Variational Bayes for variant calling

Daniel Cooke

Variational Bayes is a method to approximate the posterior distribution of latent variables in a Bayesian Network. The technique is similar in nature to EM, but whereas EM gives point estimates of the model parameters, variational Bayes gives a proper distribution over these parameters.

Here I derive a variational Bayes model for approximating the posterior distributions of a genotype model for variant calling. First I describe a generalisation to the standard EM model using my notation.

1 Definitions & conditions

Constant	Description
N	the number of samples
M	the ploidy of each sample
H	the number of haplotypes being considered
K	the number of genotypes

Note I make the assumption that all samples have the same ploidy, this assumption could be relaxed but the maths doesn't work out quite as nicely.

Variable	Description
$\boldsymbol{x}_n$	the read data for sample n
r_n	the number of reads in the n^{th} sample
h	a haplotype
$\boldsymbol{\pi}$	the probabilities (or 'frequencies') of the haplotypes in the samples
$\boldsymbol{g}$	a binary variable using a 1-of-K coding scheme representing the genotypes

In particular $\boldsymbol{g}$ satisfies $g_k \in 0, 1$ and $\sum_{k=1}^{K} g_i = 1$.

Finally I define the the function $\mu_i : \boldsymbol{g} \rightarrow \mathbb{N}$ which maps genotype binary variables to the number of occurrences of haplotype i in the genotype. Alternatively, we could have defined the function maps to a multinomial variable, but I think this notation is clearer. I will sometimes abuse notation by using $\boldsymbol{g}^k$ to mean the $\boldsymbol{g}$ with the k^{th} element set to 1. Then we have the following conditions:

$$K \leq \binom{M + H - 1}{H - 1} \tag{1}$$

$$\sum_{i=1}^{H} \pi_i = 1 \tag{2}$$

$$\mu_i(\boldsymbol{g}) \geq 0 \tag{3}$$

$$\sum_{i=1}^{H} \mu_i(\boldsymbol{g}) = M \tag{4}$$

$$\nexists i, j, \in 1..K i \neq j \quad \text{s.t} \quad \mu_k(\boldsymbol{g}^i) = \mu_k(\boldsymbol{g}^j) \quad \forall k \in 1..H \tag{5}$$

2 EM approach

Here I review the standard EM model used for genotype modelling. This presentation is actually a slight generalisation of that usually presented in the literature as it allows for any number of alleles, which are usually restricted to be biallelic.

The marginal distribution for $\boldsymbol{g}$ is given by a multinomial distribution by assuming Hardy-Weinberg equilibrium

$$p(g_k = 1) = \binom{M}{\mu_1(\boldsymbol{g}^k), \ldots, \mu_H(\boldsymbol{g}^k)} \prod_{i=1}^{H} \pi_i^{\mu_i(\boldsymbol{g}^k)} \tag{6}$$

Note due to the 1-of-K representation for $\boldsymbol{g}$ this can also be written as

$$p(\boldsymbol{g}) = \prod_{k=1}^{K} \left[\binom{M}{\mu_1(\boldsymbol{g}), \ldots, \mu_H(\boldsymbol{g})} \prod_{i=1}^{H} \pi_i^{\mu_i(\boldsymbol{g})} \right]^{g_k} \tag{7}$$

Using the abuse of notation $h \in \boldsymbol{g}$ to mean the haplotypes in $\boldsymbol{g}$, the data distribution is given by

$$p(\boldsymbol{x}|\boldsymbol{g}) = \prod_{i=1}^{r} \sum_{h \in \boldsymbol{g}} p(h|g) p(r_i|h) = \prod_{i=1}^{r} \frac{1}{M} \sum_{h \in \boldsymbol{g}} p(x_i|h) \tag{8}$$

where $p(x_i|h)$ is determined by an HMM. The marginal data distribution is then

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{g}} p(\boldsymbol{g}) p(\boldsymbol{x}|\boldsymbol{g}) = \sum_{k=1}^{K} \binom{M}{\mu_1(\boldsymbol{g}^k), \ldots, \mu_H(\boldsymbol{g}^k)} \prod_{i=1}^{H} \pi_i^{\mu_i(\boldsymbol{g}^k)} \prod_{i=1}^{r} \frac{1}{M} \sum_{h \in \boldsymbol{g}} p(x_i|h) \tag{9}$$

The posterior probabilities, often called responsibilities, of $\boldsymbol{g}$ given read data $\boldsymbol{x}$ is given by Bayes theorem

$$\begin{aligned} \gamma(g_k) &\equiv p(g_k = 1|\boldsymbol{x}) \\ &= \frac{p(g_k = 1) p(\boldsymbol{x}|g_k = 1)}{\sum_{j=1}^{K} p(g_j = 1) p(\boldsymbol{x}|g_j = 1)} \end{aligned} \tag{10}$$

For the 'E' step of the EM algorithm, we need to evaluate the expectation of the complete data log-likelihood, to do this we essentially treat each sample as an individual data point. Viewed this way the entire model can roughly be seen to be a clustering model, where samples are assigned to genotype clusters. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and $\boldsymbol{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n\}$ (note $\boldsymbol{g}_i$ is different from g_i), then the complete data likelihood can be written as

$$p(\boldsymbol{X}, \boldsymbol{G}|\boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\binom{M}{\mu_1(\boldsymbol{g}^k), \ldots, \mu_H(\boldsymbol{g}^k)} \prod_{i=1}^{H} \pi_i^{\mu_i(\boldsymbol{g}^k)} \right]^{g_{nk}} \left[\prod_{i=1}^{r} \frac{1}{M} \sum_{h \in \boldsymbol{g}} p(x_{ni}|h) \right]^{g_{nk}} \tag{11}$$

and so the complete data log-likelihood is

$$\begin{aligned} \ln p(\boldsymbol{X}, \boldsymbol{G}|\boldsymbol{\pi}) &= \sum_{n=1}^{N} \sum_{k=1}^{K} g_{nk} \left\{ \ln \binom{M}{\mu_1(\boldsymbol{g}^k), \ldots, \mu_H(\boldsymbol{g}^k)} \ln \prod_{i=1}^{H} \pi_i^{\mu_i(\boldsymbol{g}^k)} \ln \prod_{i=1}^{r_n} \frac{1}{M} \sum_{h \in \boldsymbol{g}} p(x_{ni}|h) \right\} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} g_{nk} \left\{ \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \ln \pi_i + \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M \right\} \end{aligned} \tag{12}$$

where $C(\boldsymbol{g}^k)$ is the multinomial coefficient. As $\boldsymbol{g}$ is a binary variable we have that $\mathbb{E}[\boldsymbol{g}] = \gamma(g_{nk})$ and thus

$$\mathbb{E}_{\boldsymbol{G}}[\ln p(\boldsymbol{X}, \boldsymbol{G}|\boldsymbol{\pi})] = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(g_{nk})\left\{\ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H}\mu_i(\boldsymbol{g}^k)\ln\pi_i + \sum_{i=1}^{r}\ln\sum_{h\in\boldsymbol{g}^k}p(x_{ni}|h) - r_n\ln M\right\} \tag{13}$$

To maximise w.r.t $\boldsymbol{\pi}$ note we must have $\sum_{i=1}^{H}\pi_i = 1$, and so introducing Lagrange multipliers we maximise

$$\zeta(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{G}}[\ln p(\boldsymbol{X}, \boldsymbol{G}|\boldsymbol{\pi})] + \lambda\left(\sum_{i=1}^{H}\pi_i - 1\right) \tag{14}$$

Differentiating w.r.t π_i and equating to zero gives

$$\frac{\partial\zeta}{\partial\pi_i} = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(g_{nk})\mu_i(\boldsymbol{g}^k)\frac{1}{\pi_i} + \lambda = 0 \tag{15}$$

we then sum over i to find λ

$$\begin{aligned}
\sum_{i=1}^{H}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(g_{nk})\mu_i(\boldsymbol{g}^k) &= -\lambda\sum_{i=1}^{H}\pi_i \\
\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(g_{nk})\sum_{i=1}^{H}\mu_i(\boldsymbol{g}^k) &= -\lambda \\
\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(g_{nk})M &= -\lambda \\
\lambda &= -NM
\end{aligned} \tag{16}$$

and so substituting λ into (15) we have

$$\pi_i = \frac{1}{NM}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(g_{nk})\mu_i(\boldsymbol{g}^k) \tag{17}$$

This intuitively makes sense as the mean expected number of occurrences of haplotype h_i under the posterior for $\boldsymbol{G}$.

3 Variational Bayes

Using EM we do not get a proper probability distribution over the parameters $\boldsymbol{\pi}$, and thus it is difficult to make proper downstream inferences. The optimal solution would be to use a proper Bayesian model by introducing priors over $\boldsymbol{\pi}$ (and as we will see later, possibly also over the hyperparameters of $\boldsymbol{\pi}$).

This leads to analytical difficulties as even if we choose the conjugate prior for $\boldsymbol{\pi}$, a Dirichlet distribution, the problem is still intractable as we must take the expectation over a sum of Dirichlet-Multinomials. This could be solved using stochastic approximation methods, but these methods are probably not ideal here as they can be slow to converge to the true density, and we also introduce a non-determinism in the call set that is better avoided.

There are another set of methods for approximating posterior distributions that do not suffer the above problems called variational Bayesian methods. The idea of these methods is to deterministically approximate

the posterior distribution by enforcing a certain factorisation of the posterior distribution that necessarily introduces some independence assumptions. The factor distributions are then chosen to maximise the 'similarity' to the true posterior. While these methods are usually quick to converge, unlike stochastic methods they can never converge to the true distribution.

3.1 Introduction to variational Bayes

Given a probability model $p(\boldsymbol{X}, \boldsymbol{Z})$ where $\boldsymbol{X}$ is observed and $\boldsymbol{Z}$ are latent, the true posterior density, $p(\boldsymbol{Z}|\boldsymbol{X})$, can be approximated with another distribution, $q(\boldsymbol{Z})$ subject to some measure of similarity. A natural choice of similarity is the KullbackLeibler divergence

$$\mathrm{KL}(q \,||\, p) = -\int q(\boldsymbol{Z}) \ln \frac{p(\boldsymbol{Z}|\boldsymbol{X})}{q(\boldsymbol{Z})} d\boldsymbol{Z} \tag{18}$$

This measures the additional amount of information (in nats) required to generate codes from q rather than p, it satisfies $\mathrm{KL}(q \,||\, p) \geq 0$, with equality when $p = q$. So we actually try to minimise this quantity.

We now partition the latent variables $\boldsymbol{Z}$ into a set of disjoint groups denoted by $\boldsymbol{Z_i}$ where $i = 1, \ldots, M$ and assume that the q distribution factorises into a product of these groups, i.e.

$$q(\boldsymbol{Z}) = \prod_{i=1}^{M} q_i(\boldsymbol{Z_i}) \tag{19}$$

This is the only assumption made, in particular the functional for of each q_i is not contained. The idea is then to consider each group in turn, and average over the other groups. Formally this is a problem that can be solved using calculus of variations, but it's easy to see by substituting (19) into (18) and separating one group $\boldsymbol{Z_j}$

$$\begin{aligned}
\mathrm{KL}(q \,||\, p) &= -\int \prod_{i=1}^{M} q_i \left\{ \ln p(\boldsymbol{Z}|\boldsymbol{X}) - \sum_i \ln q_i) \right\} d\boldsymbol{Z} \\
&= -\int q_j \left\{ \int \ln p(\boldsymbol{Z}|\boldsymbol{X}) \prod_{i \neq j} d\boldsymbol{Z_i} \right\} d\boldsymbol{Z_j} + \int q_j \ln q_j d\boldsymbol{Z_j} + \text{const} \\
&= -\int q_j \mathbb{E}_{i \neq j}[\ln p(\boldsymbol{Z}|\boldsymbol{X})] d\boldsymbol{Z_j} + \int q_j \ln q_j d\boldsymbol{Z_j} + \text{const} \\
&= \mathrm{KL}(q_j \,||\, \exp(\mathbb{E}_{i \neq j}[\ln p(\boldsymbol{Z}|\boldsymbol{X})])) + \text{const}
\end{aligned} \tag{20}$$

Clearly the q_j which minimises this quantity is when $q_j = \exp(\mathbb{E}_{i \neq j}[\ln p(\boldsymbol{Z}|\boldsymbol{X})])$ and therefore we find the optimal q_j

$$q_j^*(\boldsymbol{Z_j}) = \mathbb{E}_{i \neq j}[\ln p(\boldsymbol{Z}|\boldsymbol{X})] \tag{21}$$

Note these equations do not represent an explicit solution because they are interdependent. The variational Bayes algorithm therefore proceeds similar to EM by cycling through each group, updating q^*, and repeating until convergence. It can be shown that $\mathrm{KL}(q \,||\, p)$ decreases at each step.

3.2 The variational Bayes genotype model

As we are approximating a Bayesian model, we need to specify a prior density on the parameters for the genotype model $\boldsymbol{\pi}$. Although these will now be considered latent variables under the new Bayesian model, they are still distinct from the other latent variables as they do not grow in number with the number of data points (i.e. samples). I will therefore continue to refer to them as parameters.

It makes the analysis vastly simpler if we choose a conjugate prior density. We have already noted the marginal genotype density - under Hardy-Weinberg equilibrium - is Multinomial, it therefore makes most sense to choose the prior density $p(\boldsymbol{\pi})$ to be a Dirichlet distribution

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{H} \pi_i^{\alpha_i - 1} \tag{22}$$

where B is the multinomial Beta function. The joint distribution of all random variables is given by

$$p(\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{\pi}) = p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\boldsymbol{G}|\boldsymbol{\pi}) p(\boldsymbol{X}|\boldsymbol{G}) \tag{23}$$

Although this defines a Bayesian model, we could do better by introducing another prior over the hyperparameters, and viewings each parameter $\boldsymbol{\pi}$ as a draw from a 'population' density, this would constitute a hierarchical Bayesian model (the haplotype probabilities are automatically updated for each sample). It may be worth investigating whether this is worthwhile.

There are only two latent variables, so we must have a factorisation of the form

$$q(\boldsymbol{G}, \boldsymbol{\pi}) = q(\boldsymbol{G}) q(\boldsymbol{\pi}) \tag{24}$$

Using (21) we find that

$$\begin{aligned}
\ln q^*(G) &= \mathbb{E}_{\boldsymbol{\pi}}[\ln p(\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{\pi})] + \text{const} \\
&= \mathbb{E}_{\boldsymbol{\pi}}[\ln p(\boldsymbol{G}|\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{\pi}}[p(\boldsymbol{X}|\boldsymbol{G})] + \text{const} \\
&= \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{n=1}^{N} \sum_{k=1}^{K} g_{nk} \left\{ \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \ln \pi_i \right\} \right] \\
&\quad + \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{n=1}^{N} \sum_{k=1}^{K} g_{nk} \left\{ \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M \right\} \right] + \text{const} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} g_{nk} \left\{ \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \mathbb{E}_{\boldsymbol{\pi}}[\ln \pi_i] + \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M \right\} + \text{const} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} g_{nk} \ln \rho_{nk} + \text{const}
\end{aligned} \tag{25}$$

where we have defined

$$\begin{aligned}
\ln \rho_{nk} &= \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \mathbb{E}_{\boldsymbol{\pi}}[\ln \pi_i] + \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M \\
&= \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) (\psi(\alpha_i) - \psi(\alpha_0)) + \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M \\
&= \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \psi(\alpha_i) - \psi(\alpha_0) \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) + \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M \\
&= \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \psi(\alpha_i) - M \psi(\alpha_0) + \sum_{i=1}^{r_n} \ln \sum_{h \in \boldsymbol{g}^k} p(x_{ni}|h) - r_n \ln M
\end{aligned} \tag{26}$$

where ϕ is the digamma function, $\alpha_0 = \sum_{i=1}^{H} \alpha_i$, and $C(\boldsymbol{g}^k)$ is the multinomial coefficient. Hence we find the optimal density for $\boldsymbol{G}$ is given by

$$q^*(\boldsymbol{G}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \tau_{nk}^{g_{nk}} \tag{27}$$

where

$$\tau_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K} \rho_{nj}} \tag{28}$$

The sum in the denominator of τ is a normalisation constant. Next we need to find the optimal distribution for $\boldsymbol{\pi}$. Noting that $\boldsymbol{g}$ is binary so we have $\mathbb{E}[g_{nk}] = \tau_{nk}$, we again use (21) to find

$$\begin{aligned}
\ln q^*(\boldsymbol{\pi}) &= \ln p(\boldsymbol{\pi}) + \mathbb{E}[\ln p(\boldsymbol{G}|\boldsymbol{\pi})] + \text{const} \\
&= \sum_{i=1}^{H} (\alpha_i - 1) \ln \pi_i + \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[g_{nk}] \left\{ \ln C(\boldsymbol{g}^k) + \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \ln \pi_i \right\} + \text{const} \\
&= \sum_{i=1}^{H} (\alpha_i - 1) \ln \pi_i + \sum_{n=1}^{N} \sum_{k=1}^{K} \tau_{nk} \left\{ \sum_{i=1}^{H} \mu_i(\boldsymbol{g}^k) \ln \pi_i \right\} + \text{const}
\end{aligned} \tag{29}$$

and therefore

$$\begin{aligned}
q^*(\boldsymbol{\pi}) &= \prod_{i=1}^{H} \pi_i^{\alpha_i - 1} \prod_{i=1}^{H} \pi_i^{\sum_{n=1}^{N} \sum_{k=1}^{K} \tau_{nk} \mu_i(\boldsymbol{g}^k)} + \text{const} \\
&= \prod_{i=1}^{H} \pi_i^{\alpha_i + \sum_{n=1}^{N} \sum_{k=1}^{K} \tau_{nk} \mu_i(\boldsymbol{g}^k) - 1} + \text{const}
\end{aligned} \tag{30}$$

Which we can recognise as a Dirichlet distribution

$$q^*(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}') \tag{31}$$

where the components of $\boldsymbol{\alpha}'$ are given by

$$\alpha_i' = \alpha_i + \sum_{n=1}^{N} \sum_{k=1}^{K} \tau_{nk} \mu_i(\boldsymbol{g}^k) \tag{32}$$

We can see the similarity of the variational Bayes solution (32) to the EM solution (17). In particular both involve an expectation of the number of occurrences of each haplotype in the samples, the difference is that (32) implicitly accounts for our uncertainty in $\boldsymbol{\pi}$ through τ. Additionally, as in all Bayesian models, the variational Bayes model automatically penalises model complexity through the term $M\psi(\alpha_0)$. It is also worthwhile noting that this variational Bayes method has the same runtime complexity as the EM algorithm.

3.3 Approximate inferences

We can use the optimal factored densities to make approximate haplotype and genotype inferences.

3.3.1 Approximate posterior predictive distribution

We can use the approximate densities to approximate posterior predictive distributions, which we can use to make inferences about the number of haplotypes in the samples. For a vector of haplotype counts $\boldsymbol{z}$, the approximate posterior predictive distribution is

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\alpha}) &= \int p(\boldsymbol{\pi}|\boldsymbol{X}, \boldsymbol{\alpha}) p(\boldsymbol{z}|\boldsymbol{\pi}) d\boldsymbol{\pi} \\ &\approx \int q^*(\boldsymbol{\pi}) p(\boldsymbol{z}|\boldsymbol{\pi}) d\boldsymbol{\pi} \\ &= \int \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \mathrm{Mul}(\boldsymbol{z}|\boldsymbol{\pi}) d\boldsymbol{\pi} \\ &= \frac{z_0!}{\prod_{i=1}^H z_i} \frac{\Gamma(\alpha_0)}{\Gamma(z_0 + \alpha_0)} \prod_{i=1}^H \frac{\Gamma(z_i + \alpha_i)}{\Gamma(\alpha_i)} \end{aligned} \tag{33}$$

where $z_0 = \sum_{i=1}^H z_i$ and $\alpha_0 = \sum_{i=1}^H \alpha_i$.

3.3.2 Finding the most probable allele counts

We could then find the most probable ensemble of haplotypes in the samples by find a MAP estimate for $\boldsymbol{z}$, given the total number of haplotypes is $S = NM$.

$$\begin{aligned} \hat{\boldsymbol{z}}_{MAP} &= \arg\max_{\boldsymbol{z}} \check{p}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\alpha}, S = NM) \\ &= \arg\max_{\boldsymbol{z}} \frac{\check{p}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\alpha}) p(S = NM|\boldsymbol{z})}{p(S = NM)} \\ &= \arg\max_{\boldsymbol{z}} \check{p}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\alpha}) p(S = NM|\boldsymbol{z}) \\ &= \underset{\boldsymbol{z} \text{ s.t } \boldsymbol{z} \in \mathbb{N},\, \sum_k z_k = NM}{\arg\max} \frac{(NM)!}{\prod_{i=1}^H z_i} \frac{\Gamma(\alpha_0)}{\Gamma(NM + \alpha_0)} \prod_{i=1}^H \frac{\Gamma(z_i + \alpha_i)}{\Gamma(\alpha_i)} \end{aligned} \tag{34}$$

In theory this is an integer programming problem which is NP-hard, but if S is small then it is not too much work to find the optimal value.

3.3.3 Sample haplotype posteriors

The posterior haplotype densities for each sample is found using the posterior genotype approximations, τ_n.

$$\begin{aligned} p(h_{ni}|\boldsymbol{X}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{g}} p(g_{nk} = 1|\boldsymbol{X}, \boldsymbol{\alpha}) p(\mu_i(\boldsymbol{g}^k) \geq 1) \\ &\approx \sum_{k=1}^K q^*(g_{nk}) p(\mu_i(\boldsymbol{g}^k) \geq 1) \\ &= \sum_{k=1}^K \tau_{nk} [h_i \in \boldsymbol{g}^k] \end{aligned} \tag{35}$$

Where $[h_i \in \boldsymbol{g}^k]$ is a binary indicator function.

3.4 Specifying the haplotype priors

How should we set the haplotype pseudocounts that determine the haplotype priors? The naive approaches are either to use a flat (or even non-informative) prior, or bias towards the reference. However, these approaches may not be appropriate if we have information that cannot be included directly into the genotype model (e.g. knowledge of well known variants in the population). This is valuable information that should be included in the prior if possible.

3.4.1 Candidate generation & priors

A general way to proceed is to leverge all information we may have about the alleles before we examine the alignments. Strictly speaking we should not let the data influence our priors, but this is clearly infeasable as we would then have to assign every possible allele some positive probability. We get around this by allowing examination of the data to generate candidates - which has the affect of assiging most possible alleles probability zero - but require the priors be data-conditionally independent. This means that the prior of one candidate cannot be influenced by the generation of another candidate from the same source, note this does not mean different candidates cannot be given different priors. The reason for this is to minimise the affect of the data on our priors while allowing the general properties of the candidate generator to be included.

3.4.2 Haplotype priors from allele priors

A haplotype can be viewed as a set of alleles $h = \{a_1, \ldots, a_L\}$ where L is the number of alleles, using the product rule we have $p(h|\boldsymbol{a}) = p(a_1)p(a_2|a_1)\ldots p(a_L|\boldsymbol{a}_{L-1})$, where $\boldsymbol{a}_{L-1}$ denotes the previous $L-1$ alleles. If the haplotypes are short it is probably reasonable to assume independence amongst the alleles giving

$$p(h|\boldsymbol{a}) = \prod_{j=1}^{L} p(a_i) \tag{36}$$

Where $p(a_i)$ is just the prior probability for allele a_i. It therefore seems a reasonable approach to set the pseudocounts, $\boldsymbol{\alpha}$, in proportion to $p(h|\boldsymbol{a})$

$$\alpha_i = \beta \frac{p(h_i|\boldsymbol{a}_i)}{\sum_{j=1}^{H} p(h_j|\boldsymbol{a}_j)} + c \tag{37}$$

where β is a constant scaling factor, and c is a constant that will usually be 1 (c can be set to 0 if the prior in non-informative, but then we must be sure there will be at-least one of each haplotype in the data so that the posterior is proper). How can we set β? Essentially β expresses our confidence in our allele priors, which will usually decrease as the number of samples increases. Therefore β could be of the form $\beta = \max(\beta' - N, 0)$, where β' is a measure of our confidence in our priors if we had only one sample. We can motivate β' by considering how strongly the data must support a low prior allele, compared to a high prior allele, before we change our minds.

Suppose we are examining a site with two candidate haplotypes which differ at a single base. Let A be the allele with a high prior (close to 1), and B be the allele with low prior (close to 0). Now suppose there is a single well mapped read supporting B. Should we prefer homozygous (A,A) or heterozygous (A,B)? The decision will surely come down to the mapping quality of the read (which we assume is high), and the base quality of the read base supporting B. If we assume base qualities are well calibrated, we can choose β' so that we prefer heterozygous (A,B) when the probability of the read base supporting B being wrong is below a certain value.

To see this, suppose the log-likelihood for genotype (A,B) is given by L_{AB}, then given the haplotypes only differ in one position we know that the log-likelihood for genotype (A, A) will take the form $L_{AA} = L_{AB} + C$ ($C < 0$), where C is determined by the base quality of the read base supporting B. For example, if the

base quality of the base supporting B is 20 (i.e. p(base supporting B is wrong) = 0.01) then we expect the difference between the two likelihoods to be close to $\ln(0.01) \approx -4.6$. Then in general we have

$$\ln \rho_{AA} \approx \ln 2 + 2\psi(\alpha_A) - 2\psi(\alpha_0) + L_{AB} + C$$
$$\ln \rho_{AB} \approx \ln 1 + \psi(\alpha_A) + \psi(\alpha_B) - 2\phi(\alpha_0) + L_{AB}$$

Without loss of generality, assume $L_{AB} = 0$ so that

$$\rho_{AA} \approx 2\exp(2\psi(\alpha_A) - 2\psi(\alpha_0) + C)$$
$$\rho_{AB} \approx \exp(\psi(\alpha_A) + \psi(\alpha_B) - 2\psi(\alpha_0))$$

Then we can see from (28) that to have $\tau_{AA} \geq 2\tau_{AB}$ we must have $\rho_{AA} \geq 2\rho_{AB}$, and so

$$\psi(\alpha_A) \geq \psi(\alpha_B) - C \tag{38}$$

Therefore given haplotype priors $p(h_A|a_A)$ and $p(h_B|a_B)$ and a 'minimum quality tolerance' q_{min} we can solve

$$\psi\left(\beta\frac{p(h_A|a_A)}{p(h_A|a_A)+p(h_B|a_B)}+c\right) = \psi\left(\beta\frac{p(h_B|a_B)}{p(h_A|a_A)+p(h_B|a_B)}+c\right) + \frac{\ln(10)}{10}q_{min} \tag{39}$$

This can be approximated analytically, or numerically estimated. For example, if we assume $p(h_A) \approx 1$, $p(h_B) \approx 0$, and $c = 1$, and we require a single read base supporting B to have quality 20 or higher to prefer heterozygous (A,B), then we solve $\psi(\beta + 1) = \psi(1) + 5 \approx 4$ then by inspection we see $\beta \approx 60$.

3.5 Calling variants

We are interested in two probabilities: 1) The probability the allele is present in all the samples. 2) The probability the allele is present in an individual. We can answer both these questions separately.

3.5.1 Posterior probability that a variant is present in samples

To find the posterior probability that an allele appears in the samples we marginalise over each haplotype using the posterior predictive distribution found in (33), this calculates the posterior probability of observing each haplotype in a single new observation and simplifies to the expected value of the haplotype probability

$$\begin{aligned}
p(a|\boldsymbol{X},\boldsymbol{\alpha}) &= \sum_{i=1}^{H} p(a|h_i)p(\boldsymbol{z}_i|\boldsymbol{X},\boldsymbol{\alpha}) \\
&= \sum_{i=1}^{H} [a \in h_i] \frac{z_{i0}!}{\prod_{j=1}^{H} z_{ij}} \frac{\Gamma(\alpha_0)}{\Gamma(z_{i0}+\alpha_0)} \prod_{j=1}^{H} \frac{\Gamma(z_{ij}+\alpha_i)}{\Gamma(\alpha_i)} \\
&= \sum_{i=1}^{H} [a \in h_i] \frac{\Gamma(\alpha_0)\Gamma(\alpha_i+1)}{\Gamma(\alpha_i)\Gamma(\alpha_0+1)} \\
&= \sum_{i=1}^{H} [a \in h_i] \frac{\alpha_i}{\alpha_0}
\end{aligned} \tag{40}$$

where $[a \in h]$ is an indicator function that is 1 if $a \in h$ and is 0 otherwise.

3.5.2 Posterior probability that a variant is present in an individual

Similarly we find the probability of an allele appearing in a sample using the posterior probability found in (35)

$$p(a_n|\boldsymbol{X},\boldsymbol{\alpha}) = \sum_{i=1}^{H}[a_n \in h_i]\sum_{k=1}^{K}\tau_{nk}[h_i \in \boldsymbol{g}^k] \tag{41}$$

3.5.3 Decision theory approach to variant calling

Firstly, we should distinguish between variant calls and genotype calls; we will usually want only one genotype call but may wish to be sensitive to variants. To this end, we should treat genotype calls and variant calls separately.

In either case, the obvious approach is to make calls with the largest posterior probability. However, this may not always be the best approach. For example, if we are seeking deleterious mutations from a reference, then we might want to increase sensitivity at the expense of specificity. In this case it is not always optimal to call the genotype/allele with the largest posterior, instead we should specify some loss function, and call the genotype/allele that minimises the expected loss under the posterior distributions.

There are many possible loss functions we could define, but the simplest is probably that which splits alleles/genotypes into classes 'reference', and 'alternative', and then determines the loss of misclassification. This can be expressed in a matrix of the form

$$\begin{array}{c} \\ \text{reference} \\ \text{alternative} \end{array} \begin{array}{c} \begin{array}{cc} \text{reference} & \text{alternative} \end{array} \\ \begin{pmatrix} 0 & L_{ra} \\ L_{ar} & 0 \end{pmatrix} \end{array} \tag{42}$$

where L_{ra} is our loss of misclassifying a true reference allele as an alternative allele, and L_{ar} is our loss of misclassifying a true alternative allele as a reference allele. For example if we want to be sensitive to variations from the reference we might have the loss matrix

$$\begin{array}{c} \\ \text{reference} \\ \text{alternative} \end{array} \begin{array}{c} \begin{array}{cc} \text{reference} & \text{alternative} \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1000 & 0 \end{pmatrix} \end{array} \tag{43}$$

Once we have defined a loss function we then choose the genotype/allele a_j that minimises the expected loss

$$\sum_{c} L_{cC(a_j)}p(a_j|\boldsymbol{X},\boldsymbol{\alpha}) \tag{44}$$

Where $C(a_j)$ is the 'class' of a_j, Note that if we define a symmetric loss function then we revert to calling the genotype/allele with the largest posterior probability.