frac{d}{d\lambda}\left[(\alpha+r-1)\ln\lambda - \beta\lambda - n\ln\left(f+(1-f)\lambda\right)\right]_{\lambda_0} &= 0 && \text{(A5)} \\ \frac{\alpha+r-1}{\lambda_0} - \beta - \frac{n(1-f)}{f_j+(1-f_j)\lambda_0} &= 0 && \text{(A6)} \end{aligned}$$

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

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

The second derivative of $\ln f$ at λ_0 is

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

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

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

Finally, we choose c so that the values of $\ln f$ and the approximation match at λ_0:

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

Algorithm 3 shows the entire computation.

Algorithm 3 Calculating $\phi(\alpha, \beta, f, a, r)$

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