Variant Calling in NGS Data

ML vs. MAP

Peter N Robinson

November 14, 2013

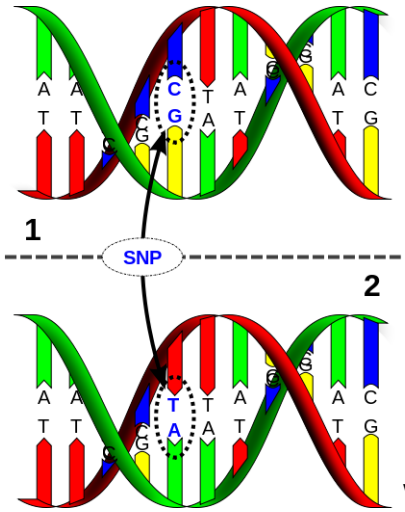
Today's Lecture

Variant calling is one of the key challenges in many areas of genomics research and diagnostics. Having aligned the fragments of one or more individuals to a reference genome, **SNP calling** identifies variable sites, whereas **genotype calling** determines the genotype for each individual at each site.

- Review: SNPs and SNVs, variant annotation
- Introduction to variant calling: pileups
- Review of Bayesian concepts we will need for more advanced variants callinig algorithms and other topics in this course

SNPs and SNVs

- SNP: Single-nucleotide polymorphism. A variant that is polymorphic within a population
- SNV: Single-nucleotide variant: A variant called in an individual sequence
- (However, SNP and SNV often are used interchangeably)



Wikipedia

Finding the needle . . .

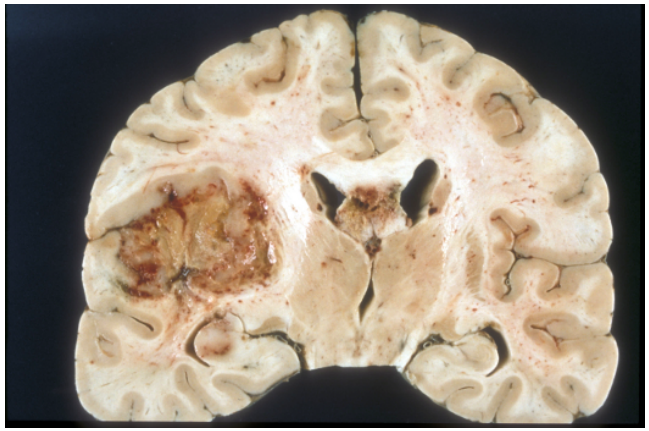
	X-ome	Exome	Genome
SNVs	800–1200	30,000-40,000	3–4 Mio.
¬ dbSNP	100–300	1,000–3,000	100K–300K
Indels (<10bp)	100–200	3,000	600K
¬ dbSNP	50	1,500	150K

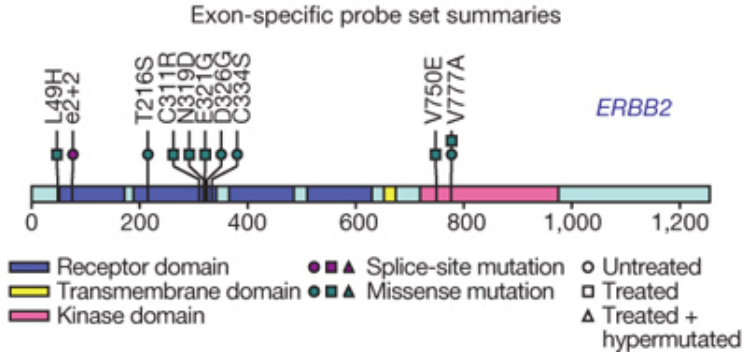
- Very approximate numbers of SNVs and other variants detected by exome/genome sequencing

Germline Mutations in Human Genetics

- Fibrodysplasia Ossificans Progressiva
- Spontaneous or trauma-induced ossification of soft tissue (muscle, tendon, ligament)
- Caused by a specific point mutation in the BMP type I receptor ACVR1, (`c.617G>A`; `p.R206H`)

Somatic Mutations in Cancer





The Cancer Genome Atlas Research Network.

Nature 2008; **455**:1061–1068

- *ERBB2* somatic mutations in glioblastoma tumours.
- Somatic mutations in *ERBB2*, *NF1*, and other genes contribute to the development of glioblastoma

Mutations

I am going to assume you know what the major classes of mutation are: missense, nonsense, insertion, deletion, splice-site mutation. We will now review the mutation nomenclature briefly.

- DNA: A,C,G,T
  - c.435C>A
- Protein: 1- or 3-letter code
  - p.A212P, Ala212Pro
- HGNC[1] gene symbols should be used, e.g., *FBN1* for Fibrillin-1
- Nice Tool for checking mutation nomenclature: `http://www.humgen.nl/mutalyzer/`

[1] HUGO Gene Nomenclature Committee

Human Genome Variation Society (HGVS) Mutation Nomenclature

DNA ...

- Simple subsitution `c.123A>G`
- Deletion `c.123delA`
- Duplication `c.123dupA`
- Insertion `c.123_124insC`

Deletionen & Insertionen

- `c.546delT`
- `c.546del`
- `c.586_591del`
- `c.586_591delTGGTCA` or `c.586_591del6`
- `c.546_547insT` (Not `c.546insT`, since this would be ambiguous)
- `c.1086_1087insGCGTGA`

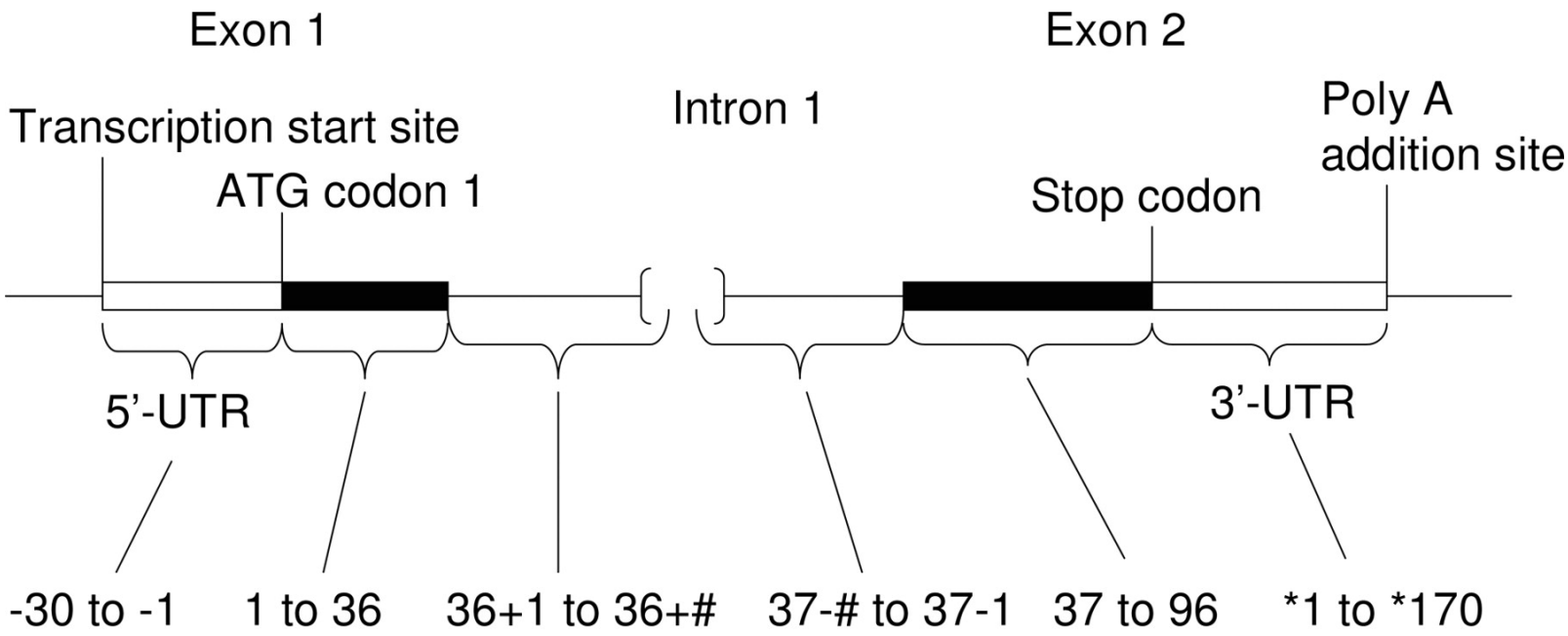
Alleles

- To denote changes in two alleles (e.g., with recessive disease):
- [...], +
- `c.[546C>T]+[2398delT]`

Frameshift

- Short form `p.Arg83fs`
- Alternative: `p.Arg83SerfsX15`
  - First amino-acid subsitution (Arg83Ser)
  - Length of the shifted reading frame until premature stop codon (X15)

Numbering

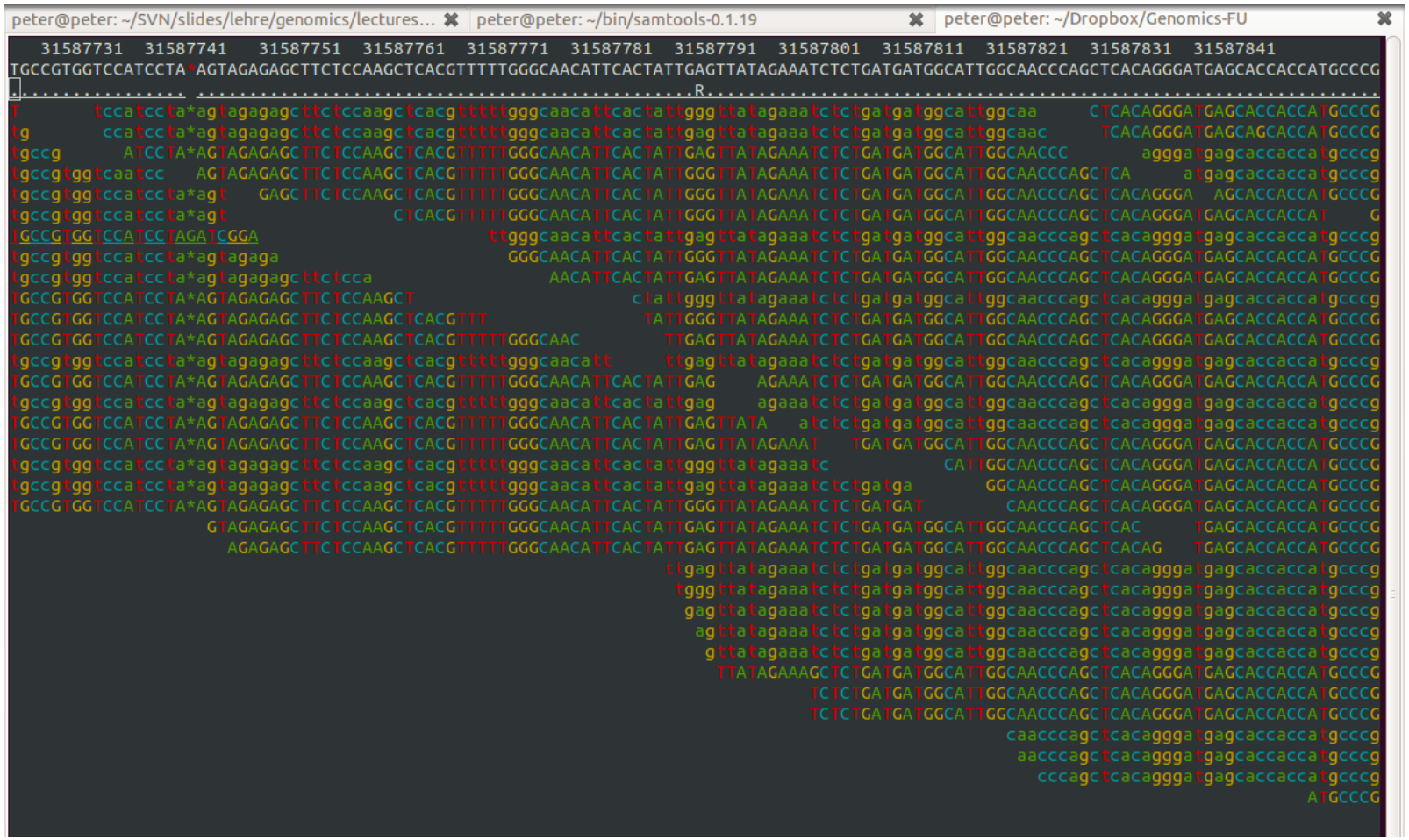


# indicates any positive integer number

- Splice mutations: z.B. 36+1G>C, 37-2A>G

Simple approaches to variant calling

- samtools to sort, index, subset, and display BAM file



Pileups

- Pileup format facilitates SNP/indel calling and brief alignment viewing by eyes.
- Each line consists of chromosome, 1-based coordinate, reference base, the number of reads covering the site, read bases and base qualities.

```
...
21 31587791 T 24 .,..,,.,,.....,..,..,,^], ?EFGDDEEEFEFFEEFDD?EE=>;
21 31587792 G 25 .,..,,.,,.....,..,..,,,^], BCHI9H89IJ7IF78G8I:9I::::
21 31587793 A 26 .,..g,Gg,.G..GG,G.gG.,,g,^], 8G=B6F56GC4BC45I5B76BA?8AA
21 31587794 G 27 .$,$..,,.,,.....,..,..,,,,,^], <D?F9G89GH7HC78F8H:9EC@>BBB
21 31587795 T 26 ..,,.,,.....,..,..,,,,,,^]. ;CEECDEAB@A@AD=@FBC@@>>=>B
...
```

- a dot: match to the reference base on the forward strand
- a comma: match on the reverse strand
- 'ACGTN': for a mismatch on the forward strand
- 'acgtn': for a mismatch on the reverse strand.

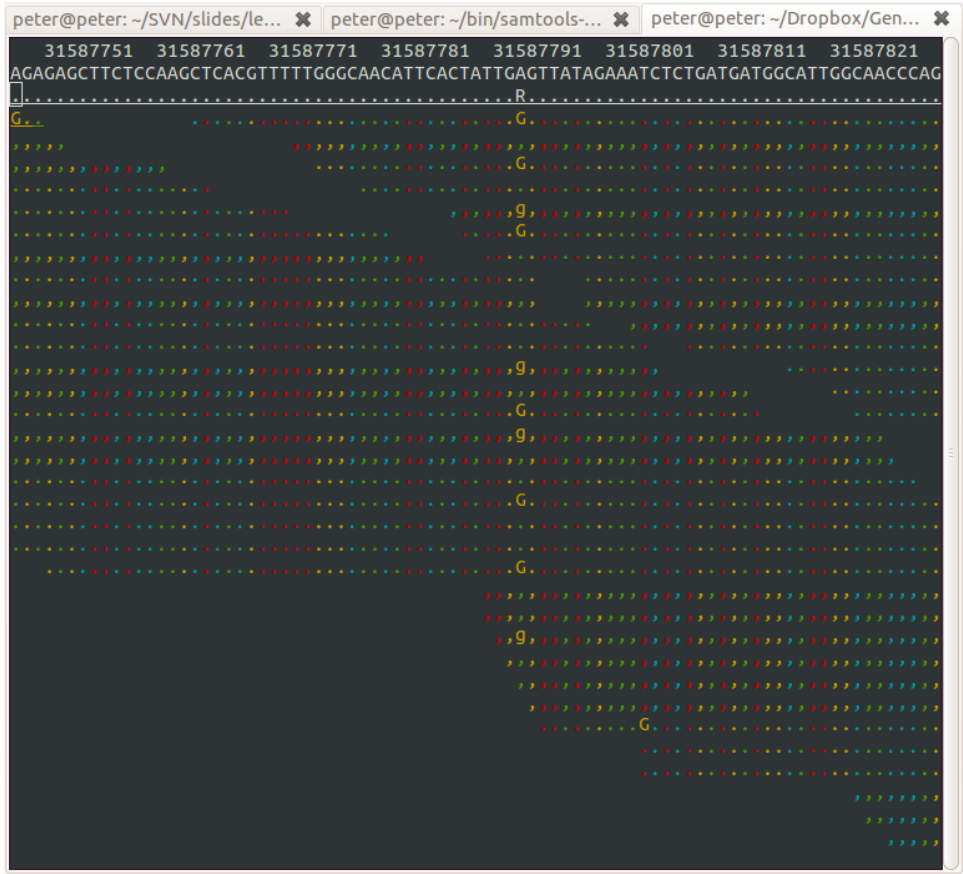
Pileups


```
21 31587794 G 27 .$,$..,.,.,.....,..,..,,,,,^], <D?F9G89GH7HC78F8H:9EC@>BBB
```

- ^: marks the start of a read segment
- The ASCII of the character following ^ minus 33 gives the mapping quality.
- A symbol $ marks the end of a read segment.
- For example, the two $ symbols state that there are two reads whose last base is position 31587794 with the last base being '.', or "G"
- For example, the ^], means that there is a read whose first base is ',' (match on reverse strand), with mapping quality], or 93 minus 33, i.e., 60.

Simple approaches to variant calling


```
21 31587794 G 27 .$,$..,.,.,.....,.,.,.,,,,^], <D?F9G89GH7HC78F8H:9EC@>BBB
```



- Examine the column next to the column with the "G" variants to see the reads corresponding to two $ symbols and one] symbol

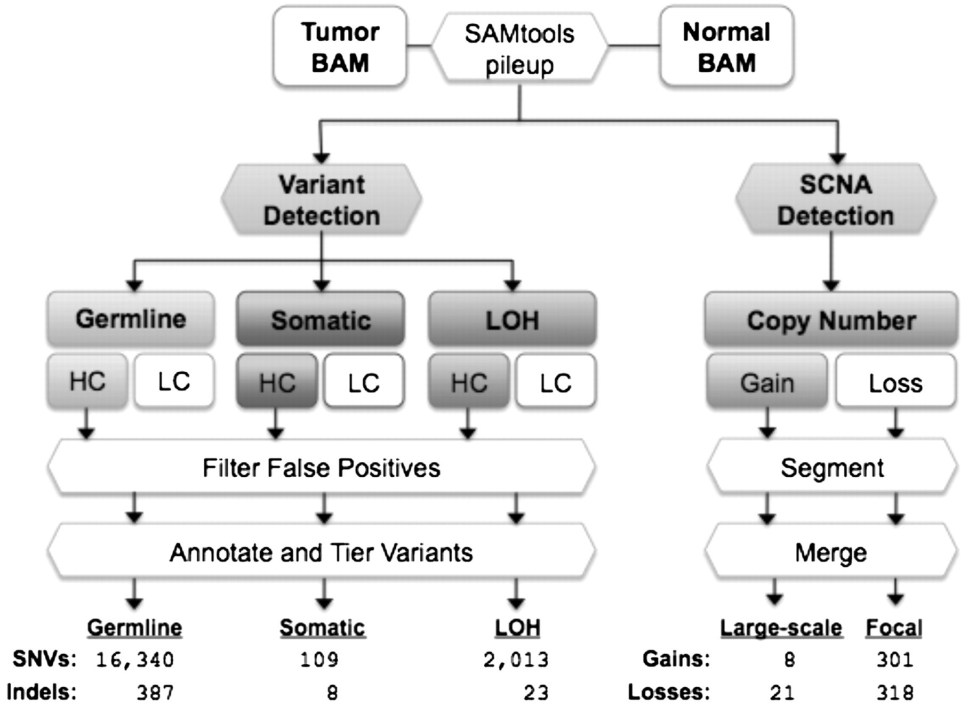
Simple approaches to variant calling

- Let us now consider position 31587793

```
21 31587793 A 26 .,..g,Gg,.G..GG,G.gG.,,g,^], 8G=B6F56GC4BC45I5B76BA?8AA
```

- This position is covered by a total of 26 reads. 16 reads favor the reference A, and 10 reads favor an alternate base, G
- Intuitively, this seems likely to be a heterozygous variant

Simple approaches to variant calling

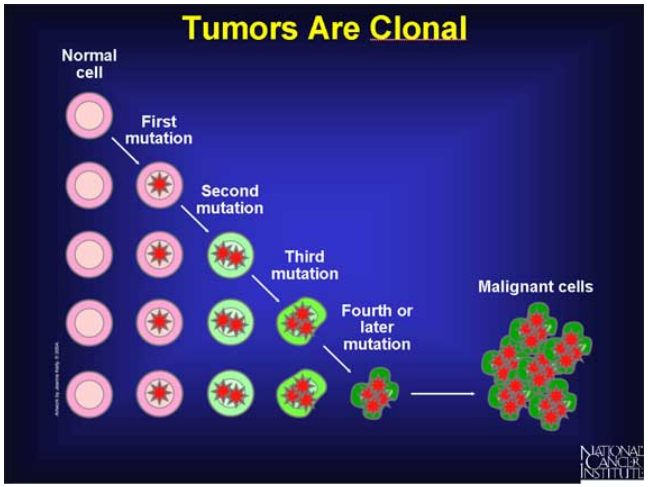


Koboldt DC et al. (2012) VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res* **22**:568-76.

- Let us now examine a heuristic, pileup-based approach towards variant calling implemented by the varscan2 algorithm.

Simple approaches to variant calling: Varscan 2

- Varscan 2 begins with pileup files generatzed for a tumor sample and matched normal control
- A variant detected only in the tumor sample and not in the matched blood sample from the same patient represents a somatic mutation; although many somatic mutations in tumors are "passengers", some are related to the development of cancer, such as the ERBB2 mutations shown earlier.



- Goal: Find mutations specific to the cancer tissue

Simple approaches to variant calling: Varscan 2

- The following steps are performed for each position of genome in parallel for the tumor sample and the matched normal sample
  1. Determine if both samples meet the minimum coverage requirement (by default, three reads with base quality ≥ 20)
  2. Determine a genotype for each sample individually based upon the read bases observed. By default, a variant allele must be supported by at least two independent reads and at least 8% of all reads.
  3. Variants are called homozygous if supported by 75% or more of all reads at a position; otherwise they are called heterozygous.

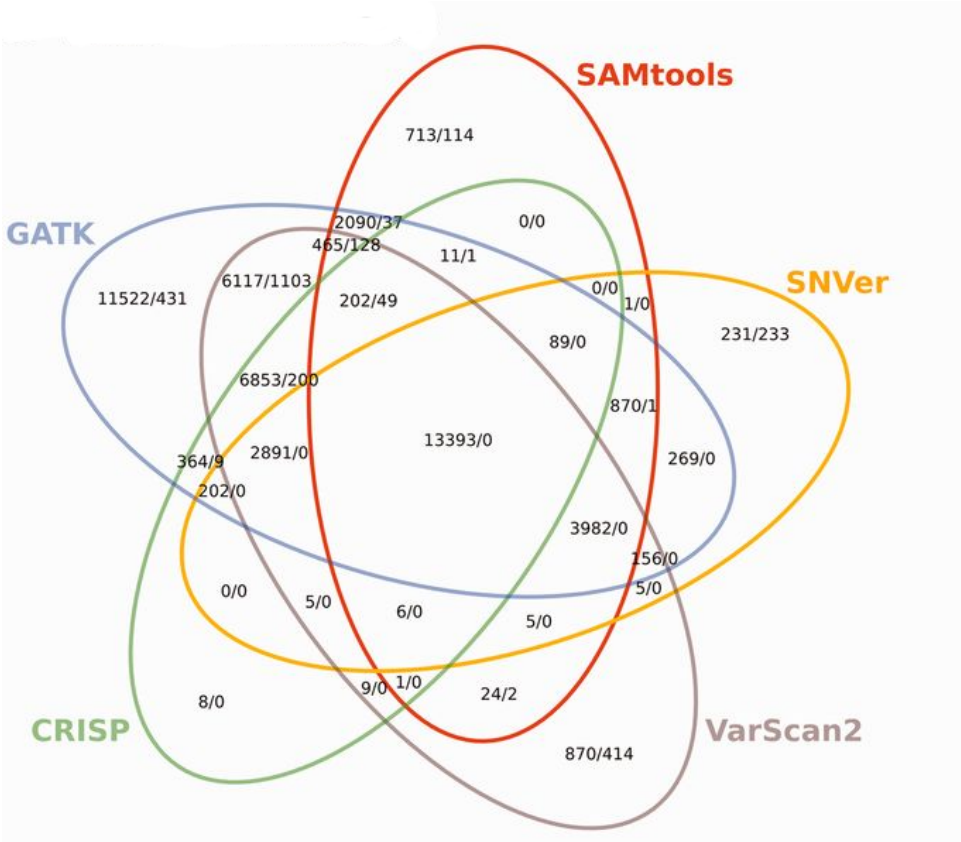
Simple approaches to variant calling: Varscan 2

If the genotypes do not match, then their read counts are evaluated by one-tailed Fisher's exact test in a two-by-two table

	reference	alternate
Tumor reads	Tumor reads 1	Tumor reads 2
Normal reads	Normal reads 1	Normal reads 2

- Fisher exact test is performed. If the P value is significant then
  - if the normal sample is called reference and the tumor sample is called alternate, then the variant is called **somatic**
- We will practice this in the exercise

If it seemed so easy...



Pabinger S et al., A survey of tools for variant analysis of next-generation genome sequencing data. Brief Bioinform. 2013, early access

- Intersection of variants called by different programs.

If it seemed so easy...

Variant calling remains difficult, programs disagree, potentially affecting all downstream analyses

Reasons for seeing a mismatch in a pileup include..

- True variant
- Error from library prep
- misalignment (mapping error)
- error in reference sequence

In addition to this, the meager overlapp in the results of the alignment programs suggests that at least $n - 1$ (and probably all) alignment programs produce partially erroneous variant calls.

Estimating Parameters from Data

In the rest of this lecture, we will review some key concepts of Bayesian statistics that we will need to understand some of the sophisticated variant calling algorithms (and much else in this course). Next time we will look at some of the algorithms in GATK, probably the best current variant caller, in more detail.

Estimating Parameters from Data

In many situations in bioinformatics, we want to estimate "optimal" parameters from data. In the examples we have seen in the lectures on variant calling, these parameters might be the error rate for reads, the proportion of a certain genotype, the proportion of nonreference bases etc. However, the hello world example for this sort of thing is the coin toss, so we will start with that.

Coin toss

Let's say we have two coins that are each tossed 10 times

- Coin 1: H,T,T,H,H,H,T,H,T,T
- Coin 2: T,T,T,H,T,T,T,H,T,T

Intuitively, we might guess that coin one is a fair coin, i.e., $P(X = H) = 0.5$, and that coin 2 is biased, i.e., $P(X = H) \neq 0.5$

Discrete Random Variable

Let us begin to formalize this. We model the coin toss process as follows.

- The outcome of a single coin toss is a random variable X that can take on values in a set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$
- In our example, of course, $n = 2$, and the values are $x_1 = 0$ (tails) and $x_2 = 1$ (heads)
- We then have a probability mass function $p : \mathcal{X} \longrightarrow [0, 1]$; the law of total probability states that $\sum_{x \in \mathcal{X}} p(x_i) = 1$
- This is a Bernoulli distribution with parameter μ:

$$p(X = 1; \mu) = \mu \tag{1}$$

Probability of sequence of events

In general, for a sequence of two events X_1 and X_2, the joint probability is

$$P(X_1, X_2) = p(X_2|X_1)p(X_1) \tag{2}$$

Since we assume that the sequence is iid (identically and independently distributed), by definition $p(X_2|X_1) = P(X_2)$. Thus, for a sequence of n events (coin tosses), we have

$$p(x_1, x_2, \ldots, x_n; \mu) = \prod_{i=1}^{n} p(x_i; \mu) \tag{3}$$

if the probability of heads is 30%, the the probability of the sequence for coin 2 can be calculated as

$$p(T, T, T, H, T, T, T, H, T, T; \mu) = \mu^2(1-\mu)^8 = \left(\frac{3}{10}\right)^2 \left(\frac{7}{10}\right)^8 \tag{4}$$

Probability of sequence of events

Thus far, we have considered $p(x;\mu)$ as a function of x, parametrized by μ. If we view $p(x;\mu)$ as a function of μ, then it is called the **likelihood function**.

Maximum likelihood estimation basically chooses a value of μ that maximizes the likelihood function given the observed data.

Maximum likelihood for Bernoulli

The likelihood for a sequence of i.i.d. Bernoulli random variables $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ with $x_i \in \{0, 1\}$ is then

$$p(\mathbf{X}; \mu) = \prod_{i=1}^{n} p(x_i; \mu) = \prod_{i=1}^{n} \mu^{x_i}(1-\mu)^{1-x_i} \tag{5}$$

We usually maximize the log likelihood function rather than the original function

- Often easier to take the derivative
- the log function is monotonically increasing, thus, the maximum (argmax) is the same
- Avoid numerical problems involved with multiplying lots of small numbers

Log likelihood

Thus, instead of maximizing this

$$p(\mathbf{X};\mu) = \prod_{i=1}^{n} \mu^{x_i}(1-\mu)^{1-x_i} \tag{6}$$

we maximize this

$$\begin{aligned}
\log p(\mathbf{X};\mu) &= \log \prod_{i=1}^{n} \mu^{x_i}(1-\mu)^{1-x_i} \\
&= \sum_{i=1}^{n} \log \left\{ \mu^{x_i}(1-\mu)^{1-x_i} \right\} \\
&= \sum_{i=1}^{n} \left[\log \mu^{x_i} + \log(1-\mu)^{1-x_i} \right] \\
&= \sum_{i=1}^{n} \left[x_i \log \mu + (1-x_i) \log(1-\mu) \right]
\end{aligned}$$

Log likelihood

Note that one often denotes the log likelihood function with the symbol $\mathcal{L} = \log p(\mathbf{X}; \mu)$.

A function f defined on a subset of the real numbers with real values is called monotonic (also monotonically increasing, increasing or non-decreasing), if for all x and y such that $x \leq y$ one has $f(x) \leq f(y)$

Thus, the monotonicity of the log function guarantees that

$$\underset{\mu}{\operatorname{argmax}}\, p(\mathbf{X}; \mu) = \underset{\mu}{\operatorname{argmax}} \log p(\mathbf{X}; \mu) \tag{7}$$

ML estimate

The ML estimate of the parameter μ is then

$$\underset{\mu}{\operatorname{argmax}} \sum_{i=1}^{n} [x_i \log \mu + (1 - x_i) \log(1 - \mu)] \tag{8}$$

We can calculate the argmax by setting the first derivative equal to zero and solving for μ

ML estimate

Thus

$$\begin{aligned}
\frac{\partial}{\partial \mu} \log p(\mathbf{X}; \mu) &= \sum_{i=1}^{n} \frac{\partial}{\partial \mu} \left[x_i \log \mu + (1 - x_i) \log(1 - \mu) \right] \\
&= \sum_{i=1}^{n} x_i \frac{\partial}{\partial \mu} \log \mu + \sum_{i=1}^{n} (1 - x_i) \frac{\partial}{\partial \mu} \log(1 - \mu) \\
&= \frac{1}{\mu} \sum_{i=1}^{n} x_i - \frac{1}{1 - \mu} \sum_{i=1}^{n} (1 - x_i)
\end{aligned}$$

ML estimate

and finally, to find the maximum we set $\frac{\partial}{\partial \mu} \log p(\mathbf{X}; \mu) = 0$:

$$
\begin{aligned}
0 &= \frac{1}{\mu} \sum_{i=1}^{n} x_i - \frac{1}{1-\mu} \sum_{i=1}^{n} (1 - x_i) \\
\frac{1-\mu}{\mu} &= \frac{\sum_{i=1}^{n} (1 - x_i)}{\sum_{i=1}^{n} x_i} \\
\frac{1}{\mu} - 1 &= \frac{\sum_{i=1}^{n} 1}{\sum_{i=1}^{n} x_i} - 1 \\
\frac{1}{\mu} &= \frac{n}{\sum_{i=1}^{n} x_i} \\
\hat{\mu}_{ML} &= \frac{1}{n} \sum_{i=1}^{n} x_i
\end{aligned}
$$

Reassuringly, the maximum likelihood estimate is just the proportion of flips that came out heads.

Problems with ML estimation

Does it really make sense that

- H,T,H,T $\rightarrow \hat{\mu} = 0.5$
- H,T,T,T $\rightarrow \hat{\mu} = 0.25$
- T,T,T,T$\rightarrow \hat{\mu} = 0.0$

ML estimation does not incorporate any prior knowledge and does not generate an estimate of the certainty of its results.

Maximum a posteriori Estimation

Bayesian approaches try to reflect our belief about μ. In this case, we will consider μ to be a random variable.

$$p(\mu|\mathbf{X}) = \frac{p(\mathbf{X}|\mu)p(\mu)}{p(\mathbf{X})} \tag{9}$$

Thus, Bayes' law converts our prior belief about the parameter μ (before seeing data) into a posterior probability, $p(\mu|\mathbf{X})$, by using the likelihood function $p(\mathbf{X}|\mu)$. The maximum a-posteriori (MAP) estimate is defined as

$$\hat{\mu}_{MAP} = \underset{\mu}{\text{argmax}}\, p(\mu|\mathbf{X}) \tag{10}$$

Maximum a posteriori Estimation

Note that because $p(\mathbf{X})$ does not depend on μ, we have

$$\begin{aligned} \hat{\mu}_{MAP} &= \underset{\mu}{\text{argmax}}\, p(\mu|\mathbf{X}) \\ &= \underset{\mu}{\text{argmax}}\, \frac{p(\mathbf{X}|\mu)p(\mu)}{p(\mathbf{X})} \\ &= \underset{\mu}{\text{argmax}}\, p(\mathbf{X}|\mu)p(\mu) \end{aligned}$$

This is essentially the basic idea of the MAP equation used by SNVMix for variant calling

MAP Estimation; What does it buy us?

To take a simple example of a situation in which MAP estimation might produce better results than ML estimation, let us consider a statistician who wants to predict the outcome of the next election in the USA.

- The statistician is able to gather data on party preferences by asking people he meets at the Wall Street Golf Club[2] which party they plan on voting for in the next election
- The statistician asks 100 people, seven of whom answer "Democrats". This can be modeled as a series of Bernoullis, just like the coin tosses.
- In this case, the maximum likelihood estimate of the proportion of voters in the USA who will vote democratic is $\hat{\mu}_{ML} = 0.07$.

[2] i.e., a notorious haven of ultraconservative Republicans

MAP Estimation; What does it buy us?

Somehow, the estimate of $\hat{\mu}_{ML} = 0.07$ doesn't seem quite right given our previous experience that about half of the electorate votes democratic, and half votes republican. But how should the statistician incorporate this prior knowledge into his prediction for the next election?

The MAP estimation procedure allows us to inject our prior beliefs about parameter values into the new estimate.

Beta distribution: Background

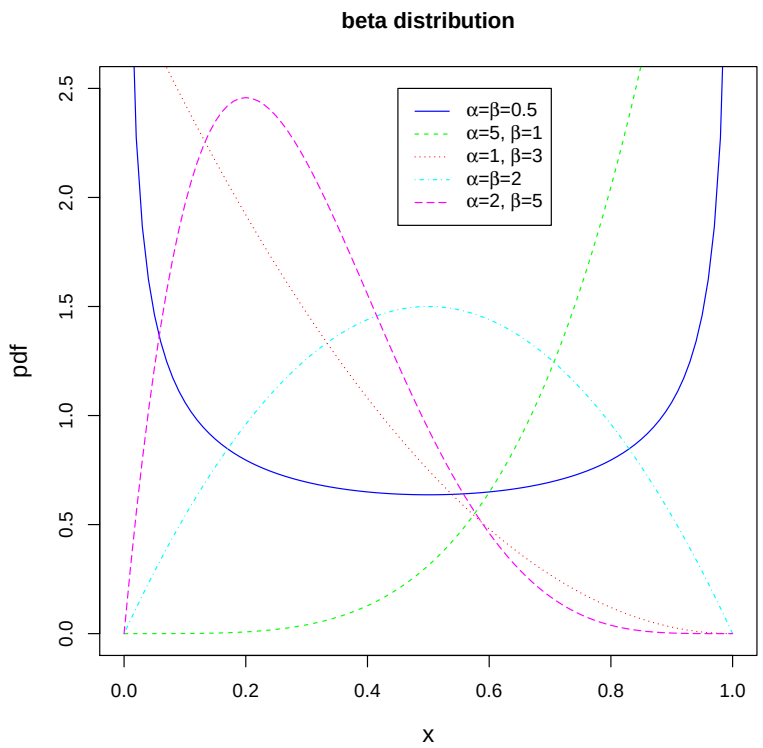
The Beta distribution is appropriate to express prior belief about a Bernoulli distribution. The Beta distribution is a family of continuous distributions defined on $[0, 1]$ and parametrized by two positive shape parameters, α and β

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

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

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

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



Beta distribution: Background

Random variables are either discrete (i.e., they can assume one of a list of values, like the Bernoulli with heads/tails) or continuous (i.e., they can take on any numerical value in a certain interval, like the Beta distribution with μ).

- A probability density function (pdf) of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value, i.e., $p(\mu) : \mathbb{R} \rightarrow \mathbb{R}^+$ such that

$$\Pr(\mu \in (a, b)) = \int_a^b p(\mu) d\mu \tag{11}$$

The probability that the value of μ lies between a and b is given by integrating the pdf over this region

Beta distribution: Background

Recall the difference between a PDF (for continuous random variable) and a probability mass function (PMF) for a discrete random variable

- A PMF is defined as $\Pr(X = x_i) = p_i$, with $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$
- e.g., for a fair coin toss (Bernoulli), $\Pr(X = \text{heads}) = \Pr(X = \text{tails}) = 0.5$
- In contrast, for a PDF, there is no requirement that $p(\mu) \leq 1$, but we do have

$$\int_{-\infty}^{+\infty} p(\mu) d\mu = 1 \tag{12}$$

Beta distribution: Background

- We calculate the PDF for the Beta distribution for a sequence of values $0, 0.01, 0.02, \ldots, 1.00$ in R as follows

```
x <- seq(0.0, 1.0, 0.01)
y <- dbeta(x, 3, 3)
```

- Recalling how to approximate an integral with a Riemann sum, $\int_a^b p(\mu)d\mu \approx \sum_{i=1}^{n} p(\mu_i)\Delta_i$, where μ_i is a point in the subinterval Δ_i and the subintervals span the entire interval $[a, b]$, we can check that $\int_0^1 \mathrm{Beta}(\mu)d\mu = 1$

```
> sum ((1/101)*y)
[1] 0.990099
```

Here, $\Delta_i = \frac{1}{101}$ and the vector y contains the various values of μ_i

Beta distribution: Background

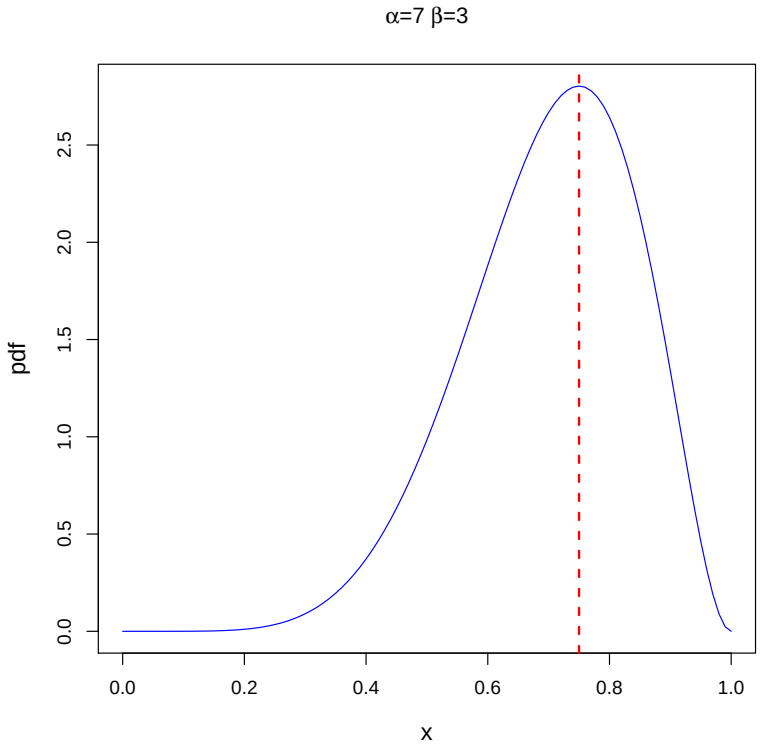
- The mode of a continuous probability distribution is the value x at which its probability density function has its maximum value
- The mode of the $\mathrm{Beta}(\alpha, \beta)$ distribution has its mode at

$$\frac{\alpha - 1}{\alpha + \beta - 2} \tag{13}$$

```
alpha <- 7
beta <- 3
x <- seq(0.0, 1.0, 0.01)
y <- dbeta(x, alpha, beta)
md <- (alpha-1)/(alpha + beta - 2)
title <- expression(paste(alpha,"=7 ",beta,"=3"))
plot(x, y, type="l",main=title,
xlab="x",ylab="pdf",col="blue",lty=1,cex.lab=1.25)
abline(v=md,col="red",lty=2,lwd=2)
```

Beta distribution: Background

The code from the previous slide leads to



The mode, shown as the dotted red line, is calculated as $\frac{\alpha - 1}{\alpha + \beta - 2} = \frac{7 - 1}{7 + 3 - 2} = 0.75$

Maximum a posteriori (MAP) Estimation

With all of this information in hand, let's get back to MAP estimation!

- Going back to Bayes rule, again, we seek the value of μ that maximizes the posterior $\Pr(\mu|\mathbf{X})$:

$$\Pr(\mu|\mathbf{X}) = \frac{\Pr(\mathbf{X}|\mu)\Pr(\mu)}{\Pr(\mathbf{X})} \tag{14}$$

Maximum a posteriori (MAP) Estimation

We then have

$$
\begin{aligned}
\hat{\mu}_{MAP} &= \underset{\mu}{\text{argmax}} \Pr(\mu|\mathbf{X}) \\
&= \underset{\mu}{\text{argmax}} \frac{\Pr(\mathbf{X}|\mu)\Pr(\mu)}{\Pr(\mathbf{X})} \\
&= \underset{\mu}{\text{argmax}} \Pr(\mathbf{X}|\mu)\Pr(\mu) \\
&= \underset{\mu}{\text{argmax}} \prod_{x_i \in \mathbf{X}} \Pr(x_i|\mu)\Pr(\mu)
\end{aligned}
$$

Maximum a posteriori (MAP) Estimation

As we saw above for maximum likelihood estimation, it is easier to calculate the argmax for the logarithm

$$\begin{aligned}
\underset{\mu}{\operatorname{argmax}} \Pr(\mu|\mathbf{X}) &= \underset{\mu}{\operatorname{argmax}} \log \Pr(\mu|\mathbf{X}) \\
&= \underset{\mu}{\operatorname{argmax}} \log \prod_{x_i \in \mathbf{X}} \Pr(x_i|\mu) \cdot \Pr(\mu) \\
&= \underset{\mu}{\operatorname{argmax}} \sum_{x_i \in \mathbf{X}} \{\log \Pr(x_i|\mu)\} + \log \Pr(\mu)
\end{aligned}$$

Maximum a posteriori (MAP) Estimation

Let's go back now to our problem of predicting the results of the next election. Essentially, we plug in the equations for the distributions of the likelihood (a Bernoulli distribution) and the prior (A Beta distribution).

$$\Pr(\mu|\mathbf{X}) \propto \Pr(x_i|\mu) \cdot \Pr(\mu)$$

- posterior
- Likelihood (Bernoulli)
- prior (Beta)

Maximum a posteriori (MAP) Estimation

We thus have that

- $\Pr(x_i|\mu) = \text{Bernoulli}(x_i|\mu) = \mu^{x_i}(1-\mu)^{1-x_i}$
- $\Pr(\mu) = \text{Beta}(\mu|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} \cdot \mu^{\alpha-1}(1-\mu)^{\beta-1}$

thus

$$\Pr(\mu|\mathbf{X}) \propto \Pr(\mathbf{X}|\mu)\Pr(\mu)$$

is equivalent to

$$\Pr(\mu|\mathbf{X}) \propto \left\{ \prod_i \text{Bernoulli}(x_i|\mu) \right\} \cdot \text{Beta}(\mu|\alpha,\beta) \qquad (15)$$

Maximum a posteriori (MAP) Estimation

Furthermore

$$\begin{aligned}\mathcal{L} &= \log \Pr(\mu|\mathbf{X}) \\ &= \log \left\{ \prod_i \text{Bernoulli}(x_i|\mu) \right\} \cdot \text{Beta}(\mu|\alpha,\beta) \\ &= \sum_i \log \text{Bernoulli}(x_i|\mu) + \log \text{Beta}(\mu|\alpha,\beta)\end{aligned}$$

We solve for $\hat{\mu}_{MAP} = \text{argmax}_\mu \mathcal{L}$ as follows

$$\underset{\mu}{\text{argmax}} \sum_i \log \text{Bernoulli}(x_i|\mu) + \log \text{Beta}(\mu|\alpha,\beta)$$

Note that this is almost the same as the ML estimate except that we now have an additional term resulting from the prior

Maximum a posteriori (MAP) Estimation

Again, we find the maximum value of μ by setting the first derivative of $\mathcal{L}$ equal to zero and solving for μ

$$\frac{\partial}{\partial \mu}\mathcal{L} = \sum_i \frac{\partial}{\partial \mu} \log \text{Bernoulli}(x_i|\mu) + \frac{\partial}{\partial \mu} \log \text{Beta}(\mu|\alpha,\beta)$$

The first term is the same as for ML[3], i.e.

$$\sum_i \frac{\partial}{\partial \mu} \log \text{Bernoulli}(x_i|\mu) = \frac{1}{\mu}\sum_{i=1}^{n} x_i - \frac{1}{1-\mu}\sum_{i=1}^{n}(1-x_i) \quad (16)$$

[3] see slide 11

Maximum a posteriori (MAP) Estimation

To find the second term, we note

$$\begin{aligned}
\frac{\partial}{\partial\mu}\log \mathrm{Beta}(\mu|\alpha,\beta) &= \frac{\partial}{\partial\mu}\log\left\{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)}\cdot\mu^{\alpha-1}(1-\mu)^{\beta-1}\right\} \\
&= \frac{\partial}{\partial\mu}\log\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)} + \frac{\partial}{\partial\mu}\log\mu^{\alpha-1}(1-\mu)^{\beta-1} \\
&= 0 + \frac{\partial}{\partial\mu}\log\mu^{\alpha-1}(1-\mu)^{\beta-1} \\
&= (\alpha-1)\frac{\partial}{\partial\mu}\log\mu + (\beta-1)\frac{\partial}{\partial\mu}\log(1-\mu) \\
&= \frac{\alpha-1}{\mu} - \frac{\beta-1}{1-\mu}
\end{aligned}$$

Maximum a posteriori (MAP) Estimation

To find $\hat{\mu}_{MAP}$, we now set $\frac{\partial}{\partial \mu}\mathcal{L} = 0$ and solve for μ

$$\begin{aligned} 0 &= \frac{\partial}{\partial \mu}\mathcal{L} \\ &= \frac{1}{\mu}\sum_{i=1}^{n} x_i - \frac{1}{1-\mu}\sum_{i=1}^{n}(1-x_i) + \frac{\alpha-1}{\mu} - \frac{\beta-1}{1-\mu} \end{aligned}$$

and thus

$$\begin{aligned} \mu\left[\sum_{i=1}^{n}(1-x_i) + \beta - 1\right] &= (1-\mu)\left[\sum_i x_i + \alpha - 1\right] \\ \mu\left[\sum_{i=1}^{n}(1-x_i) + \sum_i x_i + \beta - 1 + \alpha - 1\right] &= \sum_i x_i + \alpha - 1 \\ \mu\left[\sum_{i=1}^{n} 1 + \beta + \alpha - 2\right] &= \sum_i x_i + \alpha - 1 \end{aligned}$$

Maximum a posteriori (MAP) Estimation

Finally, if we let our Bernoulli distribution be coded as `Republican=1` and `Democrat=0`, we have that

$$\sum_i x_i = n_r \quad \text{where } n_r \text{ denotes the number of Republican voters} \tag{17}$$

Then,

$$\begin{aligned} \mu\left[\sum_{i=1}^{n} 1 + \beta + \alpha - 2\right] &= \sum_i x_i + \alpha - 1 \\ \mu\left[n + \beta + \alpha - 2\right] &= n_R + \alpha - 1 \end{aligned}$$

and finally

$$\hat{\mu}_{MAP} = \frac{n_R + \alpha - 1}{n + \beta + \alpha - 2} \tag{18}$$

And now our prediction

It is useful to compare the ML and the MAP predictions. Note again that α and β are essentially the same thing as pseudo-counts, and the higher their value, the more the prior affects the final prediction (i.e., the posterior).

Recall that in our poll of 100 members of the Wall Street Golf club, only seven said they would vote democratic. Thus

- $n = 100$
- $n_r = 93$
- We will assume that the mode of our prior belief is that 50% of the voters will vote democratic, and 50% republican. Thus, $\alpha = \beta$. However, different values for alpha and beta express different strengths of prior belief

And now our prediction

n	n_R	α	β	$\hat{\mu}_{ML}$	$\hat{\mu}_{MAP}$
100	93	1	1	0.93	0.93
100	93	5	5	0.93	0.90
100	93	100	100	0.93	0.64
100	93	1000	1000	0.93	0.52
100	93	10000	10000	0.93	0.502

Thus, MAP "pulls" the estimate towards the prior to an extent that depends on the strength of the prior

The use of MAP in MAQ

Recall from the lecture that we 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.

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

MAQ: Consensus Genotype Calling

Note that MAQ does not attempt to learn the parameters, rather it uses user-supplied parameter r which roughly corresponds to μ in the election.

MAQ calls the the genotype with the highest posterior probability:

$$\hat{g} = \underset{g \in (\langle a,a \rangle, \langle a,b \rangle, \langle b,b \rangle)}{\text{argmax}} p(g|D)$$

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

MAQ: Consensus Genotype Calling

A major problem in SNV calling is false positive heterozygous variants. It seems less probable to observe a heterozygous call at a position with a common SNP in the population. For this reason, MAQ uses a different prior (r) for previously known SNPs ($r = 0.2$) and "new" SNPs ($r = 0.001$).

Let us examine the effect of these two priors on variant calling. In R, we can write

$$p(G = \langle a, b \rangle | D) \propto \binom{n}{k} \frac{1}{2^n} \cdot r$$

as

```
> dbinom(k,n,0.5) * r
```

where k is the number of non-reference bases, n is the total number of bases, and 0.5 corresponds to the probability of seeing a non-ref base given that the true genotype is heterozygous.

MAQ: Consensus Genotype Calling

A major problem in SNV calling is false positive heterozygous variants. It seems less probable to observe a heterozygous call at a position with a common SNP in the population. For this reason, MAQ uses a different prior (r) for previously known SNPs ($r = 0.2$) and "new" SNPs ($r = 0.001$).

Let us examine the effect of these two priors on variant calling. In R, we can write

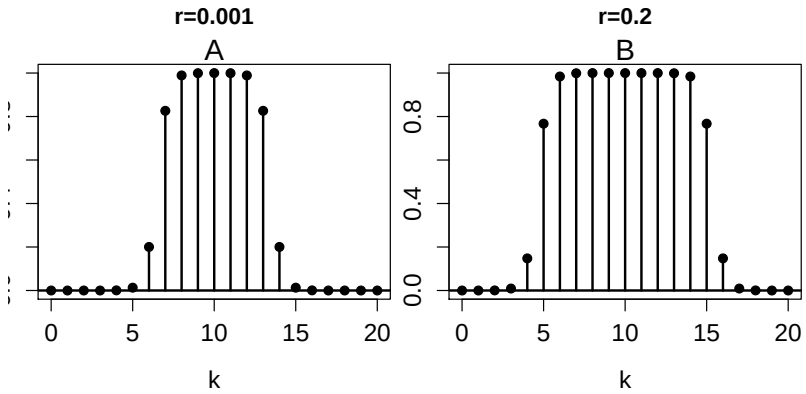
$$p(G = \langle a, b \rangle | D) \propto \binom{n}{k} \frac{1}{2^n} \cdot r$$

as

```
> dbinom(k,n,0.5) * r
```

where k is the number of non-reference bases, n is the total number of bases, and 0.5 corresponds to the probability of seeing a non-ref base given that the true genotype is heterozygous.

MAQ: Consensus Genotype Calling



- The figures show the posterior for the heterozygous genotype according to the simplified MAQ algorithm discussed in the previous lecture
- The prior $r = 0.0001$ means than positions with 5 or less ALT bases do not get called as heterozygous, whereas the prior with $r = 0.2$ means that positions with 5 bases do get a het call

MAQ: Consensus Genotype Calling

What have we learned?

- Get used to Bayesian techniques important in many areas of bioinformatics
- understand the difference between ML and MAP estimation
- understand the Beta function, priors, pseudocounts
- Note that MAQ is no longer a state of the art algorithm, and its use of the MAP framework is relatively simplistic
- Nonetheless, a good introduction to this topic, and we will see how these concepts are used in EM today