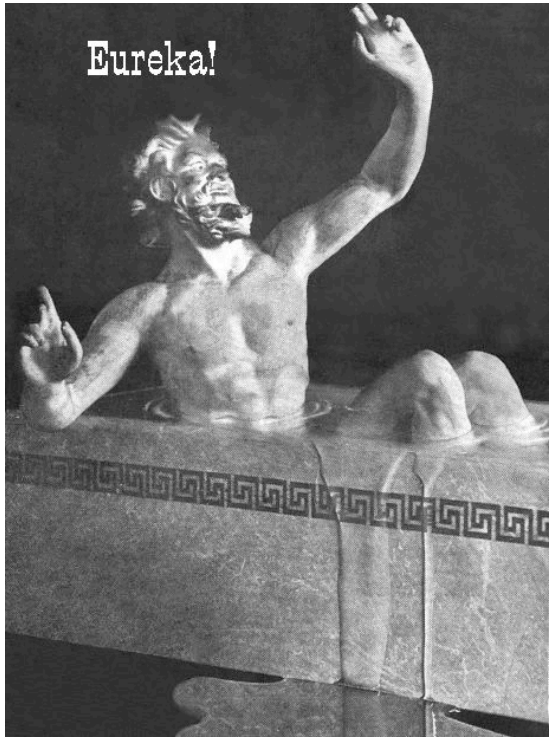
Next-Generation Sequencing (NGS) Technologies and Data Analysis

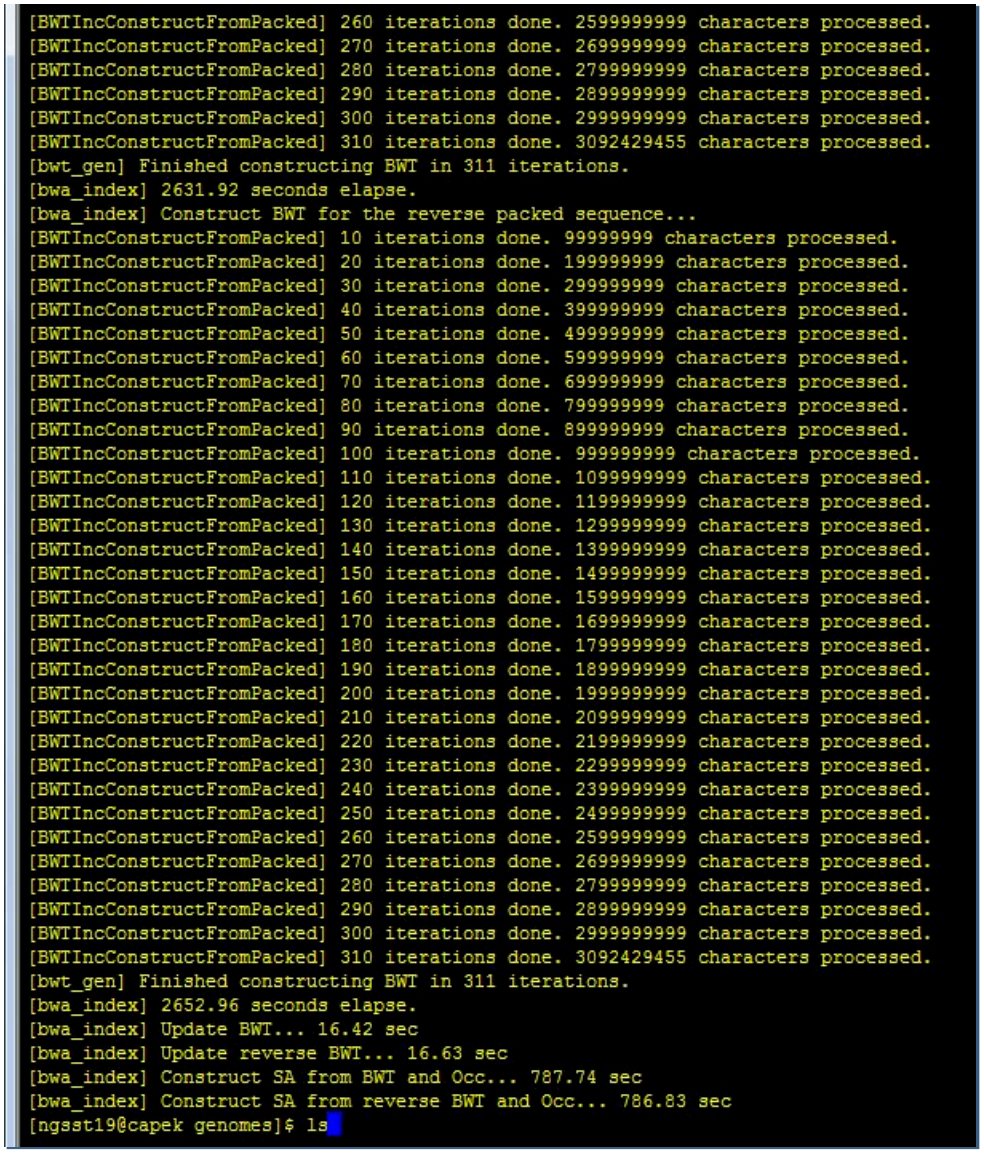
Christopher E. Mason
TA: Paul Zumbo
Spring 2010

Class #2:
Alignments, QC, and data processing

```
AAGGAGCAGGTGGCGGCGACCCTTGGCACCAGCTTG 0 0
ATTGAGTCAGGTTTCTCAGAAAATCCATCAAGATTC 18000 1 chr2:113004813 F
AAACACTAACCTTTTTCATTTTTCTTTAAAGTTTAA 16953 1 chr16:19777877 F
AAAAAGTTTTAACACTGTATGTAGAATGCACACATC 18000 1 chr8:47332993 F
ATTGCCCTCCTTTTACCCCTACCATGAGCCCTACAA 18000 1 chrM:10264 F ATT
ATAAGTCAGTGCATTGCCAAGATGTTCAAATGCCTT 18000 1 chr15:70809029 F
AAAAAAGAAAAAAACAACAACAAAAACACCACCTCT 18000 1 chr2:25982071 R
AAGGGAACCCCAAATCCTAAAGCCTCCATCTCTACT 16953 1 chr19:41197663 F
AAAAAATAAAAATCCAGAGGACATAGTATCAGTTCT 15906 1 chr2:136106265 F
AAAGCAGTGAGCCACTTGTTCTGTGTGATTTATGGT 18000 1 chr6:116705887 F
ATGAATGTTACATAAAGCATCCAGTTTGCGGTTACA 18000 1 chr1:158541939 F
AAAGGTTAACACTGATCAGGTAAACCGGGGGTGGGG 18000 1 chr19:50748240 F
GTCACTCCAGGTTTATGGAGGGTTCTTCTACTATTA 18000 2
AAACAGGCTCTGCCAAACTGACGACAGCCTTTGTGC 16953 1 chr19:62825147 F
ATTGACAAACATATCTAGTATGGCATATTAGTTCTA 16953 1 chr5:85952292 R
AAAACATTCTCCTCCGCATAAGCCTGCGTCAGATTA 16953 1 chrM:2320 F AAAA
```



A BWT Human Genome



copy the version of BWA into the 1KG directory

```
]cp BWA ../1KG/
```

Get Some Data(1KG)

http://www.1000genomes.org

Get one paired-end lane

]
ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/data/NA06985/sequence_read/ERR001014_1.filt.fastq.gz

]
ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/data/NA06985/sequence_read/ERR001014_2.filt.fastq.gz

Gunzip the files:

```
] gunzip *.gz
```

Look at the data:

```
] ls –lh
```

Count the lines (NL, WC, BYTE):

```
] wc ERR001014_2.filt.fastq
```

Do some math:

```
] expr 23494300 / 4
```

Performing Alignments

Perform the alignment (aln):

```
]./bwa aln ../genomes/hg19.fa ERR001014_1.filt.fastq > ERR001014_1.sai
]./bwa aln ../genomes/hg19.fa ERR001014_2.filt.fastq > ERR001014_2.sai
```

Other options listed at: http://bio-bwa.sourceforge.net/bwa.shtml

OPTIONS:

-n NUM	Maximum edit distance if the value is INT, or the fraction of missing alignments given 2% uniform base error rate if FLOAT. In the latter case, the maximum edit distance is automatically chosen for different read lengths. [0.04]
-o INT	Maximum number of gap opens [1]
-e INT	Maximum number of gap extensions, -1 for k-difference mode (disallowing long gaps) [-1]
-d INT	Disallow a long deletion within INT bp towards the 3'-end [16]
-i INT	Disallow an indel within INT bp towards the ends [5]
-l INT	Take the first INT subsequence as seed. If INT is larger than the query sequence, seeding will be disabled. For long reads, this option is typically ranged from 25 to 35 for '-k 2'. [inf]
-k INT	Maximum edit distance in the seed [2]
-t INT	Number of threads (multi-threading mode) [1]
-M INT	Mismatch penalty. BWA will not search for suboptimal hits with a score lower than (bestScore-misMsc). [3]
-O INT	Gap open penalty [11]
-E INT	Gap extension penalty [4]
-R INT	Proceed with suboptimal alignments if there are no more than INT equally best hits. This option only affects paired-end mapping. Increasing this threshold helps to improve the pairing accuracy at the cost of speed, especially for short reads (~32bp).
-c	Reverse query but not complement it, which is required for alignment in the color space.
-N	Disable iterative search. All hits with no more than *maxDiff* differences will be found. This mode is much slower than the default.
-q INT	Parameter for read trimming. BWA trims a read down to argmax_x{\sum_{i=x+1}^l(INT-q_i)} if q_l<INT where l is the original read length. [0]

Convert Suffix Arrays to Positions

Generate Alignments in SAM format (Single End Reads)

```
] ./bwa-0.5.7/bwa samse hg19.fa ERR001014_1.sai ERR001014_1.filt.fastq >ERR001014_1.sam
```

Generate Alignments in SAM format (Paired End Reads)

```
] ./bwa-0.5.7/bwa sampe hg19.fa ERR001014_1.sai ERR001014_2.sai ERR001014_1.filt.fastq
    ERR001014_2.filt.fastq >ERR001014_PE.sam
```

samse bwa samse [-n maxOcc] <in.db.fasta> <in.sai> <in.fq> > <out.sam>

Generate alignments in the SAM format given single-end reads. Repetitive hits will be randomly chosen.

OPTIONS:

-n INT Maximum number of alignments to output in the XA tag for reads paired properly. If a read has more than INT hits, the XA tag will not be written. [3]

sampe bwa sampe [-a maxInsSize] [-o maxOcc] [-n maxHitPaired] [-N maxHitDis] [-P] <in.db.fasta> <in1.sai> <in2.sai> <in1.fq> <in2.fq> > <out.sam>

Generate alignments in the SAM format given paired-end reads. Repetitive read pairs will be placed randomly.

OPTIONS:

-a INT Maximum insert size for a read pair to be considered being mapped properly. Since 0.4.5, this option is only used when there are not enough good alignment to infer the distribution of insert sizes. [500]

-o INT Maximum occurrences of a read for pairing. A read with more occurrneces will be treated as a single-end read. Reducing this parameter helps faster pairing. [100000]

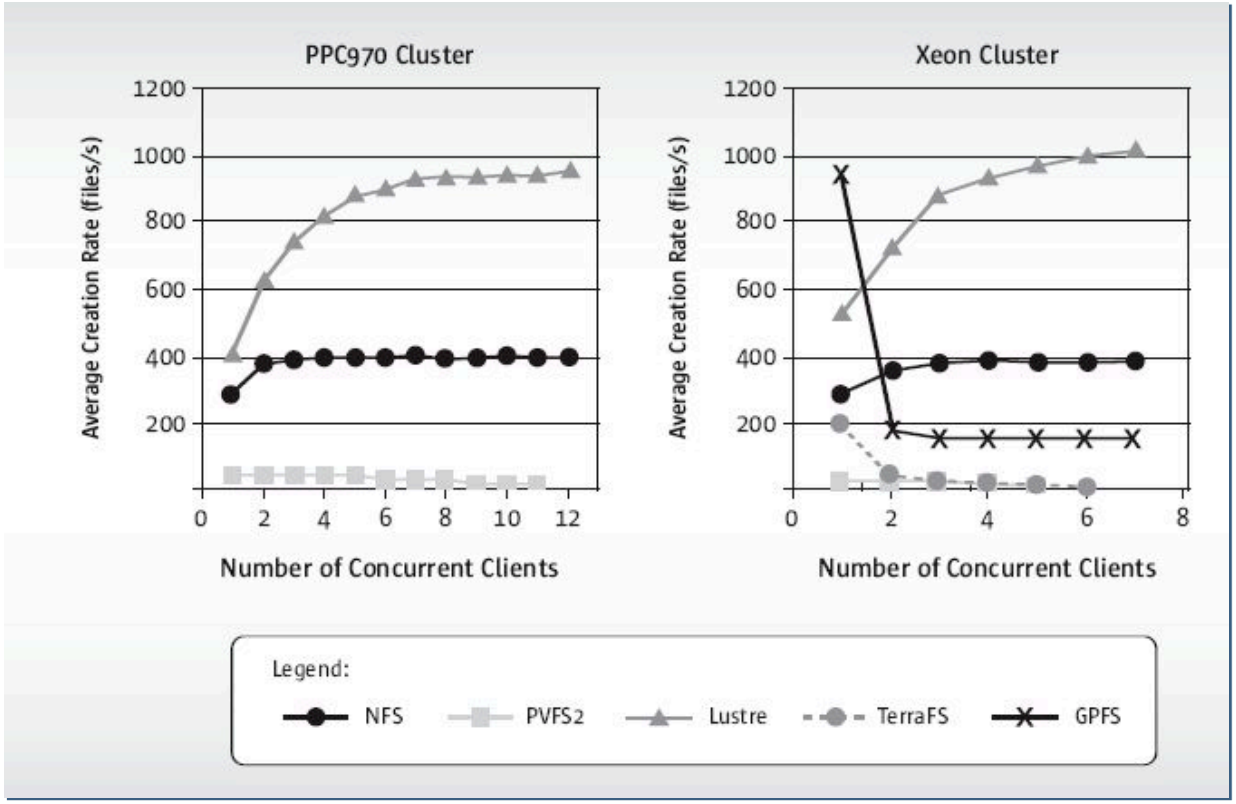
-P Load the entire FM-index into memory to reduce disk operations (base-space reads only). With this option, at least 1.25N bytes of memory are required, where N is the length of the genome.

-n INT Maximum number of alignments to output in the XA tag for reads paired properly. If a read has more than INT hits, the XA tag will not be written. [3]

-N INT Maximum number of alignments to output in the XA tag for disconcordant read pairs (excluding singletons). If a read has more than INT hits, the XA tag will not be written. [10]

Also, your I/O system can make a difference

Get a lustre filesystem if you can! (short for Linux Cluster)



Sun Microsystems

(NPS) Network File System
(PVFS) Parallel Virtual File System
(TerraFS)TerraScale Tech. File System
(GPFS)IBM General Parallel File System

Threads, Errors, and Indels strongly affect the alignments' speed and accuracy

```
] time ./bwa aln ../genomes/hg19.fa ERR001014_1.filt.fastq
   25m36.070s

] time./bwa aln –t 8 ../genomes/hg19.fa ERR001014_1.filt.fastq
   4m5.055s

] time ./bwa aln –t 8 –e 10 ../genomes/hg19.fa ERR001014_1.filt.fastq
   4m10.784s
```

Examples with a file with known variants:

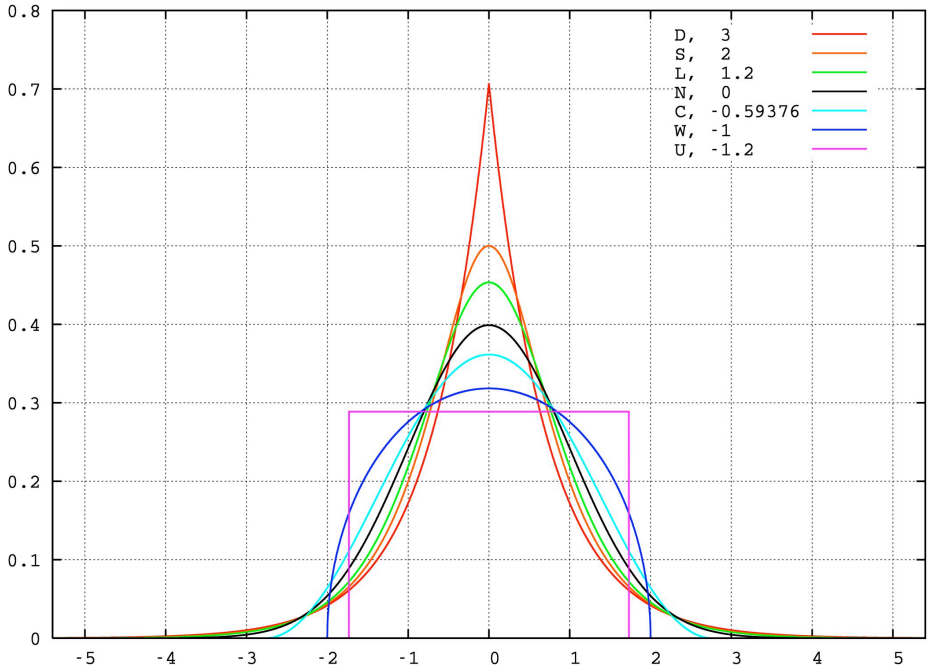
http://physiology.med.cornell.edu/faculty/mason/lab/data/NGS/

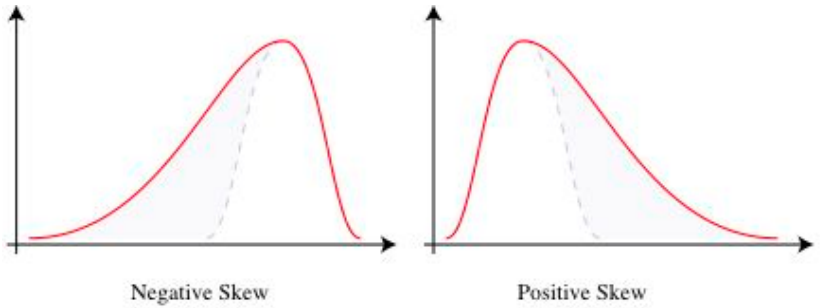
```
]./bwa aln ../genomes/hg19.fa Indels.fastq >Indels.sai
]./bwa samse ../genomes/hg19.fa Indels.sai Indels.fastq >Indels.sam
```

Now find sequences with larger indels

```
] ./bwa aln –e 10 ../genomes/hg19.fa Indels.fastq >Indels_e10.sai
] ./bwa samse ../genomes/hg19.fa Indels_e10.sai Indels.fastq >Indels_e10.sam
```

Kurtosis and Skewness Estimate PE QC





Kurtosis:
As high as possible
(at least >0.6)

Skewness:
As close to zero as possible

```
christopher-masons-macbook-pro-105:1KG christophermason$ ./bwa sampe ../genomes/hg19.fa ERR0
01014_1.sai ERR001014_2.sai ERR001014_1.filt.fastq ERR001014_2.filt.fastq >ERR001014_PE.sam
[bwa_sai2sam_pe_core] convert to sequence coordinate...
[infer_isize] (25, 50, 75) percentile: (124, 131, 138)
[infer_isize] low and high boundaries: 96 and 166 for estimating avg and std
[infer_isize] inferred external isize from 178601 pairs: 131.750 +/- 11.203
[infer_isize] skewness: 0.121; kurtosis: 0.815
[infer_isize] inferred maximum insert size: 211 (7.09 sigma)
```

What is SAM?

SAM is a rapidly developing data specification and format for the storage of sequence alignments and their mapping coordinates.

Sequence Alignment/Map (SAM) also has a binary version of the format, called BAM.

SAMtools is a set of tools for manipulating and controlling SAM/BAM files

Bam-Bam of the Flintstones is currently unrelated to Heng Li and Richard Durbin's work with SAM/BAM

SAM Output

```
@HD    VN:1.0
@SQ    SN:chr20 LN:62435964
@RG    ID:L1 PU:SC_1_10 LB:SC_1 SM:NA12891
@RG    ID:L2 PU:SC_2_12 LB:SC_2 SM:NA12891
read_28833_29006_6945 99  chr20 28833 20 10M1D25M = 28993 195 \
       AGCTTAGCTAGCTACCTATATCTTGGTCTTGGCCG <<<<<<<<<<<<<<<<<<<<:<9/,&,22;;<<< \
       NM:i:1 RG:Z:L1
read_28701_28881_323b 147 chr20 28834 30 35M      = 28701 -168 \
       ACCTATATCTTGGCCTTGGCCGATGCGGCCTTGCA <<<<<;<<<<7;:<<<6;<<<<<<<<<<<7<<<< \
       MF:i:18 RG:Z:L2
```

@HD = Header
@SQ = Sequence Dictionary
    LN=length of sequence
@RG= Read Group
    ID=unique read group identifier'
    PU=Platform Unit
    LB=Library
    SM=Sample

SAM Output

```
Test_ch21_33031597      0       chr21   33031597        37      50M     *       0       0       GCATCCATCTTGGGGCGTCCCAATTGCTGAGTAACAAATGAGACG
bcbabsbcbeaYVQZYQZXyvucdyuwf;gfhd       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:50
```

QNAME = name of read
FLAG = Bitwise FLAG (2^{16}-1)
RNAME = Reference sequence name
POS = Position (1-based)
MAPQ = Mapping Quality (Phred-based)
CIGAR = CIGAR STRING
MRNM = Mate Reference Sequence
MPOS = 1-based Mate Position of the other seq
ISIZE = Inferred Insert Size
SEQ = Sequence reported on the + strand
QUAL = Quality scores (ASCII-33 = Phred)
TAG = TAG

Tag	Meaning
NM	Edit distance
MD	Mismatching positions/bases
AS	Alignment score
X0	Number of best hits
X1	Number of suboptimal hits found by BWA
XN	Number of ambiguous bases in the referenece
XM	Number of mismatches in the alignment
XO	Number of gap opens
XG	Number of gap extentions
XT	Type: Unique/Repeat/N/Mate-sw
XA	Alternative hits; format: (chr,pos,CIGAR,NM;)*
XS	Suboptimal alignment score
XF	Support from forward/reverse alignment
XE	Number of supporting seeds

Bitwise Flags are Combined Bits

0100101001010

Bit 0 = The read was part of a pair during sequencing
Bit 1 = The read is mapped in a pair
Bit 2 = The query sequence is unmapped
Bit 3 = The mate is unmapped
Bit 4 = Strand of query (0=forward 1=reverse)

To find the value from the individual flags is additive. If the flag is false, don't add anything to the total. If it's true then add 2 raised to the power of the bit position.

For example:
Bit 0 - false - add nothing
Bit 1 - true - add 2**1 = 2
Bit 2 - false - add nothing
Bit 3 - true - add 2**3 = 8
Bit 4 - true - add 2**4 = 16

Bit pattern = 11010 = 16+8+2 = 26
So the flag value would be 26.

<u>Other Examples:</u>
0=0000000
99 = 01100011
147 = 10010011

0 = Not paired, mapped, forward strand.
99 = Paired, Proper Pair, Mapped, Mate Mapped, Forward, Mate Reverse, First in pair, Not second in pair
147 = Paired, Proper Pair, Mapped, Mate Mapped, Reverse, Mate Forward, Not first in pair, Second in pair

Bitwise Flag Explanation

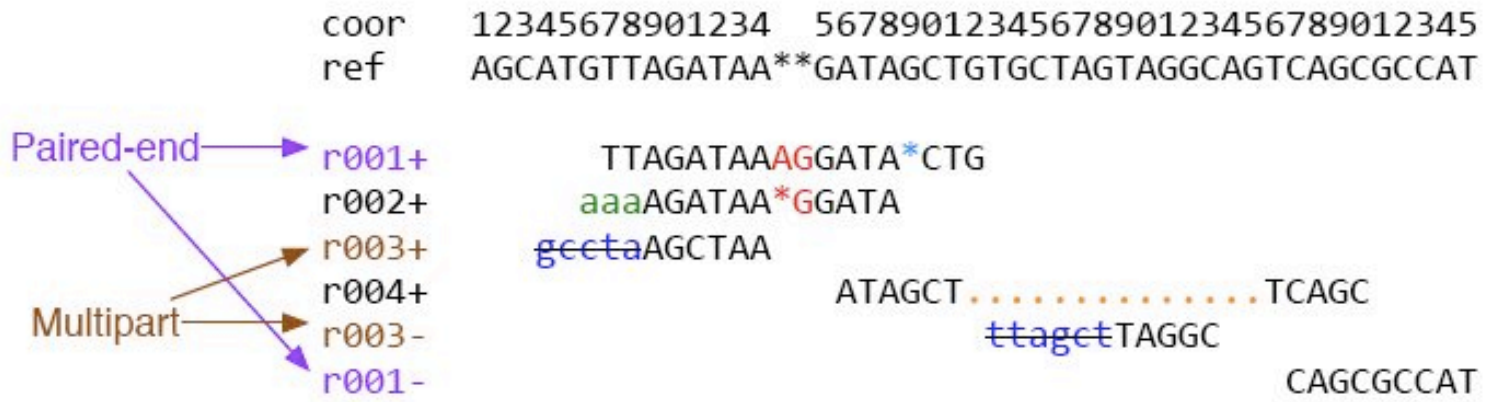
Index	$Index^2$	MethodName	Flag
0	1	isReadPartOfAPairedAlignment	0x0001
1	2	isReadAProperPairedAlignment	0x0002
2	4	isQueryUnmapped	0x004
3	8	isMateUnMapped	0x008
4	16	isQueryReverseStrand	0x0010 (false +; true -)
5	32	isMateReverseStrand	0x0020
6	64	isReadFirstPair	0x0040
7	128	isReadSecondPair	0x0080
8	256	isAlignmentNotPrimary	0x0100
9	512	doesReadFailVendorQC	0x0200
10	1024	isReadADuplicate	0x0400

SAM Specifications

SAM can store various alignments as a CIGAR format:

1. Standard
2. Clipped
   a. soft-clipped= non-matched sequence present in alignment
   b. Hard-clipped= non-matched sequence missing from alignment
4. Spliced (Intron (N))
5. Multi-part
6. Padded (Insertions (I) and Deletions (D))
7. Color-space

SAM Specifications



Ins & padding
Soft clipping
Splicing
Hard clipping

```
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTA *
r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003   0 ref  9 30 5H6M       *  0   0 AGCTAA    *   NM:i:1
r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       *  0   0 TAGGC     *   NM:i:0
r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

```
ref  7 T 1 .     | ref 12 T 3 ...       | ref 17 T 3 ...
ref  8 T 1 .     | ref 13 A 3 ...       | ref 18 A 3 .-1G..
ref  9 A 3 ...   | ref 14 A 2 .+2AG.+1G. | ref 19 G 2 *.
ref 10 G 3 ...   | ref 15 G 2 ..         | ref 20 C 2 ..
ref 11 A 3 ..C   | ref 16 A 3 ...        | ...
```

Li et al, 2010

Some ambiguities remain

CIGAR format is a short way of storing mis-aligned bases to a reference genome.

In certain cases, CIGAR will need pileup-based padding, though this is currently not supported.

```
  REF: CACGATCA**GACCGATACGTCCGA
READ1:   CGATCAGAGACCGATA
READ2:     ATCA*AGACCGATAC
READ3:    GATCA**GACCG
```

```
  REF: CACGATCA**GACCGATACGTCCGA
READ1:   CGATCAGAGACCGATA
READ2:     ATCAA*GACCGATAC
READ3:    GATCA**GACCG
```

The padded CIGAR are different:

```
READ1: 6M2I8M
READ2: 4M1P1I9M
READ3: 5M2P5M
```

```
READ1: 6M2I8M
READ2: 4M1I1P9M
READ3: 5M2P5M
```

What if I don't have fastq files, or I already have alignments?

Some tools already exist to change formats :

BWA

qualfa2fq.pl
solid2fastq.pl (not recommended!)

SAMtools

Converters

blast2sam.pl
bowtie2sam.pl
export2sam.pl
novo2sam.pl
sam2vcf.pl
soap2sam.pl
zoom2sam.pl

Tools

samtools.pl
wgsim_eval.pl

Predicting Genetic Variation with SAMtools

First, you will need to get a toolkit, and we will use SAMtools:
http://sourceforge.net/projects/samtools/files/
http://samtools.sourceforge.net/

Download the source code:
]
http://sourceforge.net/projects/samtools/files/samtools/0.1.7/samtools-0.1.7a.tar.bz2/download

Unzip the tarball (or double-click if on Desktop):
```
]bzip2 -cd samtools-0.1.7a.tar.bz2 | tar xvf -
```

Change into the new directory
```
]cd samtools-0.1.7a
```

Compile the program
```
]make
```

Move the executable into main directory
```
]cp samtools ../
```

SAMtools main options

```
samtools view -bt ref_list.txt -o aln.bam aln.sam.gz

samtools sort aln.bam aln.sorted

samtools index aln.sorted.bam

samtools view aln.sorted.bam chr2:20,100,000-20,200,000

samtools merge out.bam in1.bam in2.bam in3.bam

samtools faidx ref.fasta

samtools pileup -f ref.fasta aln.sorted.bam

samtools tview aln.sorted.bam ref.fasta
```

Pileup the Reads to call variants

Import the reads into BAM (binary SAM) format

```
]./samtools import ../genomes/hg19.fa Indels_e10.sam Indels_e10.bam
```

Sort the BAM file (for faster processing later)

```
]./samtools sort Indels_e10.bam Indels_e10.sorted
```

Perform a Pileup (layer the reads on top of each other)

```
] ./samtools pileup -vcf ../genomes/hg19.fa Indels_e10.sorted.bam
   >Indels_e10.pileup_raw
```

If you want to clean up the pileup by depth of coverage:

```
]perl samtools-0.1.7/samtools.pl varFilter -d 10 file.pileup.raw
   >Indels_e10.pileup_10X
```

If you want to clean up the pileup by quality scores of q20 (column 6). *Thanks to Alfred Aho, Peter Weinberger, and Brian Kernighan (AWK).*

```
]awk '$6>=20' file.pileup.raw >file.pileup_RMS20.out
```

To look closer at your list of variants:

```
]less ERR.pileup_10X
```

What is in the output?

1. **Chromosome:** reference sequence name
2. **Position:** reference coordinate in position (1-based)
3. **Reference Base:** base of the genome, or `*' for an indel line
4. **Genotype:** where heterozygotes are encoded in the IUPAC/IUB code: M=A/C, R=A/G, W=A/T, S=C/G, Y=C/T and K=G/T; indels are indicated by, for example, */+A, -A/* or +CC/-C. There is no difference between */+A or +A/*.
5. **Consensus Quality:** Phred-scaled likelihood that the genotype is wrong
6. **SNP Quality:** Phred-scaled likelihood that the genotype is identical to the reference, which is also called `SNP quality'. Suppose the reference base is A and in alignment we see 17 G and 3 A. We will get a low consensus quality because it is difficult to distinguish an A/G heterozygote from a G/G homozygote. We will get a high SNP quality, though, because the evidence of a SNP is very strong.
7. **RMS:** root mean square (RMS) mapping quality, a measure of the variance of quality scores
8. **Coverage:** # reads covering the position
9. **Bases with Support/Indel#1:** Bases used for SNP line, "^" from CIGAR N/S/H break, "$" end of read segment; the 1st indel allele otherwise
10. **Quality of bases/Indel#2:** base quality at a SNP line; the 2nd indel allele otherwise
11. **INDEL#1:** # reads directly supporting the 1st indel allele
12. **INDEL#2:** # reads directly supporting the 2nd indel allele
13. **INDEL#3:** # reads supporting a third indel allele
14. **Blank**

$$x_{\mathrm{rms}} = \sqrt{\frac{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}{n}}$$

SAMTools pileup uses the SNP model from MAQ, and has a few options

```
Options:
-c Create the consensus base at each position
-v Show positions that do not agree with the
  reference genome (hg19.fa)
-f The reference genome is in FASTA format
```

What are some other filters?

Parameter INTEGER [Default Value]

-Q INT minimum RMS mapping quality for SNPs [25]
-q INT minimum RMS mapping quality for gaps [10]
-d INT minimum read depth [3]
-D INT maximum read depth [100]
-G INT min indel score for nearby SNP filtering [25]
-w INT SNP within X bp around a gap to filter [10]
-W INT window size for filtering dense SNPs [10]
-N INT max number of SNPs in a window [2]
-l INT window size for filtering adjacent gaps [30]

International Union of Biochemistry (IUB) / or Intn'l Union of Pure and Applied Chemistry (IUPAC) Codes

Code	Definition	Meaning
A	Adenine	**A**
C	Cytosine	**C**
G	Guanine	**G**
T	Thymine	**T**
R	AG	pu**R**ine
Y	CT	p**Y**rimidine
K	GT	**K**eto
M	AC	a**M**ino
S	GC	**S**trong
W	AT	**W**eak
B	CGT	Not A
D	AGT	Not C
H	ACT	Not G
V	ACG	Not T
N	AGCT	a**N**y

Tview

First, make your BAM Index

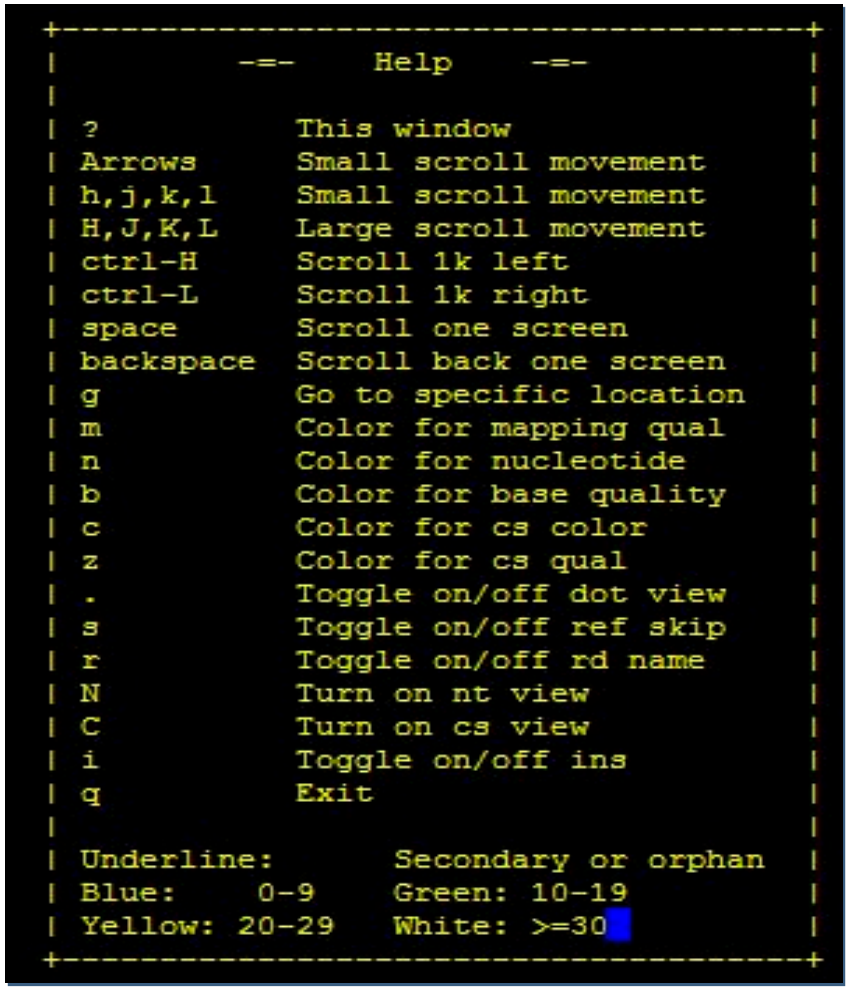
```
]./samtools index Indels_e10.sorted.bam
```

Now you can look at your alignments

```
]./samtools tview Indels_e10.sorted.bam ../genomes/hg19.fa
]./samtools tview ERR001014_PE.sorted.bam ../genomes/
hg19.fa
```

```
]Go to a specific interval
g
```

```
Then type:
chr7:57546416
```



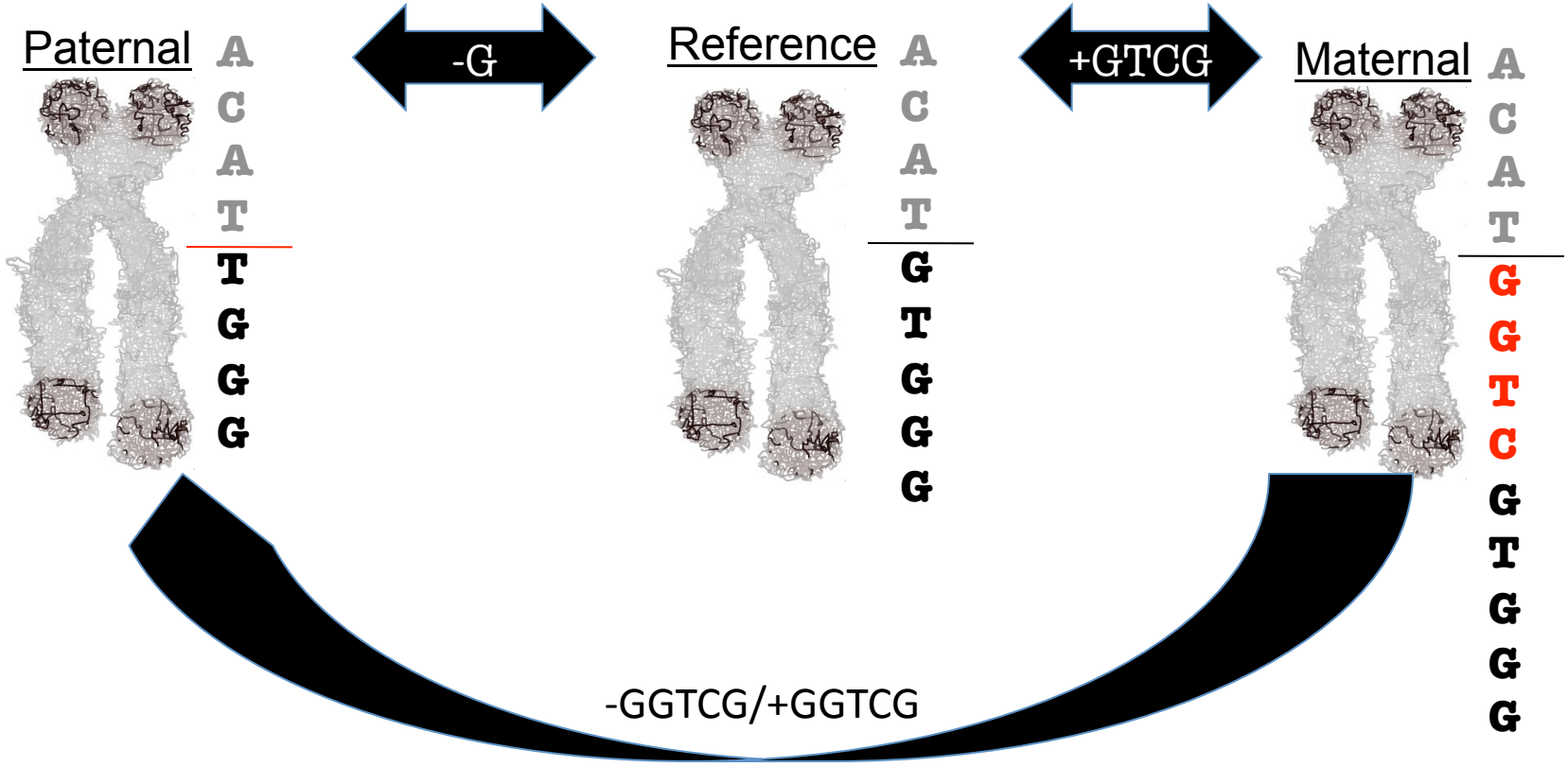
Open Factors of a Variant's Fidelity

How do we know the quality is good?

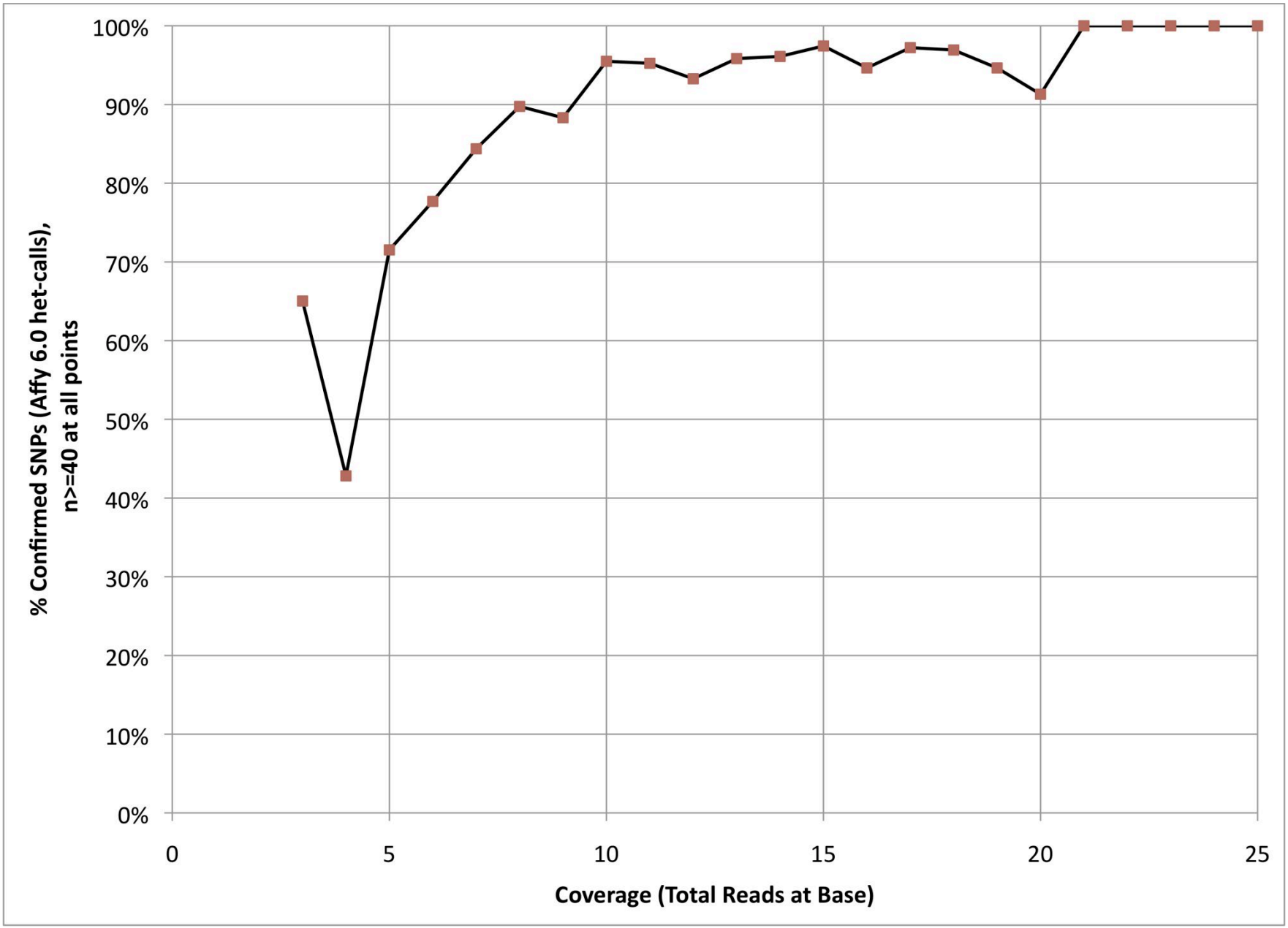
(N) Number of reads supporting that site,
(P_v) Probability of that platform-specific variant change,
(QVD) The average deviation of the quality values,
(T) The set of alignments with unique start sites,
(D) PCR Duplicates,
(S) Strand representation (half on one, half on the other),
(Z) Zygosity change (CNV regions)
(C) Cellular heterogeneity

Genomic Relativity can create interesting compound heterozygotes

chromo	location	Patient1D	reads	Patient1R	reads	Patient2D	reads	Patient2R	reads
chr11	61652197	NA	0	-G/+GTCG	12	NA	0	-G/-G	8



8-10X coverage sufficient for high-quality SNP calls



Variant Call Format

```
##format=PCFv1
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
#CHROM  POS     ID        REF   ALT    QUAL  FILTER  INFO                              FORMAT       NA00001         NA00002          NA00003
20      14370   rs6054257 G     A      29    0       NS=58;DP=258;AF=0.786;DB;H2       GT:GQ:DP:HQ  0|0:48:1:51,51  1|0:48:8:51,51   1/1:43:5
20      13330   .         T     A      3     q10     NS=55;DP=202;AF=0.024             GT:GQ:DP:HQ  0|0:49:3:58,50  0|1:3:5:65,3     0/0:41:3
20      1110696 rs6040355 A     G,T    67    0       NS=55;DP=276;AF=0.421,0.579;AA=T;DB GT:GQ:DP:HQ  1|2:21:6:23,27  2|1:2:0:18,2     2/2:35:4
20      10237   .         T     .      47    0       NS=57;DP=257;AA=T                 GT:GQ:DP:HQ  0|0:54:7:56,60  0|0:48:4:51,51   0/0:61:2
20      123456  microsat1 G     D4,IGA 50    0       NS=55;DP=250;AA=G                 GT:GQ:DP     0/1:35:4        0/2:17:2         1/1:40:3
```

Columns:

1. #CHROM
2. POS
3. ID
4. REF
5. ALT
6. QUAL
7. FILTER
8. INFO

http://1000genomes.org/wiki/doku.php?id=1000_genomes:analysis:vcfv3.2

Evaluating SNP call quality

Did I get the right number of calls?

- The number of SNP calls should be close to the average human heterozygosity of 1 variant per 1000 bases
- Only detects gross under/over calling

Concordance with hapmap chip results?

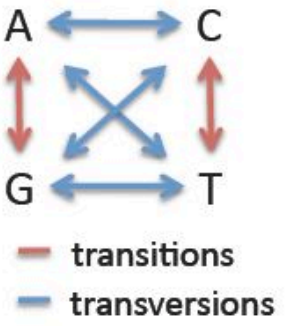
- Often we have genotype chip data that indicates the hom-ref, het, hom-var status at millions of sites
- Good SNP calls should be >99.5% consistent these chip results, and >99% of the variable sites should be found
- The chip sites are in the better parts of the genome, and so are not representative of the difficulties at novel sites

What fraction of my calls are already known?

- dbSNP catalogs most common variation, so most of the true variants found will be in dbSNP
- For single sample calls, ~90 of variants should be in dbSNP
- Need to adjust expectation when considering calls across samples

Reasonable transition to transversion ratio (Ti/Tv)?

- Transitions are twice as frequent as transversions (see *Ebersberger, 2002*)
  - Validated human SNP data suggests that the Ti/Tv should be ~2.1 genome-wide and ~2.8 in exons
- FP SNPs should has Ti/Tv around 0.5
- Ti/Tv is a good metric for assessing SNP call quality

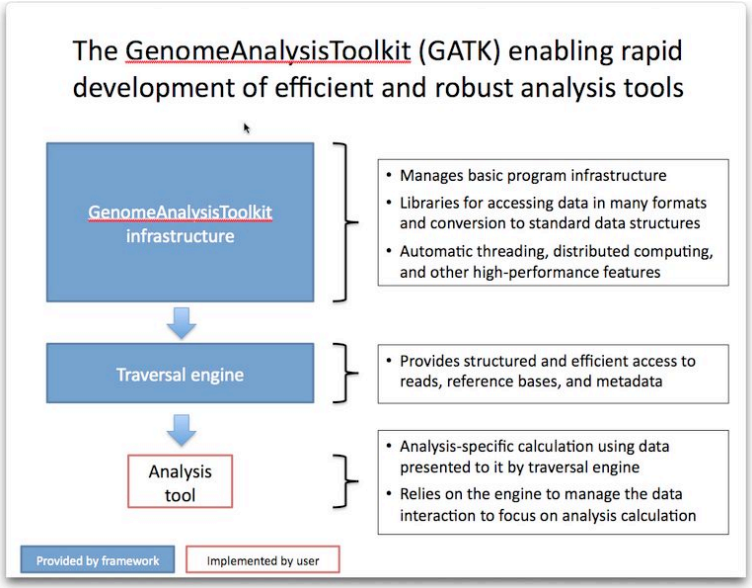


1Kg

Other Analysis Options

The Genome Analysis Toolkit (GATK)

http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit



Picard

http://picard.sourceforge.net/index.shtml

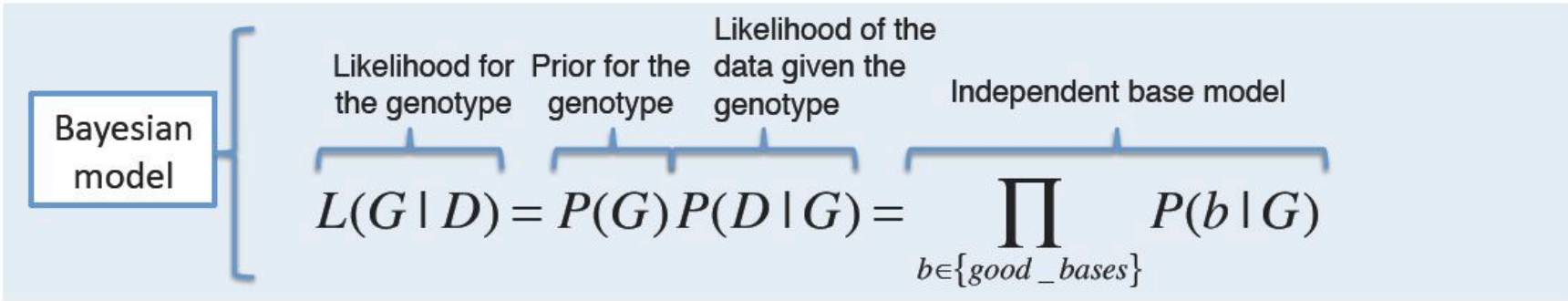
BioPerl:

Bio::DB::Sam

Analyze Features of Errors for each base with GATK

- Reported quality score
- The position within the read
- The preceding and current nucleotide (sequencing chemistry effect) observed by the sequencing machine
- Probability of mismatching the reference genome
- Re-calculate Qscores

GATK single sample genotype likelihoods



- Priors applied during multi-sample calculation; P(G) = 1
- Likelihood of data computed using pileup of bases and associated quality scores at given locus
- Only "good bases" are included: those satisfying minimum base quality, mapping read quality, pair mapping quality, NQS
- *P(b | G)* uses platform-specific confusion matrices

GATK Examples (extra)

First, let's make sure you have java

] java –version

Now, let's get GATK

] wget ftp://ftp.broadinstitute.org/pub/gsa/GenomeAnalysisTK/GenomeAnalysisTK-latest.tar.bz2

Bunzip2, and untar the file. Cd into that directory.

Let's get some example data to work with:

] wget ftp://ftp.broadinstitute.org/pub/gsa/exampleFiles/exampleFiles.tar.bz2

How many reads do you have?

] java -jar GenomeAnalysisTK.jar -R exampleFASTA.fasta -I exampleBAM.bam -T CountReads

How many loci do you have?

] java -jar GenomeAnalysisTK.jar -R exampleFASTA.fasta -I exampleBAM.bam -T CountLoci

GATK Genome Processing

```
/usr/local/cluster/software/jre1.6-amd64/jre1.6.0_01/bin/java -jar
GenomeAnalysisTK.jar -R hg18.fasta -I ../751351R.bam.sorted.bam -T
CountReads
```

GATK Options

Some Available analyses:

VariantAnnotator Annotates variant calls with context information.

DepthOfCoverage Computes the depth of coverage at all loci in the specified region of the reference.

VariantFiltration Filters variant calls using a number of user-selectable, parameterizable criteria.

UnifiedGenotyper A variant caller which unifies the approaches of several disparate callers.

IndelGenotyperV2 This is a simple, counts-and-cutoffs based tool for calling indels from aligned sequencing data.

IndelRealigner Performs local realignment of reads based on misalignments due to the presence of indels.

RealignerTargetCreator Emits intervals for the Local Indel Realigner to target for cleaning.

CountLoci Walks over the input data set, calculating the total # of covered loci for diagnostic purposes.

CountReads Walks over the input data set, calculating the # of reads seen for diagnostic purposes.

ValidatingPileup At every locus in the input set, compares the pileup data (reference base, aligned base

VariantEval A robust and general purpose tool for characterizing the quality of SNPs, Indels, and other variants that includes basic counting, ti/tv, dbSNP% (if -D is provided), concordance to chip or validation data, and will show interesting sites (-V) that are FNs, FP, etc.walkers

If sharing a cluster, be a good netizen

Use these tips to be a good portmanteau:

1. Check your disk usage (df –h)
2. Submit jobs to the queue, if available
3. Share genome indices in one place
4. Leave the camp site better than you found it. Clean up old files!
5. Never assume a backup is there; make your own
    1. LiveSync for synchonrization
    2. Time Capsule for Backup