Variant Calling and Annotation

Genomics Lecture #8/#9

Peter N Robinson

November 29, 2013

Variant Calling

Variant calling is an important procedure for whole-exome and whole-genome sequencing, and for some experiments also for RNA-seq.

Two major classes of variant

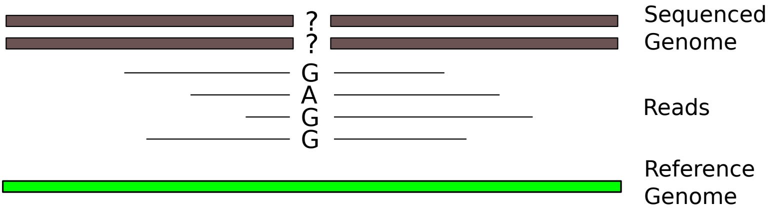
- Single-nucleotide variant (SNV)
- Structural variant

In this lecture, we will discuss issues and algorithms of SNV calling

In the second half of the lecture, we will explain some of the issues and algorithms surrounding variant annotation.

Variant Calling

Genotyping: figuring out collection of alleles in an individual



SNP (pronounced: "Snip") stands for Single Nucleotide Polymorphism, and refers to a position in the genome at which two different bases occur with a frequency of at least 1% in the population. SNV (pronounced "Sniv") stands for Single Nucleotide Variant, and refers to a position in an individual sequence that differs from the reference genome.

Germline variants

Ignoring for the moment everything but SNVs, our goal is to characterize each column of the sequence alignment. We will symbolize the reference base as a and the alternate base as b. There are then only three possible outcomes:

- Homozygous wildtype (aa)
- Heterozygous (ab)
- Homozygous alternate (bb).

Germline variants

Thus, if the true genotype is homozygous reference (`a,a`), and we observe k reference bases at such a position, then the remaining $n-k$ bases must represent sequencing errors, and analogously for homozygous variant (`b,b`), positions.

True Genotype	Number of errors
`a,a`	$n-k$
`b,b`	k

If the true genotype is heterozygous, then we can approximate the probability of the genotype as

$$\mathrm{dbinom}(n,k,p=0.5)=\binom{n}{k}p^k(1-p)^{n-k}=\binom{n}{k}\frac{1}{2^n}$$

A naive algorithm

Early NGS studies basically filtered base calls according to quality and then used a frequency filter.
Typically, a quality filter of PHRED $Q20$ was used (i.e., probability of error 1%). Then, the following frequency thresholds were used according to the frequency of the non-ref base, $f(b)$:

$f(b)$	genotype call
$[0, 0.2)$	homozygous reference
$[0.2, 0.8]$	heterozygous
$(0.8, 1]$	homozygous variant

A naive algorithm

The frequency heuristic works reasonably well if the sequencing depth is high, so that the probability of a heterozygous nucleotide falling outside of the 20% – 80% region is low.

Problems with frequency heuristic:

- For low sequencing depth, leads to undercalling of heterozygous genotypes
- Use of quality threshold leads to loss of information on individual read/base qualities
- Does not provide a measure of confidence in the call

A naive algorithm

For these reasons, a number of probabilistic methods have been developed.
We will discuss two of them and provide some algorithmic background.

- MAQ: An early algorithm.
- SNVmix: A more flexible Bayesian algorithm

The MAQ SNV calling algorithm makes use of the MAP formalism, which will be explained in the following.

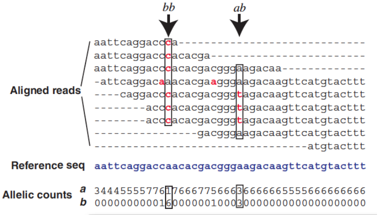
Germline variants

Read mapping ⇒ aligned columns of nucleotides with:

1. mapping quality for each read
2. base call quality for each sequenced nucleotide
3. A stack of nucleotides

We can use this information to improve accuracy of variant calling.

- k wildtype nucleotides a
- $n - k$ nucleotides b
- $a, b \in \{a, c, g, t\}$ and a $\neq$ b



```
                            bb        ab
                aattcaggacca------------------------------
                aattcaggacccacacga------------------------
                aattcaggacccacacgacgggaagacaa-------------
Aligned reads   -attcaggacaaacacgaagggaagacaagttcatgtacttt
                ----caggacccacacgacgggtagacaagttcatgtacttt
                --------acccacacgacgggtagacaagttcatgtacttt
                --------acccacacgacgggtagacaagttcatgtacttt
                ----------------gacgggaagacaagttcatgtacttt
                ---------------------------------atgtacttt

Reference seq   aattcaggaccaacacgacgggaagacaagttcatgtacttt

Allelic counts a 344455557761766677566636666665555666666666
               b 000000000016000000100030000000000000000000
```

Bayes Theorem

Bayes' theorem follows from the definition of the conditional probability and relates the conditional probability $P(A|B)$ to $P(B|A)$ for two events A and B such that $P(B) \neq 0$:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- posterior
- likelihood
- prior
- normalization constant

Bayes Theorem

Bayes' theorem is often used for a set of n mutually exclusive events $E_1, E_2, \ldots, E_n$ such that $\sum_i P(E_i) = 1$. Then, we have

$$P(E_i|B) = \frac{P(B|E_i)P(E_i)}{\sum_i P(B|E_i)P(E_i)}. \qquad (1)$$

- This form of Bayes' theorem makes it clear why $P(B) = \sum_i P(B|E_i)P(E_i)$ is called the **normalization constant**, because it forces the sum of all $P(E_i|B)$ to be equal to one, thus making $P(\cdot|B)$ a real probability measure

Bayes Theorem

In the context of bioinformatics, Bayesian inference is often used to **identify the most likely model**: For instance, we observe a DNA sequence and would like to know if it is a gene (M_1) or not (M_2).

Often, the **model** is symbolized by M and the **observed data** by D. Then, Bayes' theorem can be given as:

$$P(M_1|D) = \frac{P(D|M_1)P(M_1)}{P(D|M_1)P(M_1) + P(D|M_2)P(M_2)} \tag{2}$$

maximum a posteriori (MAP)

In Bayesian statistics, maximum a posteriori (MAP) estimation is often used to generate an estimate of the maximum value of a probability distribution.

That is, if x is used to refer to the data (x can be an arbitrary expression), and θ is used to refer to the parameters of a model, then Bayes' law states that:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)} \tag{3}$$

maximum a posteriori (MAP)

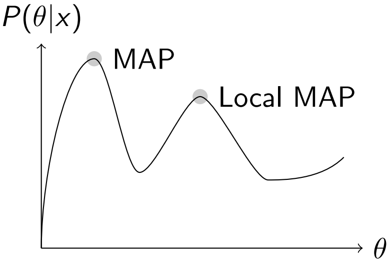
The term $P(\theta|x)$ is referred to as the posterior probability, and specifies the probability of the parameters θ given the observed data x. The denominator on the right-hand side can be regarded as a normalizing constant that does not depend on θ, and so it can be disregarded for the maximization of θ.

The MAP estimate of θ is defined as:

$$\hat{\theta} = \arg\max_{\theta} P(\theta|x) = \arg\max_{\theta} P(x|\theta)P(\theta) \tag{4}$$

maximum a posteriori (MAP)

One important issue about MAP estimation procedures (that we will not discuss further here), is that they tend to have the disadvantage that they "get stuck" in local maxima without being able to offer a guarantee of finding the global maximum.



MAQ: Mapping

One of the first widely used read mappers and variant callers

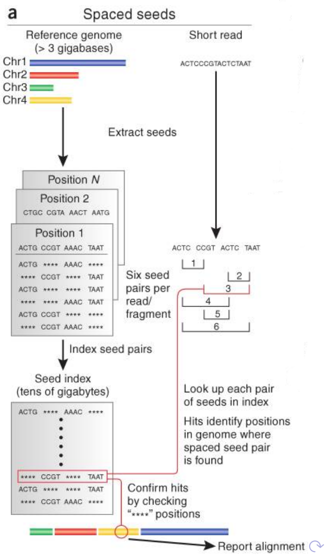
Li H, Ruan J, Durbin R (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores Genome Research 18:1851–1858

- MAQ uses a number of interesting heuristics for read mapping and variant calling
- MAQ calls the genotype that maximizes the posterior probability

MAQ

- index seed pairs of reference and of reads and store in look up table
- Maq indexes the reads in batches and treats substrings of the reference as queries

Trapnell C, Salzberg SL (2009) How to map billions of short reads onto genomes. *Nat Biotechnol* **27**:455-7.



MAQ

We will not review the entire MAQ mapping algorithm, but just those parts that are relevant to the variant calling process

The Mapping Quality for the assigned alignment of a read s is denoted as Q_s, the PHRED-scaled probability that the read alignment is wrong.

$$Q_s = -10 \log_{10} \Pr[\text{read is wrongly mapped}]$$

For example $Q_s = 30$ implies there is a 1:1000 probability that the read s has been wrongly mapped, $Q_s = 20$ implies a 1:100 probability, and so on.

FASTQ and PHRED-like Quality Scores

Recall from lecture #1:

- Illumina sequences are reported in FASTQ format.

```
@My-Illu:6:73:941:1973#0/1
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!’’*((((***+))%%%++)(%%%%).1***-+*’’))**55CCF>>>>>>CCCCCCC65
```

1. Read identifier
2. sequence reported by the machine
3. '+' (can optionally include a sequence description)
4. ASCII encoded base quality scores

PHRED Quality Scores

- The PHRED quality score is defined as

$$\boxed{Q_{PHRED} = -10 \log_{10} p}$$

where p is the probability that the corresponding base call is wrong.

- The PHRED quality score is nothing more than a simple transformation.

Q_{PHRED}	p	Accuracy
10	10^{-1}	90%
20	10^{-2}	99%
30	10^{-3}	99.9%
40	10^{-4}	99.99%
50	10^{-5}	99.999%

Consult slides for lecture # 1 for more details.

MAQ: Mapping quality

We consider the probability that a read z comes from position u of a reference sequence $\mathcal{R}$. Let $\{\mathcal{MM}\}$ refer to the set of mismatched positions in the read.

$$p(z|\mathcal{R},u) = \prod_{i\in\{\mathcal{MM}\}} 10^{-\frac{q_i}{10}} = 10^{-\frac{\sum_i q_i}{10}}$$

That is, the probability that a read z comes from position u of reference sequence $\mathcal{R}$ is modeled as the product of the PHRED quality scores for each of the bases that are mismatched in the alignment.

For instance, if the alignment at position u has one mismatch with PHRED base quality 20 and one with PHRED quality 10, then

$$p(z|\mathcal{R},u) = 10^{-\frac{20+10}{10}} = 10^{-3} = 0.001$$

MAQ: Mapping/base call quality

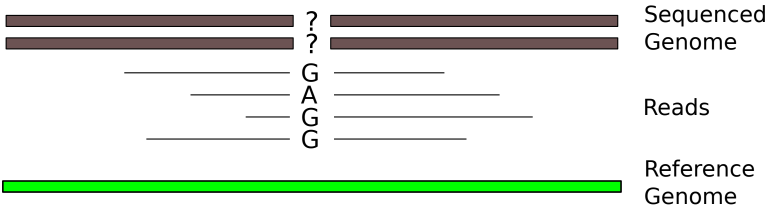
We now calculate the posterior probability of the mapping at position u, $p_s(u|\mathcal{R},z)$ using Bayes law

$$p_s(u|\mathcal{R},z) = \frac{p(z|\mathcal{R},u)p(u|\mathcal{R})}{\sum_v p(z|\mathcal{R},v)p(v|\mathcal{R})}$$

- MAQ actually uses various heuristics to calculate the probability the mapping quality of read z, resulting in a Phred-scaled score q_z for the probability that the read is wrongly mapped
- MAQ then redefines base qualities are redefined as the minimum of base/mapping quality:

$$\boxed{q_i = \min(q_i, q_z)}$$

MAQ: Base/mapping quality



MAQ then uses maximum a posteriori methodology to identify the genotype that maximizes the probabilities

- $p(<a,a>|D)$: homozygous ref
- $p(<a,b>|D)$: heterozygous
- $p(<b,b>|D)$: homozygous alt

Where a refers to the reference base, b to the alternate base, and D to the data (the alignment column)

MAQ: Consensus Genotype Calling

MAQ uses the base quality values to call the most likely genotype. We assume we have a column of an alignment with

- k references bases (a)
- $n-k$ alternate bases (b)[1]

True Genotype	# errors	Cond. Prob. of Genotype[2]
a,a	$n-k$	$\alpha_{n,n-k}$
b,b	k	$\alpha_{n,k}$
a,b	?	$\binom{n}{k}\frac{1}{2^n}$

[1] Any other bases are ignored as being probably sequencing errors.

[2] We will explain α shortly.

MAQ: Consensus Genotype Calling

The goal is now to decide which of the three possible genotypes has the highest posterior probability given the data (the mapping and alignment): $p(g|D)$.

MAQ now assumes the prior for the genotypes is

- $P(\langle a, a\rangle) = (1 - r)/2$
- $P(\langle b, b\rangle) = (1 - r)/2$
- $P(\langle a, b\rangle) = r$

Here, r is the probability of observing a heterozygous genotype. MAQ uses $r = 0.001$ for new SNPs, and 0.2 for known SNPs, but site-specific values for r could also be used.

MAQ: Consensus Genotype Calling

MAQ thus calls the genotypes as

$$\hat{g} = \arg\max_{g} p(g|D)$$

Here, the genotype $g \in (\langle a, a\rangle, \langle a, b\rangle, \langle b, b\rangle)$ with the maximum posterior probability is sought. The quality of this genotype call can then be calculated as

$$Q_g = -10 \log_{10}[1 - P(\hat{g}|D)]$$

- Q_g is then the reported PHRED score for the variant call.
- Note that $1 - P(\hat{g}|D)$ is the calculated probability that the genotype call is wrong.

MAQ: Consensus Genotype Calling

We now need a way of calculating $\alpha_{n,k}$, the probability of observing k errors in n nucleotides in the alignment. If we assume that error rates arise independently, and error rates are identical for all bases, then we can use a binomial distribution

$$\mathrm{dbinom}(n, k, p = \epsilon) = \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k}$$

For instance, the probability of observing 2 erroneous nucleotides in 20, if the per read error rate is $\epsilon = 0.01$ can be calculated in R as

```
> dbinom(2,20,0.01)
[1] 0.01585576
```

MAQ: Consensus Genotype Calling

In practice, MAQ errors are correlated and are not identical for each base in the alignment. Therefore, MAQ does not use a binomial distribution, but a heuristic that reflects the probabilities of observing an alignment with the given pattern of per base error probabilities.

$$\alpha_{n,k} = c'_{n,k} \prod_{i=0}^{k-1} \epsilon_{i+1}^{\theta^i}$$

Here, ϵ_i is the i$^{\text{th}}$smallest base error probability for the k observed errors, $c'_{n,k}$ is a constant and θ is a parameter that controls the dependency of errors.

This equation reflects the base errors. We will not go into further detail, but if desired see the Supplemental material of the MAQ paper.

MAQ: Consensus Genotype Calling

With all of this, we can now call the posterior probabilities of the three genotypes given the data D, that is a column with n aligned nucleotides and quality scores of which k correspond to the reference a and $n-k$ to a variant nucleotide b.

$$\begin{aligned}
p(G=\langle a,a\rangle|D) &\propto p(D|G=\langle a,a\rangle)p(G=\langle a,a\rangle) \\
&\propto \boxed{\alpha_{n,k}\cdot(1-r)/2} \\
p(G=\langle b,b\rangle|D) &\propto p(D|G=\langle b,b\rangle)p(G=\langle b,b\rangle) \\
&\propto \boxed{\alpha_{n,n-k}\cdot(1-r)/2} \\
p(G=\langle a,b\rangle|D) &\propto p(D|G=\langle a,b\rangle)p(G=\langle a,b\rangle) \\
&\propto \boxed{\binom{n}{k}\frac{1}{2^n}\cdot r}
\end{aligned}$$

MAQ: Consensus Genotype Calling

Finally, the genotype with the highest posterior probability is chosen

$$\hat{g} = \underset{g \in (\langle a,a \rangle, \langle a,b \rangle, \langle b,b \rangle)}{\arg\max} \; p(g|D)$$

The probability of this genotype is used as a measure of confidence in the call.

What have we learned?

The MAQ algorithm is typical for many in genomics in that a well known statistical or algorithmic framework is used with a number of heuristics that deliver reasonable values for the parameters needed for the framework to work.

Major aspects of MAQ SNV calling algorithm

- Integrates mapping and per base quality scores
- Bayesian (MAP) framework to integrate observations and priors on genotypes
- Provides estimation of reliability of genotype call.

Expectation Maximization (EM)

We will discuss how EM is used for mixture distributions. For ease of presentation, we will discuss in detail a mixture of Gaussians, but the principles are the same for other probability distributions

The basic framework is that we assume that a data point y_j is produced as follows

- First, choose one of $i \in \{1, .., I\}$ components that produces the measurement
- Then, according to the parameters of component $C = i$, the actual measurement is generated

This is known as a mixture distribution, and the corresponding probability density function (pdf) is defined as

$$p(y_j|\boldsymbol{\theta}) = \sum_{i=1}^{I} \alpha_i p(y_j|C = i, \boldsymbol{\beta}_i) \tag{5}$$

Expectation Maximization (EM)

Note that in this notation, θ comprises both the weight parameters for the probability of one of the I components generating the data, as well as the various parameters for each of the components, β_i (where in general β_i can be a vector of parameters)

- Of course, $\sum_{i=1}^{I} \alpha_i = 1$
- The parameters $\boldsymbol{\beta}_i$ are associated with the PDF of component i.

We will now show how to perform maximum likelihood estimation using the Expectation Maximization (EM) framework to find values for the parameters θ that maximize the probability of the data. This involves maximization of the log-likelihood for θ.

$$\log L(\boldsymbol{\theta}) = \log p(\mathbf{y}|\boldsymbol{\theta}) \quad (6)$$

Expectation Maximization (EM)

We can flesh out the formula as follows

$$\begin{aligned}
\log L(\theta) &= \log p(\mathbf{y}|\boldsymbol{\theta}) \\
&= \log \left\{ \prod_{j=1}^{J} p(y_j|\boldsymbol{\theta}) \right\} \\
&= \sum_{j=1}^{J} \log p(y_j|\boldsymbol{\theta}) \\
&= \sum_{j=1}^{J} \log \left\{ \sum_{i=1}^{I} \alpha_i p(y_j|C=i,\boldsymbol{\beta}_i) \right\}
\end{aligned}$$

- Since the log is outside the sum in the last expression, there is no analytic (closed form) optimization.

Expectation Maximization (EM)

The EM algorithm is essentially like a pushme-pullyou algorithm that goes back and forth between

- find an estimate for the likelihood function
- maximizing the whole term

Expectation Maximization (EM)

If there is time in the practical session, I will explain the derivation of the EM method and show in detail how the maximization expressions are derived for a simple distribution – mixture of Gaussians. For today, I will show only a high level summary. For the practical, you will be expected to implement a simplified version of EM – known as gene counting (will explain at end of this lecture)

Expectation Maximization (EM)

The EM algorithm will be explained using a mixture of Gaussians (SNVMix uses some slightly less familiar distributions). Recall the form of a multivariate Gaussian distribution for a k-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_k]$:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{7}$$

i.e. the probability density function is

$$f(x_1, x_2, \ldots, x_k) = \frac{1}{(2\pi)^{k/2} \left|\Sigma\right|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{8}$$

Expectation Maximization (EM)

We thus wish to estimate the parameters for a mixture of Gaussians. We need to estimate both the mixture parameters $\pi_1, \pi_2, \ldots, \pi_c$ with $\sum_{i=1}^{c} \pi_i = 1$, but also the means and variances for each of the individual Gaussian distributions, $\mu_1, \mu_2, \ldots, \mu_c$, and $\Sigma_1, \Sigma_2, \ldots, \Sigma_c$.

We thus want to maximize the log likelihood given by

$$L(\theta|\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) = \log \prod_{i=1}^{n} \sum_{k=1}^{c} \pi_k f(\mathbf{x}_i|\mu_k, \Sigma_k) \qquad (9)$$

or equivalently

$$L(\theta|\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) = \sum_{i=1}^{n} \log \sum_{k=1}^{c} \pi_k f(\mathbf{x}_i|\mu_k, \Sigma_k) \qquad (10)$$

Expectation Maximization (EM)

We can now calculate the probability that a particular data point $\mathbf{x}_j$ belongs to a particular component k

We write the posterior probability that an observation $\mathbf{x}_j$ belongs to component k as

$$\hat{\tau}_{jk} = \frac{f(\mathbf{x}_j|\mu_k, \Sigma_k)\hat{\pi}_k}{\sum_{i=1}^{c} f(\mathbf{x}_j|\mu_i, \Sigma_i)\hat{\pi}_i} \qquad (11)$$

- The posterior probability $\hat{\tau}_{jk}$ is unknown but can be easily estimated if we use the current values of the parameters for the Gaussians.
- $\hat{\tau}_{jk}$ is thus an estimate for the probability that observation j was generated by component k given the data and current parameter estimates
- This is the **Expectation** step of the EM algorithm

Expectation Maximization (EM)

Given our current estimates of the component membership for each of the datapoints, we can maximize the values of the mixture parameters as well as of the Gaussians by setting their first derivative to zero etc (individual steps not shown here). This leads to the following

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{\tau}_{ik} \tag{12}$$

$$\hat{\mu}_k = \frac{1}{n} \sum_{i=1}^{n} \frac{\hat{\tau}_{ik} \mathbf{x}_i}{\hat{\pi}_k} \tag{13}$$

$$\hat{\Sigma}_k = \frac{1}{n} \sum_{i=1}^{n} \frac{\hat{\tau}_{ik} \left(\mathbf{x}_i - \hat{\mu}_k\right) \left(\mathbf{x}_i - \hat{\mu}_k\right)^T}{\hat{\pi}_k} \tag{14}$$

This is the **Maximization** step of the EM algorithm

Expectation Maximization (EM)

Thus, the individual steps of the EM algorithm are thus

1. Initial component parameters (with a reasonable guess)
2. For each data point, calculate posterior probability of membership to each component using the current parameter values
3. Then, based on these estimates, maximize the log likelihood of the parameters given the data
4. Repeat until convergence[3]

[3] and hope you have not landed in a local maximum.

SNVMix

We will now discuss an algorithm called SNVMix, that uses the EM framework to estimate optimal parameters for calling SNPs in a Bayesian framework.

- This algorithm was first described here:
  Shah SP et al. (2009) Mutational evolution in a lobular breast tumor profiled at single nucleotide resolution. *Nature* **461**:809-13.
- an improved version (which we will not discuss) was later presented here:
  Goya R et al. (2010) SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors. *Bioinformatics* **26**:730-6.

SNVMix: Model specification

The core if the model is a specification of the genotypes and the conditional probabilities of the observed distribution of alleles – similar to MAQ.

- Let $G_i = k, k \in \{\langle a, a\rangle, \langle a, b\rangle, \langle b, b\rangle\}$ be a multinomial random variable representing the genotype at nucleotide position i ($a=$ ref, b is non-ref).
- Let the observed allele frequency $X_i = [a_i, b_i]^T$, i.e., a vector of counts of the reference and non-reference alleles at position i
- Then $N_i = a_i + b_i$ is the observed read depth at position i.

SNVMix: Model specification

The central idea is that we assume the allele counts are generated by a class conditional density

Thus

$$X_i \sim \mathrm{Binom}(a_i|\mu_k, N_i) \tag{15}$$

The probability of the observed read counts (number of a_i bases amongst all N_i bases at position i) is thus conditioned on the underlying genotype $G_i = k$, and μ_k is the corresponding parameter of a Binomial distribution for genotype k.

SNVMix: Model specification

If we actually knew the genotype, then it is simplicissimo to calculate the probability of the allele counts using the binomial distribution.

Thus

$$P(X_i) = \mathrm{Binom}(a_i|\mu_k, N_i) = \binom{N_i}{a_i} \mu_k^{a_i} (1-\mu_k)^{N_i-a_k} \tag{16}$$

Intuitively, we would expect the values of μ_{aa} to be close to 1, those for μ_{ab} to be close to 0.5 and those for μ_{bb} to be near zero. However, we do not know the exact values for real data, which may depend on things such as the sequencing error rate[4].

[4] And for cancer data, on the relative mixture of normal and cancerous tissue in a biopsy.

SNVMix: Model specification

The prior probability of observing a genotype k at any position of the sequenced genome is represented as a multinomial variable π.

- $0 \leq \pi_k \leq 1, \quad \forall k$
- $\sum_{k=1}^{3} \pi_k = 1$
- Note that in general we will expect the values of π to be highly skewed towards observing homozygous reference bases (since most genomic positions are not variant in any one individual)
- SNVMix is thus a classic generative mixture model to explain the observed data.

SNVMix: Model specification

The marginal distribution of X_i (in which we have marginalized – removed – the influence of the actual genotype) can then be calculated as the convex combination of the class conditional Binomial densities, weighted by the multinomial π:

$$p(X_i) = \sum_{k=1}^{3} \pi_k \binom{N_i}{a_i} \mu_k^{a_i} (1 - \mu_k)^{N_i - a_k} \tag{17}$$

Again, the sum is taken over k representing the three genotypes *aa*, *ab*, and *bb*.

SNVMix: Model specification

We can then use this equation to calculate the log likelihood of our entire dataset, which comprises positions $1 \dots T$.

$$\log p(X_{1:T}|\mu_{1:K}, \pi) = \sum_{i=1}^{T} \log \sum_{k=1}^{3} \pi_k \binom{N_i}{a_i} \mu_k^{a_i} (1-\mu_k)^{N_i - a_k} \tag{18}$$

Our problem is that the model parameters $\theta = (\pi, \mu)$ are not known. If the true genotype were somehow known, we could simply calculate them from the training data. But, instead, we will learn (estimate) the parameters from data by using maximum a posteriori (MAP) expectation maximization (EM).

SNVMix: Model specification

Assuming we have solved for the parameters (we will get to that shortly), then we can easily calculate the posterior probability of any genotype using Bayes rule

$$p(G_i = k|X_{1:N}, \pi, \mu_k) = \frac{\pi_k \mathrm{Binom}(X_i|\mu_k, N_i)}{\sum_{j=1}^{3} \pi_j \mathrm{Binom}(X_i|\mu_j, N_i)} \tag{19}$$

For notational simplicity, we will denote $p(G_i = k|X_{1:N}, \pi, \mu_k)$ as $\gamma_i(k)$, the marginal probability of the genotype for position i given all the data and the model parameters.

SNVMix: Prior Distributions

Bayesian mixture models use hyperparameters, i.e., parameters of a prior distribution; the term is used to distinguish them from parameters of the model used for the final analysis. We will use two underlying distributions to calculate these hyperparameters for SNVMix.

- $\pi \sim \mathrm{Dirichlet}(\pi|\delta)$
- $\mu \sim \mathrm{Beta}(\mu_k|\alpha_k, \beta_k)$

Beta distribution (review)

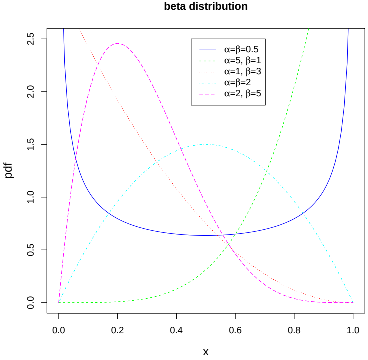
Beta distribution: A family of continuous distributions defined on $[0,1]$ and parametrized by two positive shape parameters, α and β

$$p(x) = \frac{1}{B(\alpha,\beta)} \cdot x^{\alpha-1}\,(1-x)^{\beta-1}$$

here, $x \in [0,1]$, and

$$B(\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)}$$

where Γ is the Gamma function (extension of factorial).



Beta distribution

Important in the current context is that the Beta distribution is the conjugate prior to the binomial distribution.

- That is, we can express our **prior belief** about the value of the μ_k parameter of the binomial distribution for read counts using a Beta distribution
- We say that μ_k is conjugately distributed according to a Beta distribution: $\mu_k \sim \mathrm{Beta}(\mu_k|\alpha_k, \beta_k)$.
- This requires us to express our prior belief about μ_{aa}, μ_{ab}, and μ_{bb} by specifying values for α_k, β_k

Beta distribution

For instance, let us say we are not very sure about what we think the value of $\mu_{a,b}$ should be, perhaps because we do not know if the sample being sequenced contains tumorous or non-tumorous tissues[5].

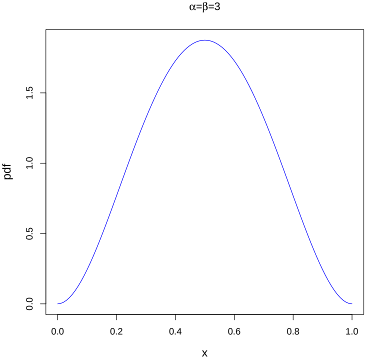
- We might then try $\alpha_{ab} = \beta_{ab} = 3$

```
x <- seq(0.0, 1.0, 0.01)
y <- dbeta(x, 3, 3)
title <- expression(paste(alpha,"=",beta,"=3"))
plot(x, y, type="l",main=title,
xlab="x",ylab="pdf",col="blue",lty=1,cex.lab=1.25)
```

[5] Tumor tissue may be characterized by the loss of heterozygosity (LOH) of large chromosomal regions.

Beta distribution

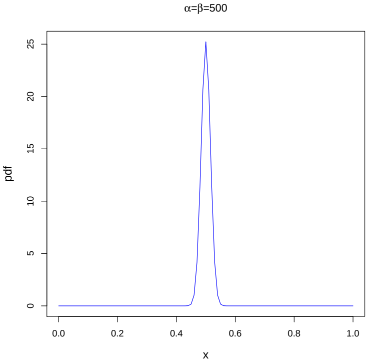
$\alpha_{ab} = \beta_{ab} = 3$



- x is here representing a value for μ_{ab}, and the y axis reflects our belief about the prior probability of this value
- Question: Are we very sure about μ_{ab}?

Beta distribution

$\alpha_{ab} = \beta_{ab} = 500.$



- Question: How sure are we now about μ_{ab}?

Beta distribution in SNVMix

The expected value of a $\mathrm{Beta}(\alpha + \beta)$ distribution is simply

$$\frac{\alpha}{\alpha + \beta}$$

In SNVMix, these values are defined as

- $\alpha_{aa} = 1000$, $\beta_{aa} = 1$, that is, our prior belief in reference reads given homozygous reference sequence is 0.999001
- $\alpha_{ab} = 500$, $\beta_{ab} = 500$, that is, our prior belief in reference reads given a het true sequence is 0.5
- $\alpha_{bb} = 1$, $\beta_{bb} = 1000$ (vice versa to $\alpha_{aa} = 1000$, $\beta_{aa} = 1$)

SNVMix: M-step updating equation for μ

The maximization step updating equation basically adds to observed counts for a certain true genotype to our prior.

$$\mu_k^{new} = \frac{\sum_{i=1}^{T} a_i^{I(G_i=k)} + \alpha_k}{\sum_{i=1}^{T} N_i^{I(G_i=k)} + \alpha_k + \beta_k - 2} \tag{20}$$

- Note that in this notation, $I(G_i = k)$ is an indicator function so that the expression is zero unless $G_i = k$
- The update is simply the proportion of the observed reference reads with "pseudocounts" added from the Beta prior (amongst all positions called to genotype k).

For details recall lecture # 2

Dirichlet

The Dirichlet distribution is the multivariate generalization of the Beta distribution and represents the conjugate prior of the multinomial distribution. Thus, just as SNVMix used the Beta distribution as a prior for μ (binomial distribution of read counts), it uses the Dirichlet as a prior for π (multinomial distribution for the three possible genotypes).

In SNVMix, the values for the prior are set to

$\delta(\langle a, a\rangle)$	$\delta(\langle b, b\rangle)$	$\delta(\langle b, b\rangle)$
1000	100	100

Dirichlet

Thus, the prior is skewed toward π_{aa} assuming that most positions will be homozygous for the reference allele. The pseudocounts are essentially equivalent to having seen $1000 + 100 + 100 = 1200$ positions with the distribution 83.3% $\langle a, a \rangle$, and 8.3% each for $\langle b, b \rangle$ and $\langle b, b \rangle$.

The weight of the prior belief is reflected in the number of pseudocounts. For instance, the following counts result in the same proportion but there is much less weight of prior belief

$\delta(\langle a, a \rangle)$	$\delta(\langle b, b \rangle)$	$\delta(\langle b, b \rangle)$
10	1	1

Dirichlet

A Dirichlet distribution of order k (3 in our example) is a PDF that represents the belief ("probability") that the probabilities of k distinct events (in our case, the genotypes $\langle a, a\rangle, \langle a, b\rangle, \langle b, b\rangle$) are x_i given that each event has been observed $\alpha_i - 1$ times.

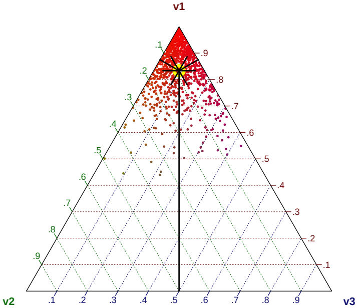
$$f(x_1, x_2, \ldots, x_{k-1}; \alpha_1, \alpha_2, \ldots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} x_i^{\alpha_i - 1} \tag{21}$$

- Note that by convention f has $k-1$ arguments. Since $\sum_{i=1}^{k} x_i = 1$ there is no need to show the k^{th} argument.
- $B(\alpha)$ is the Beta function

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)} \tag{22}$$

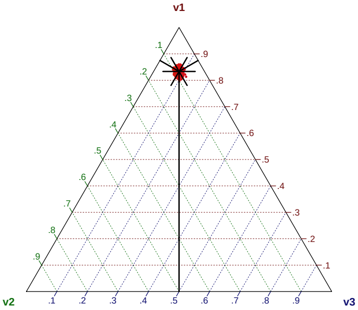
Dirichlet distribution with a low number of pseudocounts (relatively weak prior):

- $\delta(\langle a, a \rangle) = 10$
- $\delta(\langle a, b \rangle) = 1$
- $\delta(\langle b, b \rangle) = 1$



Dirichlet distribution with a higher number of pseudocounts (relatively strong prior):

- $\delta(\langle a, a\rangle) = 1000$
- $\delta(\langle a, b\rangle) = 100$
- $\delta(\langle b, b\rangle) = 100$



SNVMix: M-step updating equation for π

The update equation for π is similar to that for μ

$$\pi_k^{new} = \frac{\sum_{i=1}^{T} I(G_i = k) + \delta(k)}{\sum_{j \in \{\langle a,a \rangle, \langle a,b \rangle, \langle b,b \rangle\}} \sum_{i=1}^{T} I(G_i = j) + \delta(j)} \tag{23}$$

SNVMix: Initialize EM

We are now in a position to initialize the EM

We need:

- Mapped NGS reads comprising $i = 1, \ldots, T$ genomic positions, each of which has N_i reads with a_i reference and b_i nonreference bases.
- Initialize $\pi_k = \dfrac{\delta(k)}{N_\delta}$ where $N_\delta = \dfrac{\delta(k)}{\sum_j \delta_(j)}$
- Initialize $\mu_k = \dfrac{\alpha_k}{\alpha_k + \beta_k}$
- pick a tolerance to judge convergence

SNVMix: Run EM

The EM algorithm iterates between the E-step where we assign the genotypes using Equation (**??**) and the M-step where we re-estimate the model parameters with equations (**??**) for π, (**??**) for μ.

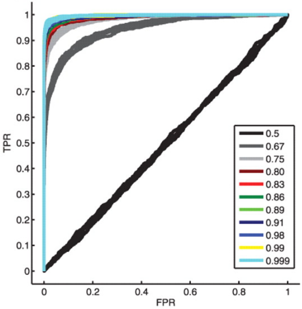
At each iteration we evaluate the complete data log-likelihood as given by Equation (**??**) and the algorithm terminates when this quantity no longer increases

SNVMix1 vs SNVMix2

The SNVMix algorithm was later extended to include mapping and base qualities into the same Bayesian framework, primarily by adapting the formulas used for the EM equations. We will not discuss this here[a]

[a] Goya R et al. (2010) *Bioinformatics* **26**:730–736.

Performance of SNVMix2 algorithm on simulated data with increasing levels of certainty in the base call



Summary

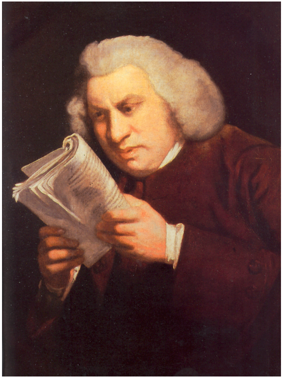
In these two lectures we have examined how to call variants. We have studied two classes of algorithms, MAP estimation and expectation maximization, and how they are used in two variant calling algorithms, MAQ and SNVMix. For practical use, newer variant callers, especially GATK, are recommended.

What you should now know:

- The kinds of data used in variant calling (mapping quality, base quality, depth, . . .)
- How this data is exploited to improve variant calling
- Bayes' law and how it can be used to estimate parameters
- Be able to interpret the major formulae of MAP and EM

The End of the Lecture as We Know It

- Email: peter.robinson@charite.de
- Office hours by appointment



Lectures were once useful; but now, when all can read, and books are so numerous, lectures are unnecessary. If your attention fails, and you miss a part of a lecture, it is lost; you cannot go back as you do upon a book... People have nowadays got a strange opinion that everything should be taught by lectures. Now, I cannot see that lectures can do as much good as reading the books from which the lectures are taken. I know nothing that can be best taught by lectures, except where experiments are to be shown. You may teach chymistry by lectures. You might teach making shoes by lectures!