Variant Calling

Michael Schatz

Sept 23, 2019

Lecture 7: Computational Biomedical Research

Assignment 3: Sequence Alignment
Due Monday Sept 30 @ 11:59pm

Assignment 3: Sequence Alignment

Assignment Date: Wednesday, Sept 18, 2019
Due Date: Wednesday, Sept. 25, 2019 @ 11:59pm

Assignment Overview

In this assignment you will consider the requirements for sequence alignment. As a reminder, any questions about the assignment should be posted to Piazza

Question 1: Minimum Alignment Lengths (10 pts)

Determine how many bases long a given pattern P should be to ensure that occurrences of P are unlikely to be chance events (e<.000001) in genomes of the following sizes:

- 1a. 5.2Mb (Bacillus anthracis – the microbe that causes anthrax)
- 1b. 100Mb (Caenorhabditis elegans - model worm)
- 1c. 3.1Gb (Homo sapiens - human)
- 1d. 18GB (Triticum aestivum – bread wheat)
- 1e. 670Gb (Polychaos dubium – amoeba, has largest known genome)

Question 2: Edit distance (10 pts)

Compute the edit distance of (a portion of) the human hemoglobin alpha and beta subunits, showing the dynamic programming matrix and the aligned sequences. Assume a fixed unit cost to substitute one amino acid for another.

```
Alpha:  EALERRFLSFPTTKTYFPHFDLSHGSAQVK
Beta:   EALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVK
```

Packaging

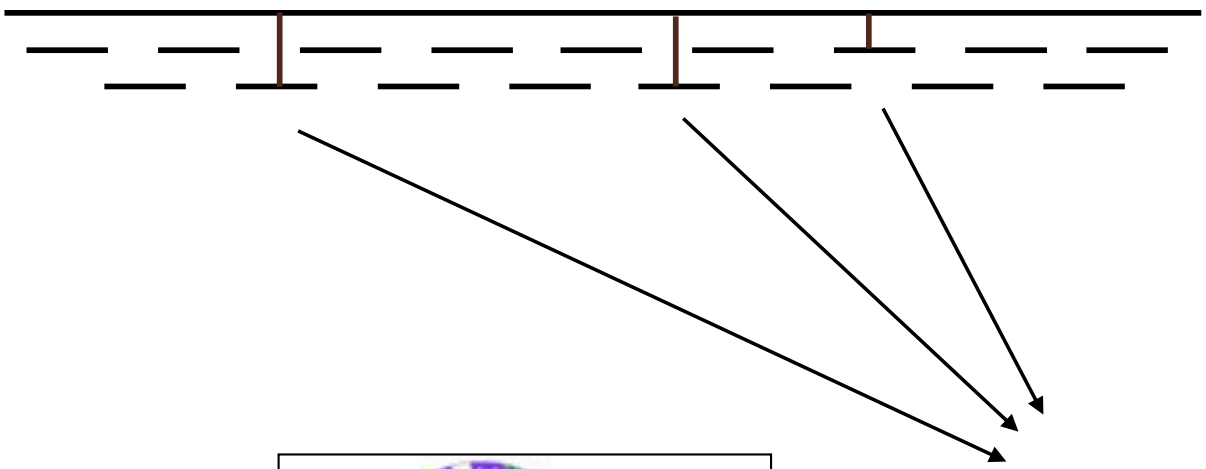
The solutions to the above questions should be submitted as a single PDF document that includes your name, email address, and all relevant figures (as needed). Make

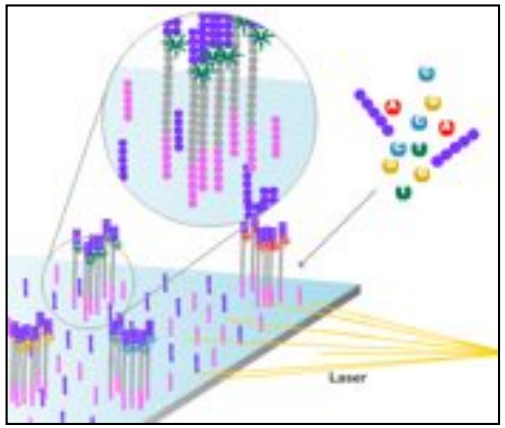
https://github.com/schatzlab/biomedicalresearch2019

Part 1: Recap

Personal Genomics

How does your genome compare to the reference?





Heart Disease

Cancer

Presidential Smile

Brute Force Analysis

- Brute Force:
  - At every possible offset in the genome:
    - Do all of the characters of the query match?

- Analysis
  - Simple, easy to understand
  - Genome length = n [3B]
  - Query length = m [7]
  - Comparisons: (n-m+1) * m [21B]

- Overall runtime: O(nm)

[How long would it take if we double the genome size, read length?]

[How long would it take if we double both?]

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC
    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 11; Mid = (9+11)/2 = 10
  - Middle = Suffix[10] = GATTACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 9; Mid = (9+9)/2 = 9
  - Middle = Suffix[9] = GATTACA…
    => Match at position 2!

	#	Sequence	Pos
	1	ACAGATTACC…	6
	2	ACC…	13
	3	AGATTACC…	8
	4	ATTACAGATTACC…	3
	5	ATTACC…	10
	6	C…	15
	7	CAGATTACC…	7
Lo	8	CC…	14
Hi	9	GATTACAGATTACC…	2
	10	GATTACC…	9
	11	TACAGATTACC…	5
	12	TACC…	12
	13	TGATTACAGATTACC…	1
	14	TTACAGATTACC…	4
	15	TTACC…	11

Binary Search Analysis

- Binary Search

  Initialize search range to entire list

  mid = (hi+lo)/2; middle = suffix[mid]

  if query matches middle: done

  else if query < middle: pick low range

  else if query > middle: pick hi range

  Repeat until done or empty range [WHEN?]

- Analysis
  - More complicated method
  - How many times do we repeat?
    - How many times can it cut the range in half?
    - Find smallest x such that: $n/(2^x) \leq 1$; $x = \lg_2(n)$ [32]

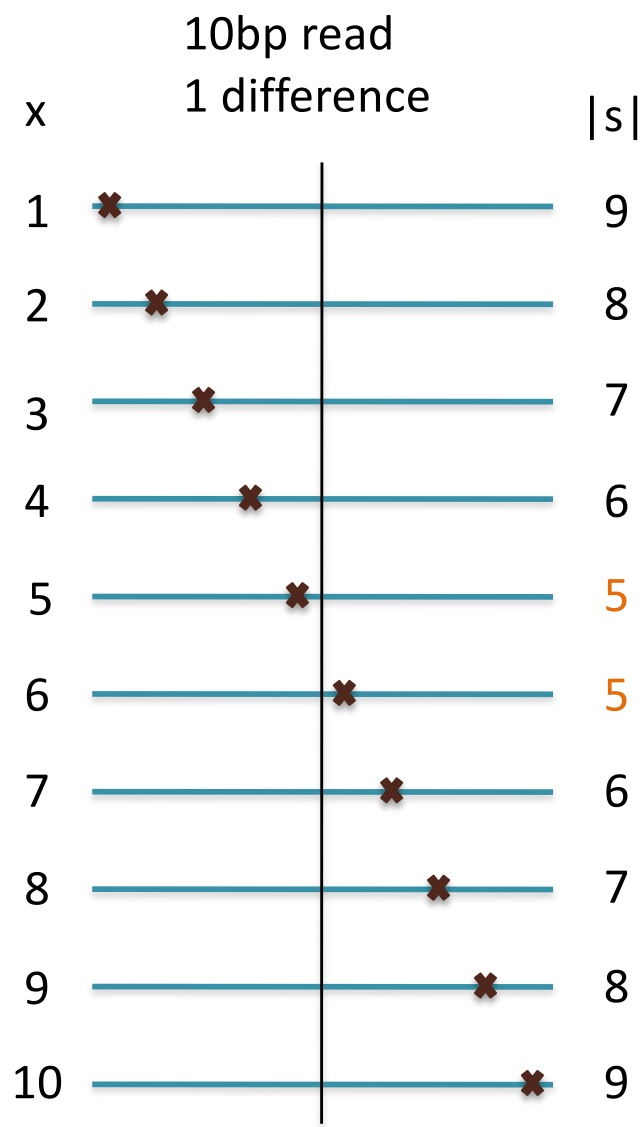
- Total Runtime: O(m lg n)
  - More complicated, but much faster!
  - Looking up a query loops 32 times instead of 3B

[How long does it take to search 6B or 24B nucleotides?]

Seed-and-Extend Alignment

Theorem: An alignment of a sequence of length *m* with at most *k* differences ***must*** contain an exact match at least $s=m/(k+1)$ bp long

(*Baeza-Yates* and Perleberg, 1996)



- Proof: Pigeonhole principle
  - 1 pigeon can't fill 2 holes
- Seed-and-extend search
  - Use an index to rapidly find short exact alignments to seed longer in-exact alignments
    - BLAST, MUMmer, Bowtie, BWA, SOAP, …
  - Specificity of the depends on seed length
    - Guaranteed sensitivity for k differences
    - Also finds some (but not all) lower quality alignments <- heuristic

Similarity metrics

- Hamming distance
  - Count the number of substitutions to transform one string into another

```
MIKESCHATZ
||X||XXXX|
MICESHATZZ
     5
```

- Edit distance
  - The minimum number of substitutions, insertions, or deletions to transform one string into another

```
MIKESCHAT-Z
||X||X|||X|
MICES-HATZZ
     3
```

Edit Distance Example

AGCACACA → ACACACTA in 4 steps

AGCACACA → (1. change G to C)
ACCACACA → (2. delete C)
ACACACA → (3. change A to T)
ACACACT → (4. insert A after T)
ACACACTA → done

[Is this the best we can do?]

Edit Distance Example

AGCACACA → ACACACTA in 3 steps

AGCACACA → (1. change G to C)
ACCACACA → (2. delete C)
ACACACA → (3. insert T after 3rd C)
ACACACTA → done

[Is this the best we can do?]

Reverse Engineering Edit Distance

`D(AGCACACA, ACACACTA) = ?`

Imagine we already have the optimal alignment of the strings, the last column can only be 1 of 3 options:

```
…M    …I    …D
…A    …-    …A
…A    …A    …-
```

The optimal alignment of last two columns is then 1 of 9 possibilities

```
…MM  …IM  …DM    …MI  …II  …DI    …MD  …ID  …DD
…CA  …-A  …CA    …A-  …--  …A-    …CA  …-A  …CA
…TA  …TA  …-A    …TA  …TA  …-A    …A-  …A-  …--
```

The optimal alignment of the last three columns is then 1 of 27 possibilities…

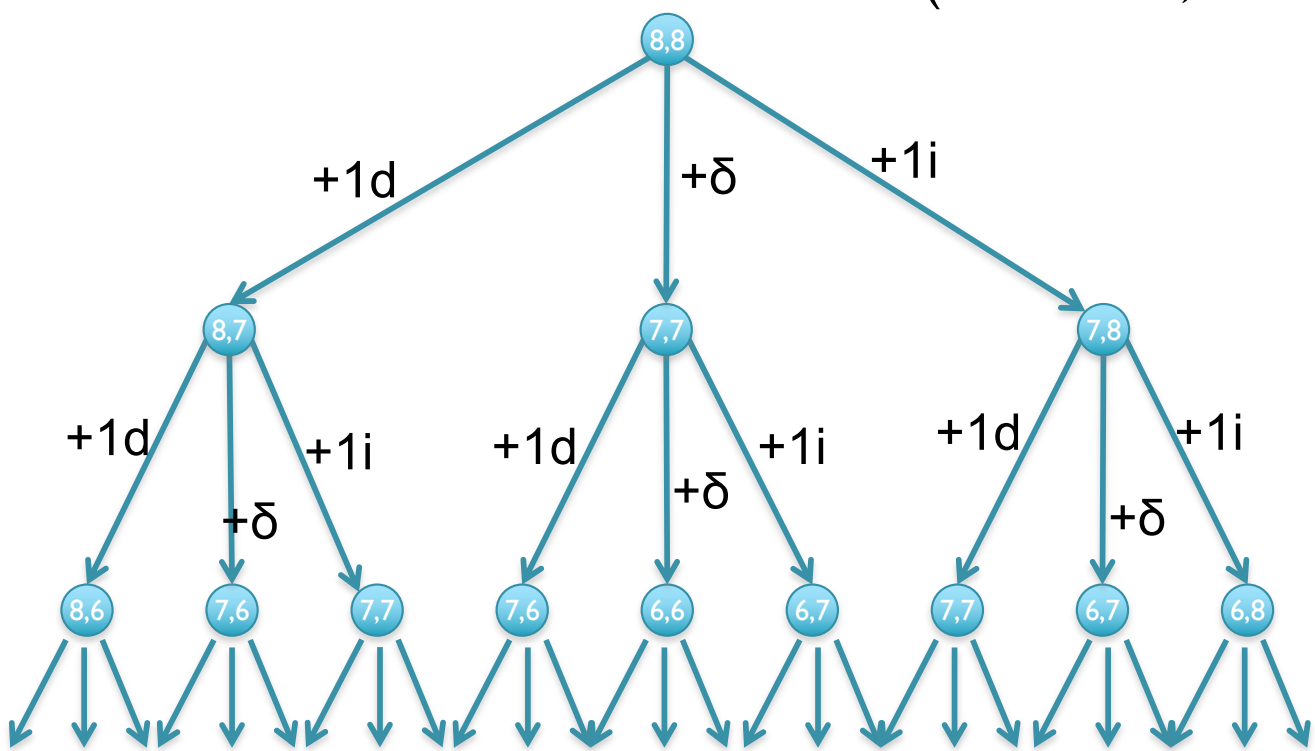
```
…M…    …I…    …D…
…X…    …-…    …X…
…Y…    …Y…    …-…
```

Eventually spell out every possible sequence of {I,M,D}

Recursive solution

- Computation of D is a recursive process.
  - At each step, we only allow matches, substitutions, and indels
  - D(i,j) in terms of D(i', j') for i' ≤ i and j' ≤ j.

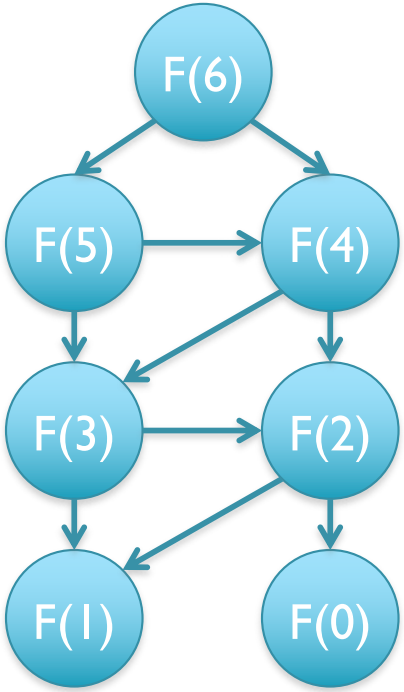
D(AGCACACA, ACACACTA) = min{D(AGCACACA, ACACACT) + 1,
D(AGCACAC, ACACACTA) + 1,
D(AGCACAC, ACACACT) +δ(A, A)}



[What is the running time?]

Dynamic Programming

- General approach for solving (some) complex problems
  - When applicable, the method takes far less time than naive methods.
    - Polynomial time ($O(n)$ or $O(n^2)$ instead of exponential time ($O(2^n)$ or $O(3^n)$)
- Requirements:
  - **Overlapping subproblems**
  - **Optimal substructure**



- Applications:
  - Fibonacci
  - Longest Increasing Subsequence (Bonus Slides!)
  - Sequence alignment, Dynamic Time Warp, Viterbi
- Not applicable:
  - Traveling salesman problem, Clique finding, Subgraph isomorphism, …
  - The cheapest flight from airport A to airport B involves a single connection through airport C, but the cheapest flight from airport A to airport C involves a connection through some other airport D.

Dynamic Programming

- We could code this as a recursive function call...

  ...with an exponential number of function evaluations

- There are only (n+1)x(m+1) pairs i and j
  - We are evaluating D(i,j) multiple times

- Compute D(i,j) bottom up.
  - Start with smallest (i,j) = (1,1).
  - Store the intermediate results in a table.
    - Compute D(i,j) *after* D(i-1,j), D(i,j-1), and D(i-1,j-1)

Recurrence Relation for D

Find the edit distance (minimum number of operations to convert one string into another) in O(mn) time

- Base conditions:
  - $D(i,0) = i$, for all $i = 0,\ldots,n$
  - $D(0,j) = j$, for all $j = 0,\ldots,m$

- For $i > 0$, $j > 0$:

$$
\begin{aligned}
D(i,j) = \min \{ & \\
& D(i-1,j) + 1, && \text{// align 0 chars from S, 1 from T} \\
& D(i,j-1) + 1, && \text{// align 1 chars from S, 0 from T} \\
& D(i-1,j-1) + \delta(S(i),T(j)) && \text{// align 1+1 chars} \\
\} &
\end{aligned}
$$

[Why do we want the min?]

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1								
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

[What does the initialization mean?]

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0							
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

$$D[A,A] = \min\{D[A,]+1,\ D[,A]+1,\ D[,]+\delta(A,A)\}$$

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1						
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

$$D[A,AC] = \min\{D[A,A]+1,\ D[,AC]+1,\ D[,A]+\delta(A,C)\}$$

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2					
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[A,ACA] = min{D[A,AC]+1, D[,ACA]+1, D[,AC]+δ(A,A)}

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[A,ACACACTA] = 7

```
-------A
*******|
ACACACTA
```

[What about the other A?]

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[AG,ACACACTA] = 7

```
----AG--
****|***
ACACACTA
```

Dynamic Programming Matrix

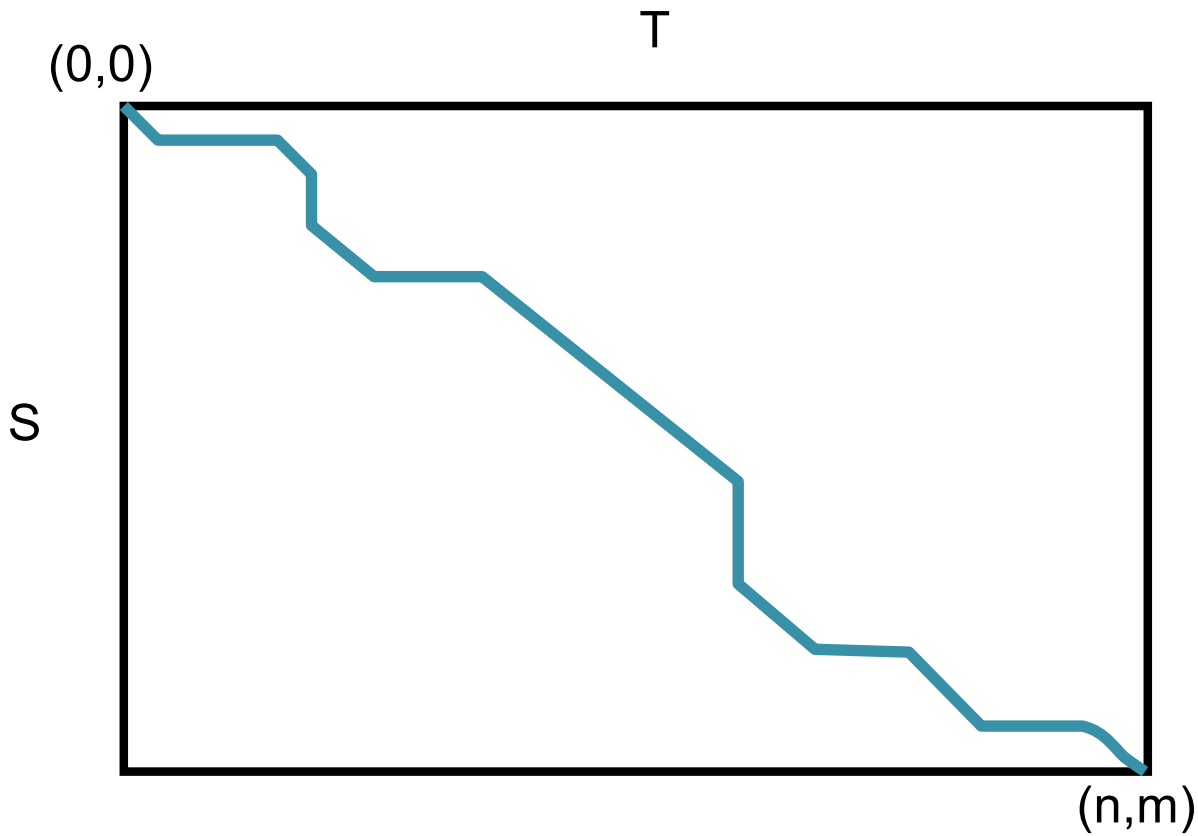
		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3	2	1	2	2	3	4	5	6
A	4	3	2	1	2	2	3	4	5
C	5	4	3	2	1	2	2	3	4
A	6	5	4	3	2	1	2	3	3
C	7	6	5	4	3	2	1	2	3
A	8	7	6	5	4	3	2	2	2

D[AGCACACA,ACACACTA] = 2

```
AGCACAC-A
|*|||||*|
A-CACACTA
```

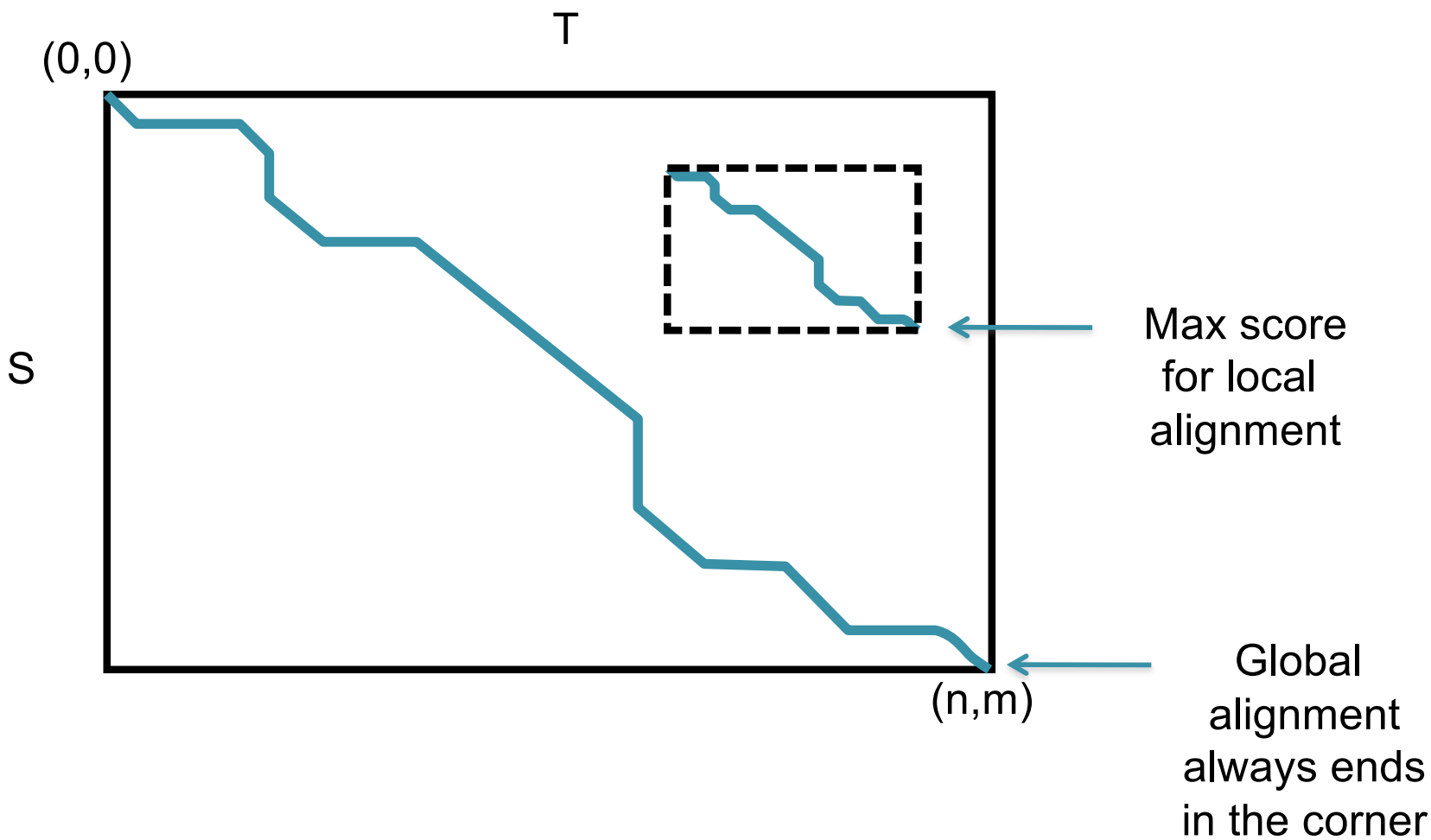
[Can we do it any better?]

Global Alignment Schematic



- A high quality alignment will stay close to the diagonal
  - If we are only interested in high quality alignments, we can skip filling in cells that can't possibly lead to a high quality alignment
  - Find the global alignment with at most edit distance d: O(2dn)

Global vs Local Alignment Schematic



Nathan Edwards

Local vs. Global Alignment (cont' d)

- Global Alignment

```
--T—-CC-C-AGT—-TATGT-CAGGGGACACG—A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || | | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG—T-CAGAT--C
```

- Local Alignment—better alignment to find conserved segment

```
                tccCAGTTATGTCAGgggacacgagcatgcagagac
                   |||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Alignment Ambiguity

Notice that the edit distance of GATTTACA and GATACA is 2,
but there are multiple possible optimal alignments:

```
GATTTACA      GATTTACA      GATTTACA
GAT--ACA      GA-T-ACA      GA--TACA
```

Which one is right?

Algorithm Overview

1. Split read into segments

Read

CCAGTAGCTCTCAGCCTTATTTTACCCAGGCCTGTA

Read (reverse complement)

TACAGGCCTGGGTAAAATAAGGCTGAGAGCTACTGG

Policy: extract 16 nt seed every 10 nt

Seeds

+, 0: CCAGTAGCTCTCAGCC
+, 10: TCAGCCTTATTTTACC
+, 20: TTTACCCAGGCCTGTA
-, 0: TACAGGCCTGGGTAAA
-, 10: GGTAAAATAAGGCTGA
-, 20: GGCTGAGAGCTACTGG

2. Lookup each segment and prioritize

Seeds

+, 0: CCAGTAGCTCTCAGCC
+, 10: TCAGCCTTATTTTACC
+, 20: TTTACCCAGGCCTGTA
-, 0: TACAGGCCTGGGTAAA
-, 10: GGTAAAATAAGGCTGA
-, 20: GGCTGAGAGCTACTGG

Ungapped alignment with FM Index

Seed alignments (as B ranges)

{ [211, 212], [212, 214] }
{ [653, 654], [651, 653] }
{ [684, 685] }
{ }
{ }
{ [624, 625] }

3. Evaluate end-to-end match

Extension candidates

SA:684, chr12:1955
SA:624, chr2:462
SA:211: chr4:762
SA:213: chr12:1935
SA:652: chr12:1945

SIMD dynamic programming aligner

SAM alignments

```
r1   0    chr12    1936     0
     36M  *    0    0
     CCAGTAGCTCTCAGCCTTATTTTACCCAGGCCTGTA
     IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
     AS:i:0    XS:i:-2   XN:i:0
     XM:i:0    XO:i:0    XG:i:0
     NM:i:0    MD:Z:36   YT:Z:UU
     YM:i:0
...
```

(Langmead & Salzberg, 2012)

Part 2: Variant Calling

Variant Calling Overview

Genotyping Theory

Heterozygous variant (3/7)

Homozygous variant (6/6)

```
                                                    GGTATAC…
…CCATAG     TGTGCGCCC    CGGAAATTT CGGTATAC
…CCAT    CTATGTGCG      TCGGAAATT  CGGTATAC
…CCAT GGCTATGTG     CTATCGGAAA    GCGGCATA
…CCA AGGCTATAT     CCTATCGGA    TTGCGGTA   C…
…CCA AGGCTATAT  GCCCTATCG     TTTGCGGT    C…
…CC  AGGCTATAT  GCCCTATCG  AAATTTGC    ATAC…
…CC TAGGCTATA GCGCCCTA     AAATTTGC GTATAC…
```

Subject

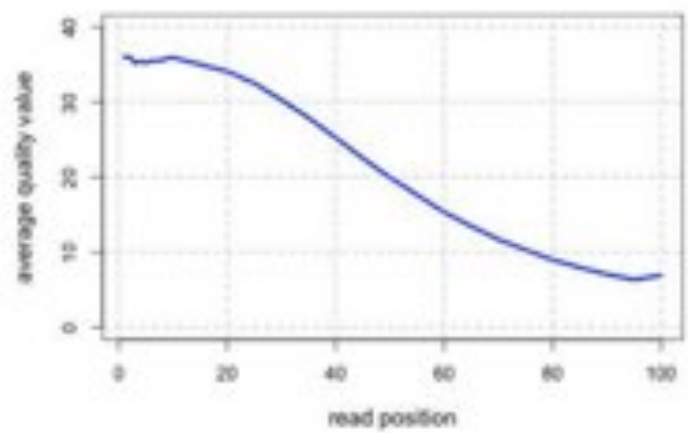
Reference

```
…CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC…
```

Error or Het (1/7)?

- If there were no sequencing errors, identifying SNPs would be very easy: any time a read disagrees with the reference, it must be a variant!

- Sequencing instruments make mistakes
  - Quality of read decreases over the read length



- A single read differing from the reference is probably just an error, but it becomes more likely to be real as we see it multiple times

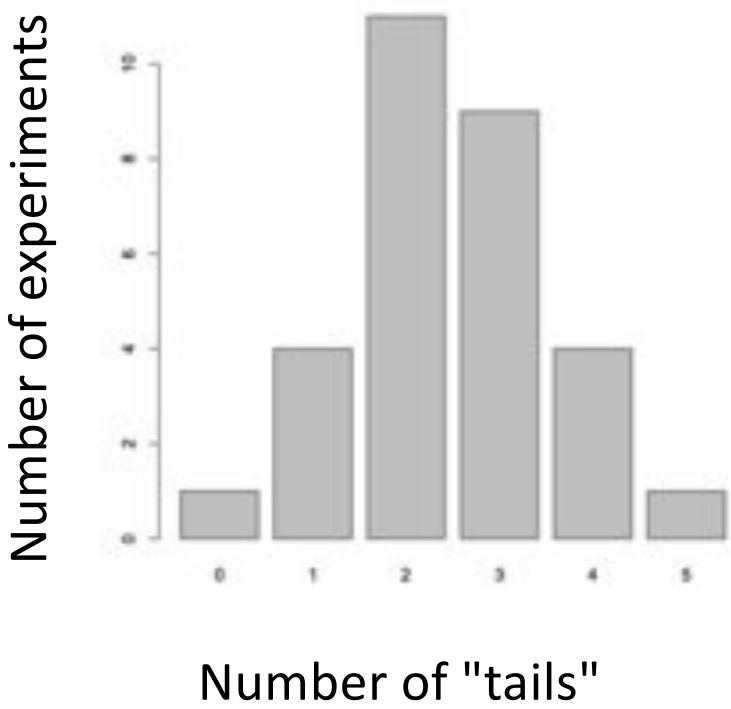
The Binomial Distribution: Adventures in Coin Flipping

P(heads) = 0.5

P(tails) = 0.5

Aaron Quinlan

What is the distribution of tails (alternate alleles) do we expect to see after 5 tosses (sequence reads)?



R code:

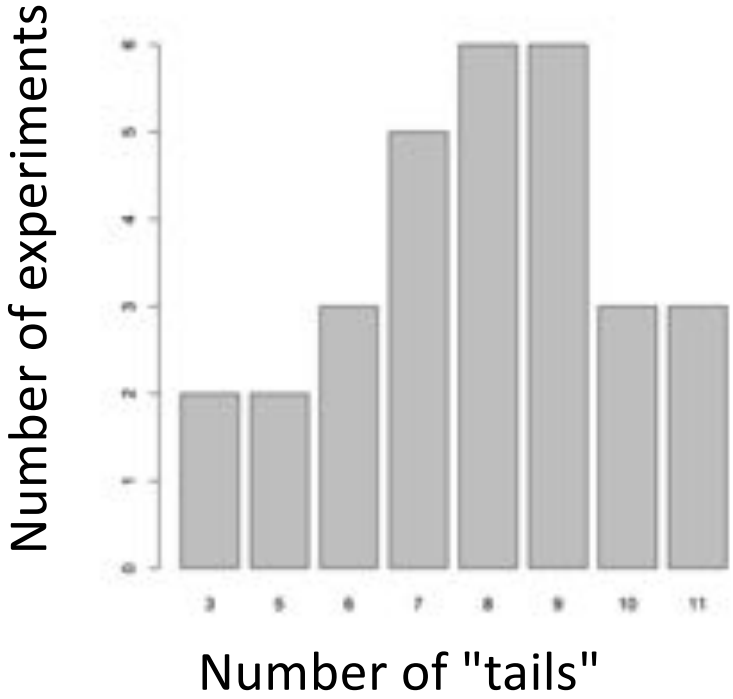
```
barplot(table(rbinom(30, 5, 0.5)))
```

30 experiments (students tossing coins)

5 tosses each

Probability of Tails

What is the distribution of tails (alternate alleles) do we expect to see after 15 tosses (sequence reads)?



R code:

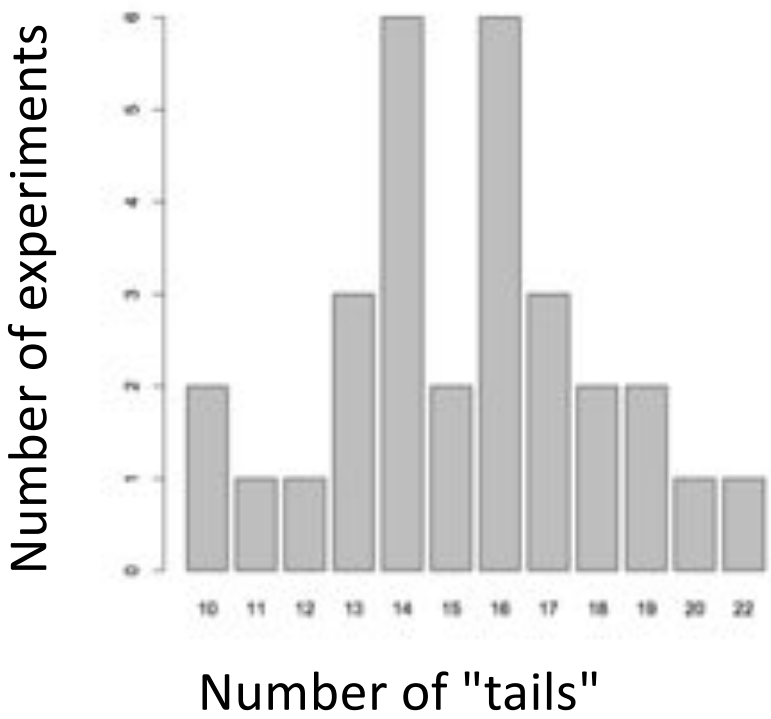
```
barplot(table(rbinom(30, 15, 0.5)))
```

30 experiments (students tossing coins)

15 tosses each

Probability of Tails

What is the distribution of tails (alternate alleles) do we expect to see after 30 tosses (sequence reads)?



R code:

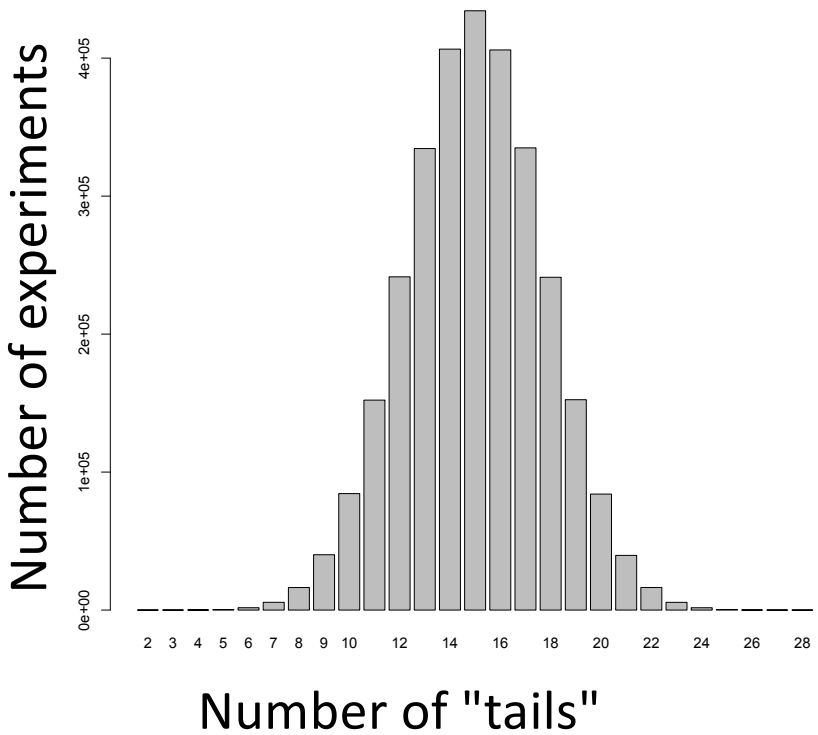
```
barplot(table(rbinom(30, 30, 0.5)))
```

30 experiments (students tossing coins)

30 tosses each

Probability of Tails

What is the distribution of tails (alternate alleles) do we expect to see after 30 tosses (sequence reads)?



R code:

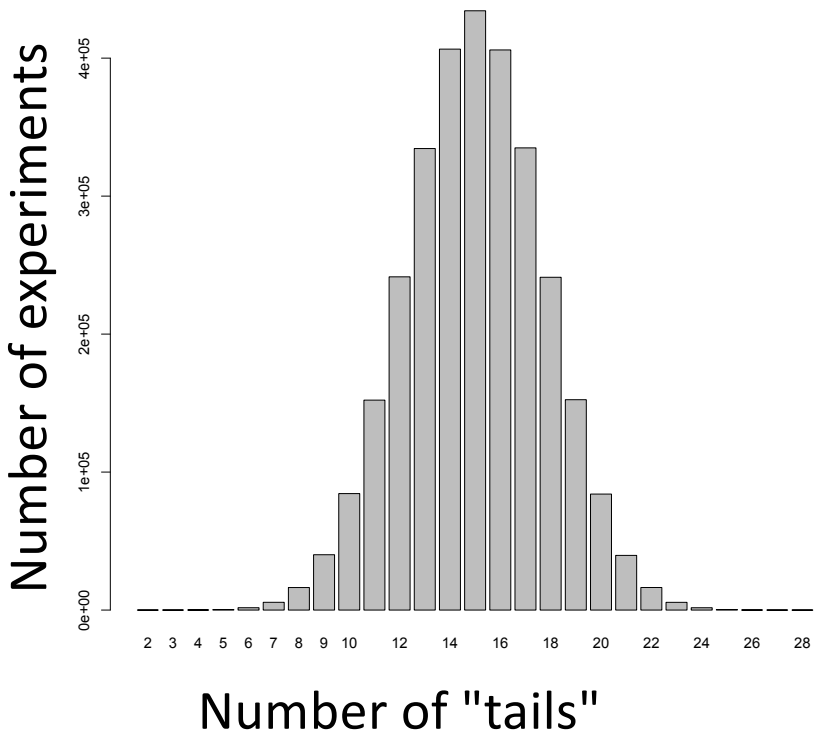
```
barplot(table(rbinom(3e6, 30, 0.5)))
```

3M experiments (students tossing coins)

30 tosses each

Probability of Tails

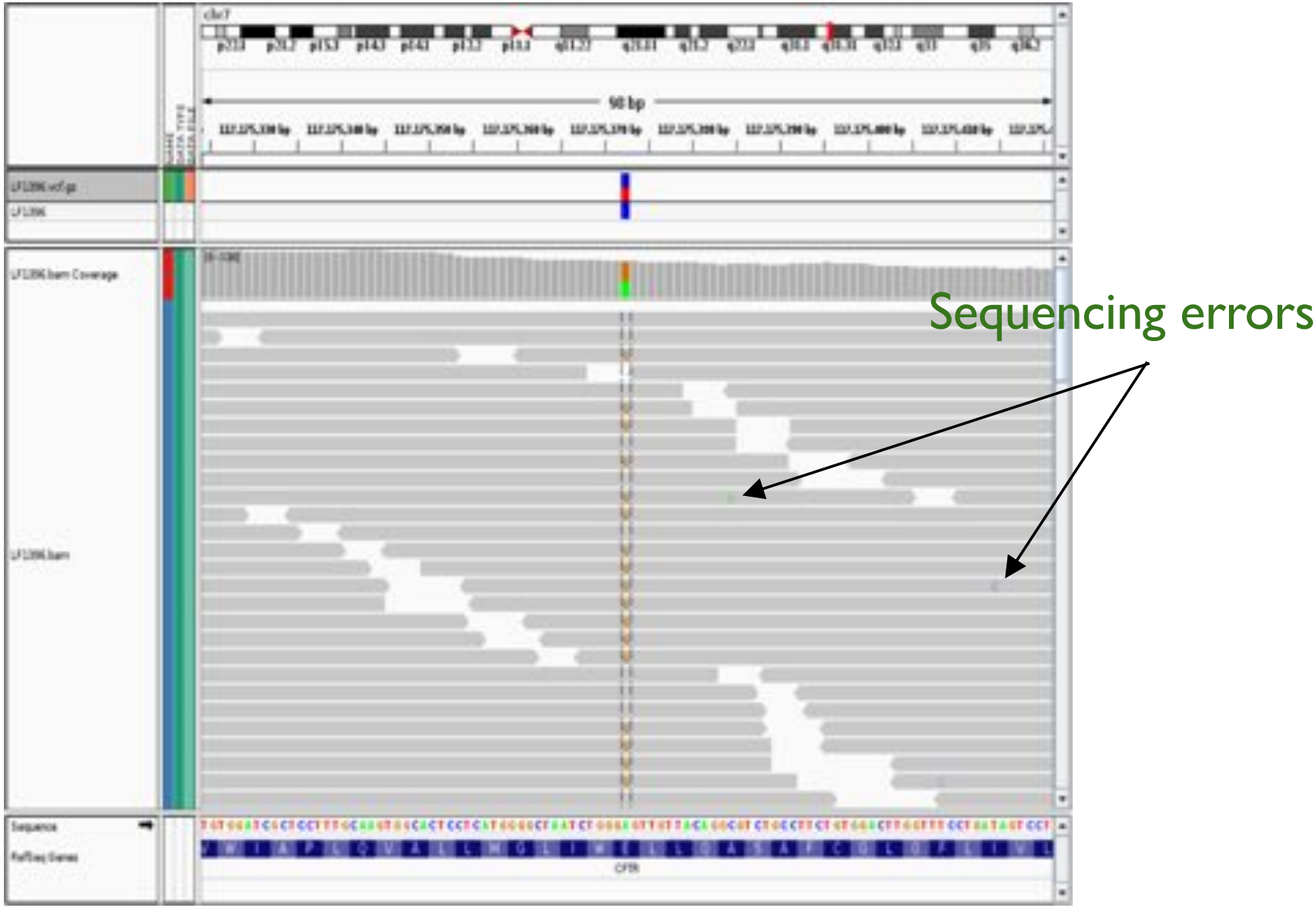
So, with 30 tosses (reads), we are much more likely to see an even mix of alternate and reference alleles at a heterozygous locus in a genome



This is why <u>at least</u> a "30X" (30 fold sequence coverage) genome is recommended: it confers sufficient power to distinguish heterozygous alleles and from mere sequencing errors

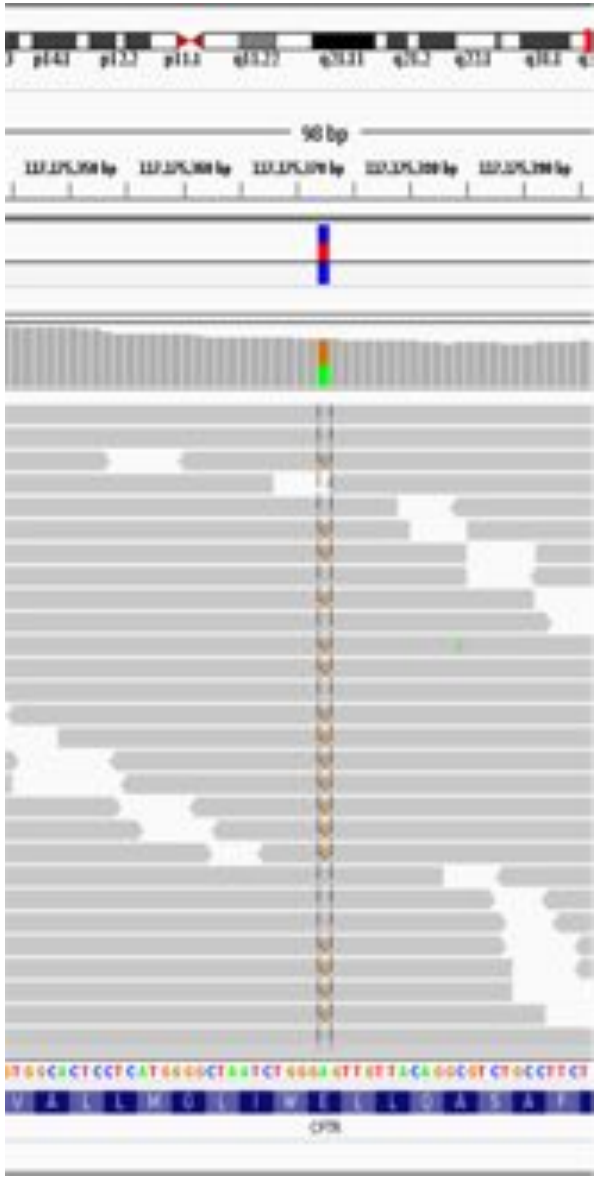
P(3/30 het) <?> P(3/30 err)

Sequencing errors fall out as noise (most of the time)



https://jchoigt.files.wordpress.com/2012/07/igv_e217g_snapshot.png

What information is needed to decide if a variant exists?



- Depth of coverage at the locus
- Bases observed at the locus
- The base qualities of each allele
- The strand composition
- Mapping qualities
- Proper pairs?
- Expected polymorphism rate

https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-10-r99

PolyBayes: The first statistically rigorous variant detection tool.

letter © 1999 Nature America Inc. - http://genetics.nature.com

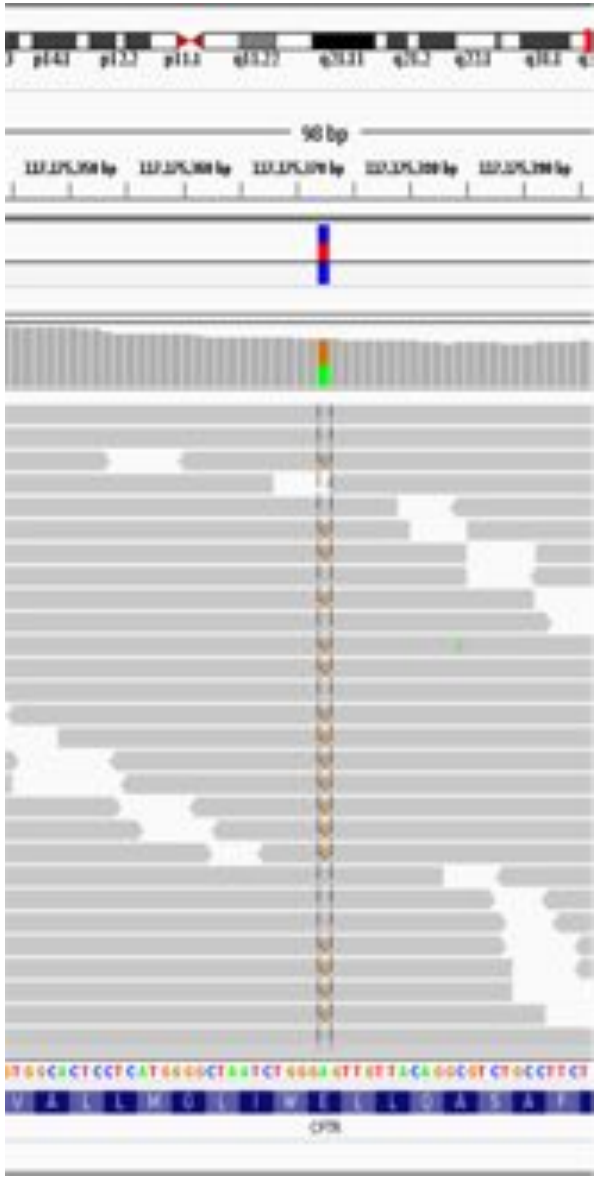
A general approach to single-nucleotide polymorphism discovery

Gabor T. Marth[1], Ian Korf[1], Mark D. Yandell[1], Raymond T. Yeh[1], Zhijie Gu[2], Hamideh Zakeri[2], Nathan O. Stitziel[1], LaDeana Hillier[1], Pui-Yan Kwok[2] & Warren R. Gish[1]

Its main innovation was the use of Bayes's theorem

http://www.nature.com/ng/journal/v23/n4/full/ng1299_452.html

Bayesian SNP calling



$$P(SNP|Data) = \frac{P(Data|SNP) * P(SNP)}{P(Data)}$$

http://www.nature.com/ng/journal/v23/n4/full/ng1299_452.html

PolyBayes: The first statistically rigorous variant detection tool.

letter © 1999 Nature America Inc. · http://genetics.nature.com

A general approach to single-nucleotide polymorphism discovery

Gabor T. Marth[1], Ian Korf[1], Mark D. Yandell[1], Raymond T. Yeh[1], Zhijie Gu[2], Hamideh Zakeri[2], Nathan O. Stitziel[1], LaDeana Hillier[1], Pui-Yan Kwok[2] & Warren R. Gish[1]

Bayesian posterior probability

Base call + Base quality

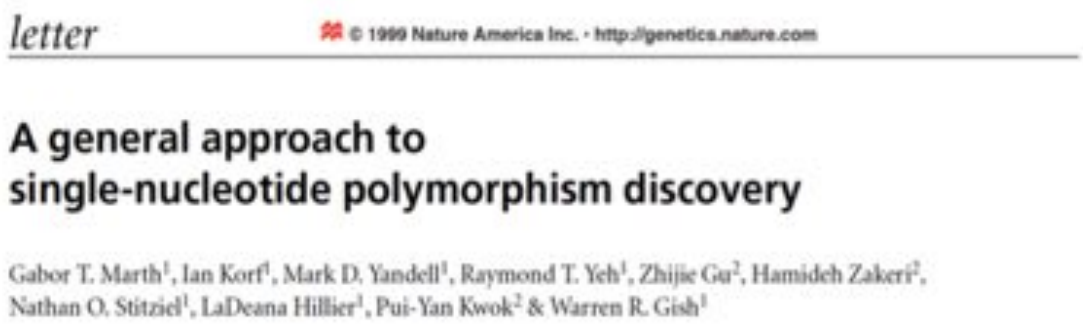
Expected (prior) polymorphism rate

$$P(SNP) = \sum_{all\ variable\ S} \frac{\frac{P(S_1 \mid R_1)}{P_{Prior}(S_1)} \cdot \ldots \cdot \frac{P(S_N \mid R_N)}{P_{Prior}(S_N)} \cdot P_{Prior}(S_1, \ldots, S_N)}{\sum_{S_{i_1} \in [A,C,G,T]} \ldots \sum_{S_{i_N} \in [A,C,G,T]} \frac{P(S_{i_1} \mid R_1)}{P_{Prior}(S_{i_1})} \cdot \ldots \cdot \frac{P(S_{i_N} \mid R_1)}{P_{Prior}(S_{i_N})} \cdot P_{Prior}(S_{i_1}, \ldots, S_{i_N})}$$

Probability of observed base composition (should model sequencing error rate)

http://www.nature.com/ng/journal/v23/n4/full/ng1299_452.html

PolyBayes: The first statistically rigorous variant detection tool.


letter © 1999 Nature America Inc. · http://genetics.nature.com

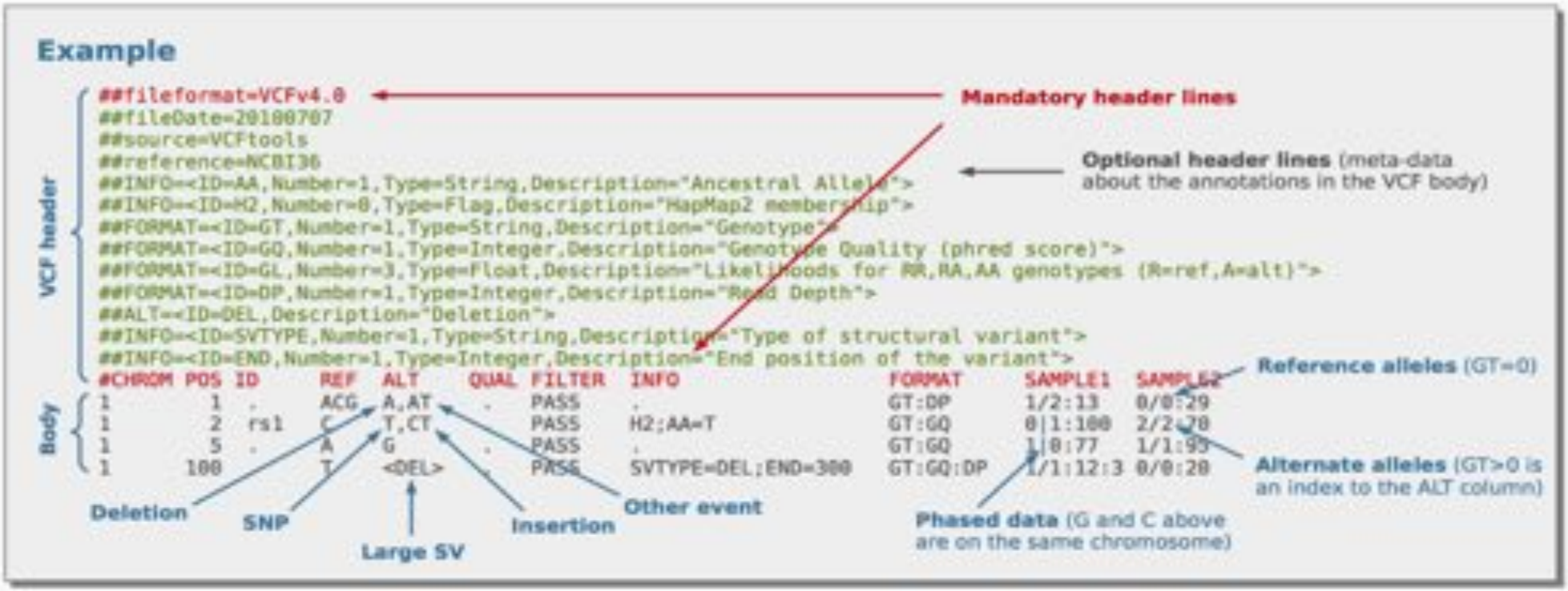
A general approach to single-nucleotide polymorphism discovery

Gabor T. Marth[1], Ian Korf[1], Mark D. Yandell[1], Raymond T. Yeh[1], Zhijie Gu[2], Hamideh Zakeri[2], Nathan O. Stitziel[1], LaDeana Hillier[1], Pui-Yan Kwok[2] & Warren R. Gish[1]

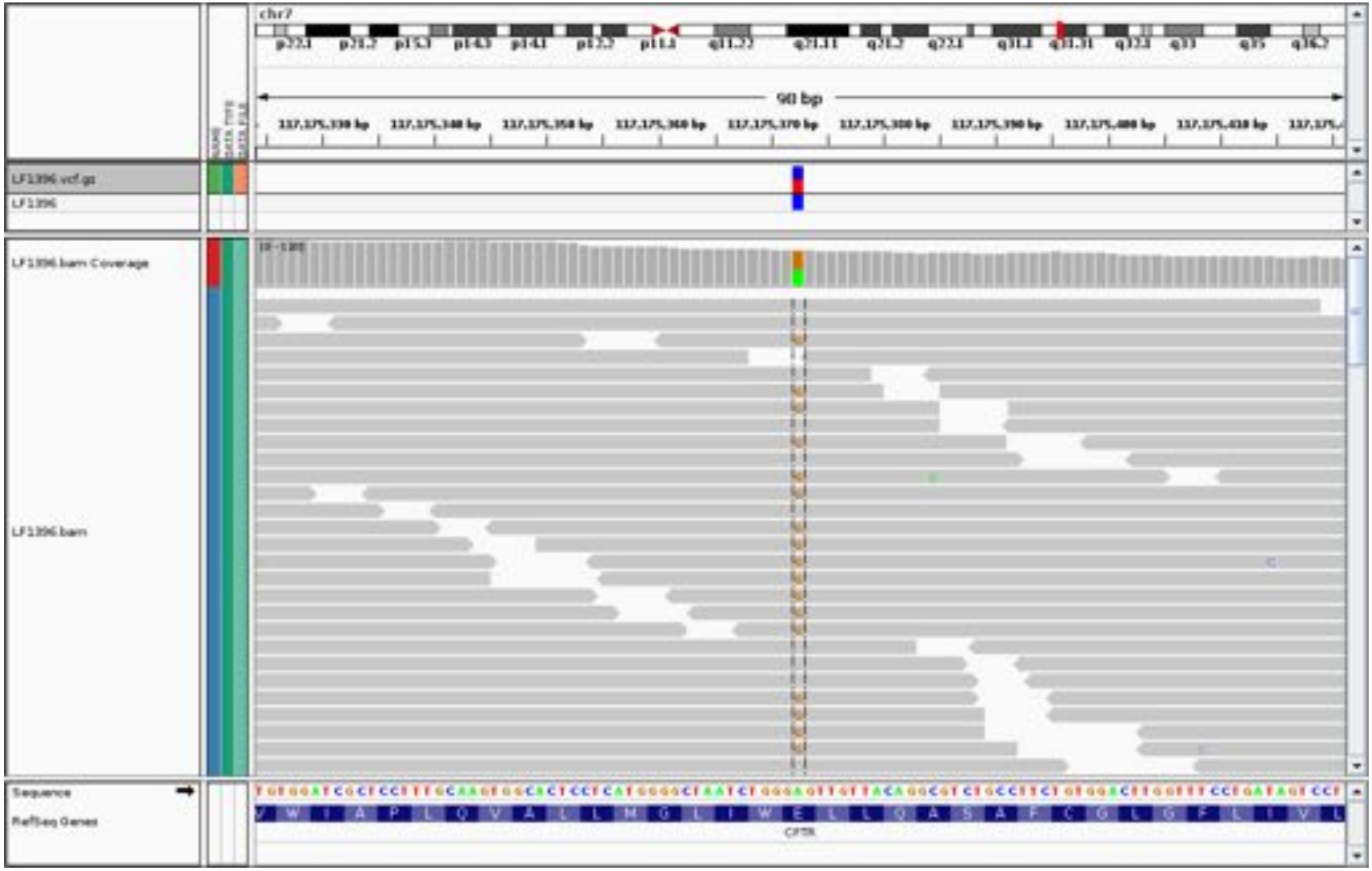
This Bayesian statistical framework has been adopted by other modern SNP/INDEL callers such as FreeBayes, GATK, and samtools

http://www.nature.com/ng/journal/v23/n4/full/ng1299_452.html

VCF Format


Example
VCF header
Body
##fileformat=VCFv4.0
##fileDate=20100707
##source=VCFtools
##reference=NCBI36
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##ALT=<ID=DEL,Description="Deletion">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT SAMPLE1 SAMPLE2
1 1 . ACG A,AT . PASS . GT:DP 1/2:13 0/0:29
1 2 rs1 C T,CT PASS H2;AA=T GT:GQ 0|1:100 2/2:78
1 5 . A G . PASS . GT:GQ 1|0:77 1/1:95
1 100 T <DEL> . PASS SVTYPE=DEL;END=300 GT:GQ:DP 1/1:12:3 0/0:20
Mandatory header lines
Optional header lines (meta-data about the annotations in the VCF body)
Reference alleles (GT=0)
Alternate alleles (GT>0 is an index to the ALT column)
Deletion
SNP
Large SV
Insertion
Other event
Phased data (G and C above are on the same chromosome)

VCF Format



#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	LF1396
chr7	117175373	.	A	G	90	PASS	AF=0.5	GT	0/1

Next Steps

1. Reflect on the magic and power of DNA ☺
2. Check out the course webpage
3. Work on HW3