Tutorial NGS

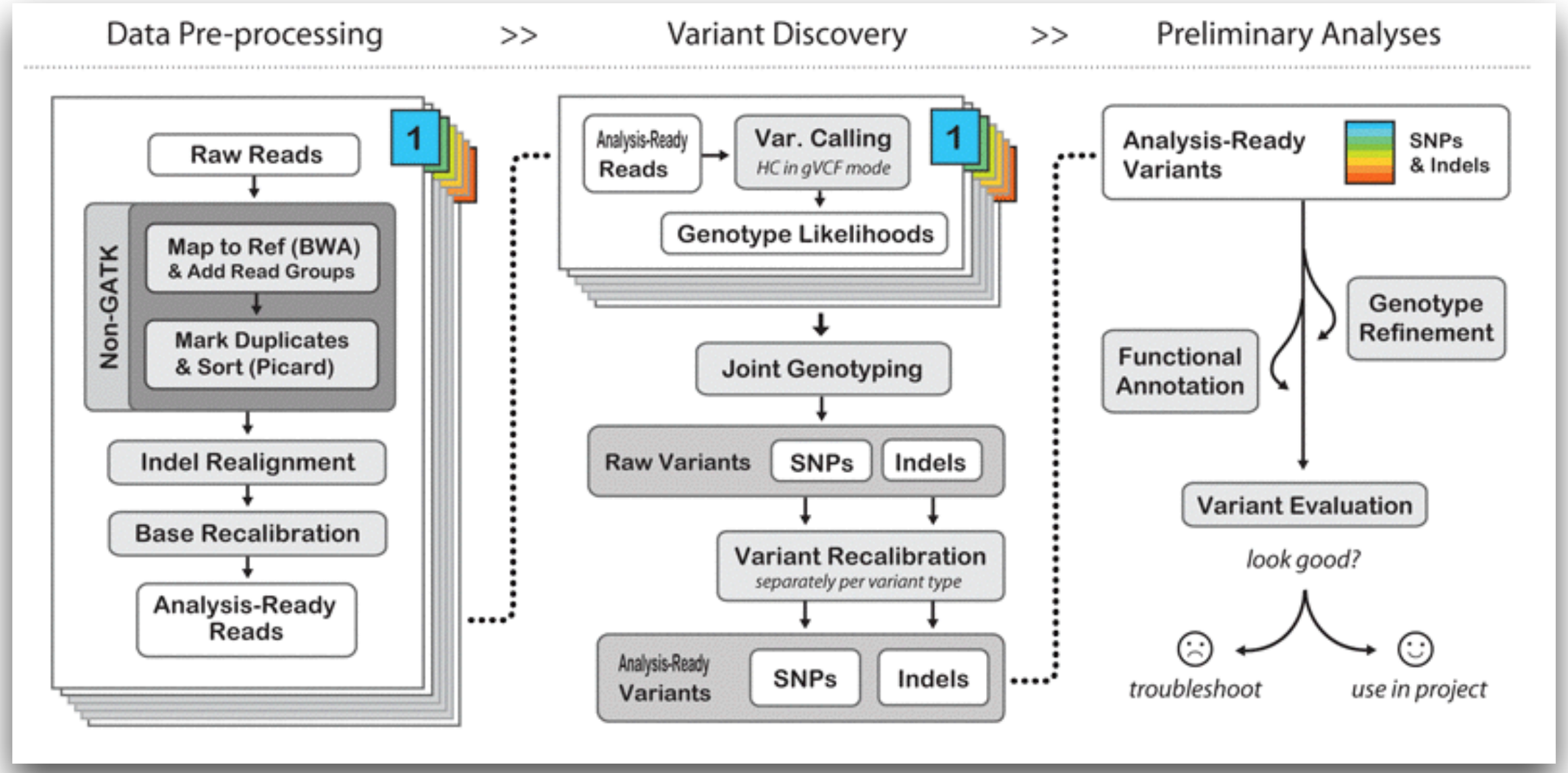
Exome sequencing

Roland Krause

Objectives

- Learn the basic workflow
- Take a glimpse at index structures and learn about the Burrows-Wheeler Transform
- From an example BAM file extract the chr22 reads and store it in FASTQ
- Learn how to quality control the reads using FastQC
- Perform an alignment
- Learn how to improve the alignments: mark duplicates, BQSR, local realignment
- Call variants

GATK pipeline



https://www.broadinstitute.org/gatk/guide/best-practices

The human reference genome

- Is stored in FASTA format build GRCh37
- For this tutorial: taken from 1000 Genomes project + Modification of chrMT to be compatible with ENSEMBL

```
$ cat human_g1k_v37_Ensembl_MT_66.fasta | grep '>’

>1 dna:chromosome chromosome:GRCh37:1:1:249250621:1
>2 dna:chromosome chromosome:GRCh37:2:1:243199373:1
….
>X dna:chromosome chromosome:GRCh37:X:1:155270560:1
>Y dna:chromosome chromosome:GRCh37:Y:2649521:59034049:1
>MT dna_rm:chromosome chromosome:GRCh37:MT:1:16569:1
>GL000207.1 dna:supercontig supercontig::GL000207.1:1:4262:1
>GL000226.1 dna:supercontig supercontig::GL000226.1:1:15008:1
>GL000229.1 dna:supercontig supercontig::GL000229.1:1:19913:1
….
```

The FASTQ format (Illumina)

```
1.  @FCC189PACXX:2:1314:19975:86201/2
2.  CTCTCTTTCTCTCTTTCTCTCTTTCTTTTCTTTTTTCTT ...
3.  +
4.  bb_eeeeegggfghhhiagfihhhhiiiihiiiiiiafg ...
```

1. Instrument name, flowcell id, coordinates within the tile, first or second read pair
2. The sequence of the read
3. Optional description
4. Quality values in ASCII (33 + Phred scaled Q)

Sequences from NGS machines are stored in that format!

The SAM alignment format

- http://samtools.github.io/hts-specs/SAMv1.pdf
- Has become the standard for storing NGS alignment data
- BAM format is the binary compressed version of SAM including indexing capabilities for fast access
- Many tools support this format
- Developed at Welcome Trust Sanger Institute by Heng Li and published in 2009 (Bioinformatics)

The SAM alignment format

- http://samtools.github.io/hts-specs/SAMv1.pdf
- Has become the standard for storing NGS alignment data
- BAM format is the binary compressed version of SAM including indexing capabilities for fast access
- Many tools support this format
- Developed at Welcome Trust Sanger Institute by Heng Li and published in 2009 (Bioinformatics)

BAM headers: an essential part of a BAM file

```
@HD    VN:1.0  GO:none SO:coordinate
@SQ    SN:chrM         LN:16571
@SQ    SN:chr1         LN:247249719
@SQ    SN:chr2         LN:242951149
[cut for clarity]
@SQ    SN:chr9         LN:140273252
@SQ    SN:chr10        LN:135374737
@SQ    SN:chr11        LN:134452384
[cut for clarity]
@SQ    SN:chr22    LN:49691432
@SQ    SN:chrX         LN:154913754
@SQ    SN:chrY         LN:57772954
@RG    ID:20FUK.1    PL:illumina    PU:20FUKAAXX100202.1    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.2    PL:illumina    PU:20FUKAAXX100202.2    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.3    PL:illumina    PU:20FUKAAXX100202.3    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.4    PL:illumina    PU:20FUKAAXX100202.4    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.5    PL:illumina    PU:20FUKAAXX100202.5    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.6    PL:illumina    PU:20FUKAAXX100202.6    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.7    PL:illumina    PU:20FUKAAXX100202.7    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.8    PL:illumina    PU:20FUKAAXX100202.8    LB:Solexa-18484 SM:NA12878    CN:BI
@PG    ID:BWA  VN:0.5.7      CL:tk
@PG    ID:GATK TableRecalibration      VN:1.0.2864
20FUKAAXX100202:1:1:12730:189900      163    chrM    1      60     101M    =      282    381
        GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTA...[more bases]
        ?BA@A>BBBBACBBAC@BBCBBCBC@BC@CAC@:BBCBBCACAACBABCBCCAB...[more quals]
        RG:Z:20FUK.1    NM:i:1  SM:i:37 AM:i:37 MD:Z:72G28      MQ:i:60 PG:Z:BWA      UQ:i:33
```

Required: Standard header

Essential: contigs of aligned reference sequence. Should be in karyotypic order.

Essential: read groups. Carries platform (PL), library (LB), and sample (SM) information. Each read is associated with a read group

Useful: Data processing tools applied to the reads

The SAM alignment format

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

The SAM alignment format

The CIGAR string

Op	BAM	Description
`M`	0	alignment match (can be a sequence match or mismatch)
`I`	1	insertion to the reference
`D`	2	deletion from the reference
`N`	3	skipped region from the reference
`S`	4	soft clipping (clipped sequences present in SEQ)
`H`	5	hard clipping (clipped sequences NOT present in SEQ)
`P`	6	padding (silent deletion from padded reference)
`=`	7	sequence match
`X`	8	sequence mismatch

The SAM alignment format

(a)

```
coor    12345678901234  5678901234567890123456789012345
ref     AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

r001+          TTAGATAAAGGATA*CTG
r002+        aaaAGATAA*GGATA
r003+     ~~gcctaAGCTAA
r004+                        ATAGCT..............TCAGC
r003-                     ~~ttagct~~TAGGC
r001-                                        CAGCGCCAT
```

(b)

```
@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         *
```

Short read alignment

- Before NGS:
  - SW, FASTA, BLAST
  - MEGABLAST, SSAH2, BLAT
- NGS produces short reads in high throughput, therefore new specialized fast aligners were needed

Short read aligner

- ELAND, RMAP, MAQ, ZOOM, SEQMAP, CLOUDBURST, SHRIMP: hashing reads and scan reference, flexible memory footprint, high overhead when scanning few reads

- SOAPV1, PASS, MOM, PROBEMATCH, NOVOALIGN, RESEQ, MOSAIK, BFAST: hash the genome, parallelizable, require large memory to build reference index, speed sensitive to sequence errors

- SOAPV2, BOWTIE, **BWA**: Burrows-Wheeler Transform (BWT), prefix tree with small memory footprint

BWA

- Uses the Burrows-Wheeler transform algorithm
- Fast and moderate memory footprint
- Gapped alignments
- Non-unique reads are placed randomly with a mapping quality=0
- Output alignments in SAM format
- Li, H. and Durbin, R., Fast and accurate short read alignment with Burrows- Wheeler transform. *Bioinformatics* **25** (14), 1754 (2009)

Searching options

- Brute force matching:
  - Trivial to implement
  - Extremely slow: O(n*l) naive or O(n+l) smart
  - Space efficient: (O(n+l)) 3 billion bytes for 3Gbp genome
- k-mer index
  - Simple to implement
  - Fast O(n) for k-mer, how to deal with multiple mapping?
  - Space inefficient (O(k+1*n) k+1 times

Suffix Array Search

```
G=GATTACA
    Suffixes Sorted Suffixes
 0  GATTACA  6      A
 1  ATTACA   4      ACA
 2  TTACA    1      ATTACA
 3  TACA     5      CA
 4  ACA      0      GATTACA
 5  CA       3      TACA
 6  A        2      TTACA
SA = 6,4,1,5,0,3,2
```

This and following material from Michael Schatz (CHSL)

schatzlab.cshl.edu/teaching/2013/2013.10.24.SBU.BWT%20Notes.pdf

Binary Search: O(l lg n); can be reduced to O(lg n) by storing LCP array

Space: N integers (offsets) + N bytes (string)

15 billion bytes for 3 Gbp genome

Burrows–Wheeler

- Want compact space O(n) bytes *and* efficient search O(lg n) or O(l)

- Goal: Optimal space index is 1 byte index per byte of text (full text index)

- BWT has these properties, plus other cool properties.

- Named for Michael Burrow and David Wheeler while working at DEC in 1994

- Original algorithm by Wheeler in 1983

Construction

Sort all cyclic rotations of G'=G$ where G is genome and $ is EOF character that is lexicographically less than all other characters in G

Example:

G=GATTACA

G’=GATTACA

BWT=ACTGA$TA

Rotations: Sorted (also called BWM)

```
GATTACA$ $GATTACA
ATTACA$G A$GATTAC
TTACA$GA ACA$GATT
TACA$GAT ATTACA$G
ACA$GATT CA$GATTA
CA$GATTA GATTACA$
A$GATTAC TACA$GAT
$GATTACA TTACA$GA
```

BWT (last column of BWM) -^

Last-first property

The magic of the BWT is the LF property: The *i*th occurrence of character C in the last column *is* the *i*th occurrence of character C in the first column.

Lets consider a schematic diagram of the BWM of a DNA string

```
$ _ _ _ _ _ _ _ _ <- By construction, first row starts with $
A _ _ _ _ _ _ _ _
A _ _ _ _ _ _ _ _ <- Followed by section for A
A _ _ _ _ _ _ _ _
...
C _ _ _ _ _ _ _ _
C _ _ _ _ _ _ _ _ <- Followed by C C _ _ _ _ _ _ _
...
G _ _ _ _ _ _ _ _
G _ _ _ _ _ _ _ _ <- Followed by G
G _ _ _ _ _ _ _ _
...
T _ _ _ _ _ _ _ _
T _ _ _ _ _ _ _ _ <- Followed by T
T _ _ _ _ _ _ _ _
```

Lets call those three rotations that start with C rotations X, Y, and Z

The first character of each of those rotations is x, y, z (without loss

of generality –– we don't know what those strings are, but we can label the characters)

. . .

C x X X X X X X

C y Y Y Y Y Y Y

C z Z Z Z Z Z Z

. . .

Rotation

Now since the BWM contains every cyclic rotation, we know those 3 C strings will also be rotated like so, someplace else in the BWM

```
CxXXXXXX        xXXXXXXC
CyYYYYYY  =>  yYYYYYYC
CzZZZZZZ        zZZZZZZC
```

Key insight: Since the rotations are sorted, we know that X < Y < Z and x <= y <= z. As such their relative placement must also be in sorted order in the BWM when C is rotated to the last column.

```
$ _ _ _ _ _ _ _
A _ _ _ _ _ _ _
A X X X X X X C <- Possible location of X (x=A)
A _ _ _ _ _ _ _
...
C x X X X X X X
C y Y Y Y Y Y Y <- Original locations of X, Y, Z
C z Z Z Z Z Z Z
...
G Y Y Y Y Y Y Y <- Possible location of Y (must be below X, y=G)
G _ _ _ _ _ _ _
G _ _ _ _ _ _ _
...
T _ _ _ _ _ _ _
T _ _ _ _ _ _ _
T Z Z Z Z Z Z C <- Possible location of Z (must be below Y, z=T)
```

Last–First property is actually a statement of the *rest* of the rotation.

When they are sorted as the second character of the rotation, they are also

sorted when they are the first character of the rotation so the ranks must

be the same.

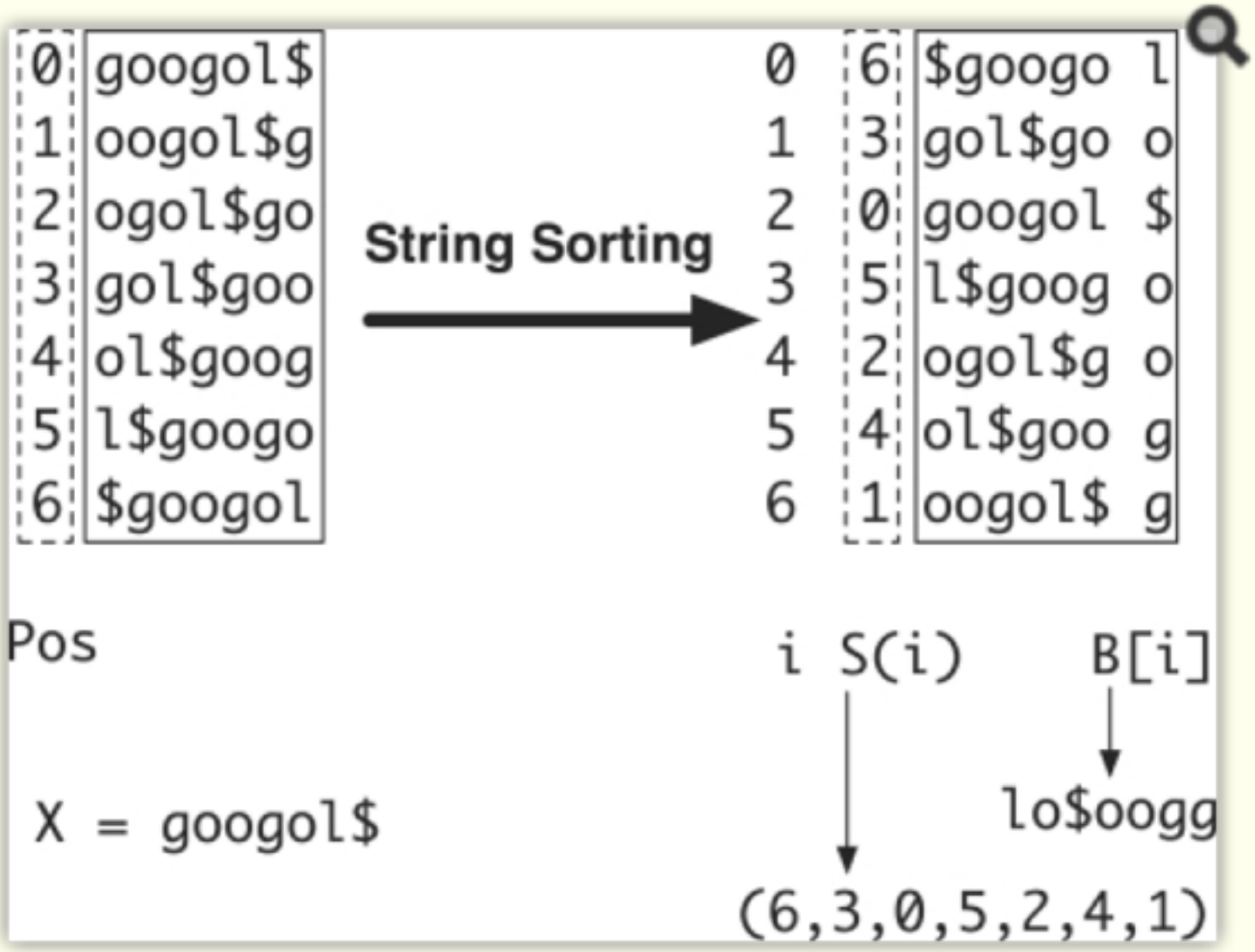
Reconstruction

```
BWT = ACTTGA$TTAA

1 2 3 4 5 6 7 8 9 0 1
$ _ _ _ _ _ _ _ _ _ A <- By construction, $ is first
A _ _ _ _ _ _ _ _ _ C <- Must have 4 A rows
A                     T "
A _ _ _ _ _ _ _ _ _ T "
A _ _ _ _ _ _ _ _ _ G "
C _ _ _ _ _ _ _ _ _ A <- 1 C row
G _ _ _ _ _ _ _ _ _ $ <- 1 G row
T _ _ _ _ _ _ _ _ _ T <- 4 T rows
T _ _ _ _ _ _ _ _ _ T "
T _ _ _ _ _ _ _ _ _ A "
T _ _ _ _ _ _ _ _ _ A "

^- Last column defined by the BWT
```

Fig. 2.



Constructing suffix array and BWT string for X=`googol$`. String X is circulated to generate seven strings, which are then lexicographically sorted. After sorting, the positions of the first symbols form the suffix array (6, 3, 0, 5, 2, 4, 1) and the concatenation of the last symbols of the circulated strings gives the BWT string `lo$oogg`.

Simulated data

- Accuracy
  - ❖BWA is more accurate than Bowtie and SOAPv2 based on criterion 1.
- Speed
  - ❖ BWA is the fastest second only to SOAPv2.
- Memory
  - ❖ MAQ's memory footprint is **1GB**, but it increases linearly with the number of reads to be aligned.
  - ❖ BWA only uses **2.3** GB for *single-end mapping* and **3GB** for *paired-end* (as much as Bowtie).
  - ❖ SOAPv2 uses **5.4 GB**.

Table 1. Evaluation on simulated data

	Single-end			Paired-end		
Program	Time (s)	Conf (%)	Err (%)	Time (s)	Conf (%)	Err (%)
Bowtie-32	1271	79.0	0.76	1391	85.7	0.57
BWA-32	823	80.6	0.30	1224	89.6	0.32
MAQ-32	19797	81.0	0.14	21589	87.2	0.07
SOAP2-32	256	78.6	1.16	1909	86.8	0.78
Bowtie-70	1726	86.3	0.20	1580	90.7	0.43
BWA-70	1599	90.7	0.12	1619	96.2	0.11
MAQ-70	17928	91.0	0.13	19046	94.6	0.05
SOAP2-70	317	90.3	0.39	708	94.5	0.34
bowtie-125	1966	88.0	0.07	1701	91.0	0.37
BWA-125	3021	93.0	0.05	3059	97.6	0.04
MAQ-125	17506	92.7	0.08	19388	96.3	0.02
SOAP2-125	555	91.5	0.17	1187	90.8	0.14

One million pairs of 32, 70 and 125 bp reads, respectively, were simulated from the human genome with 0.09% SNP mutation rate, 0.01% indel mutation rate and 2% uniform sequencing base error rate. The insert size of 32 bp reads is drawn from a normal distribution $N(170,25)$, and of 70 and 125 bp reads from $N(500,50)$. CPU time in seconds on a single core of a 2.5 GHz Xeon E5420 processor (Time), percent confidently mapped reads (Conf) and percent erroneous alignments out of confident mappings (Err) are shown in the table.

http://www.cs.utsa.edu/~jruan/teaching/CS6293_fall_2010/hw2slides/group5_Mario_Nguyen_xuepo.pptx

Removal of duplicated reads

- Introduced during library creation/ amplification
- Optical duplicates
- Many duplicates of a read with wrong indel can mask the correct one
- Will result in high read depth and can be the cause of false positives
- Duplicates: Identical 5’ coordinates and orientations
- Best: Read pair having highest sum of base qualities
- Can be removed with samtools or picard

Base Quality Score Recalibration (BQSR)

- Observed error rates differ from raw base quality scores
- More over, the base quality is not evenly distributed in a read: machine cycle bias, sequence context, sequencing chemistry effects
- BQSR is:
  - the sum of the global difference between reported quality scores and the empirical quality
  - plus the quality bin specific shift
  - plus the cycle x qual and dinucleotide x qual effect
- Sites of known variations are taken into account

Empirical versus reported BQS

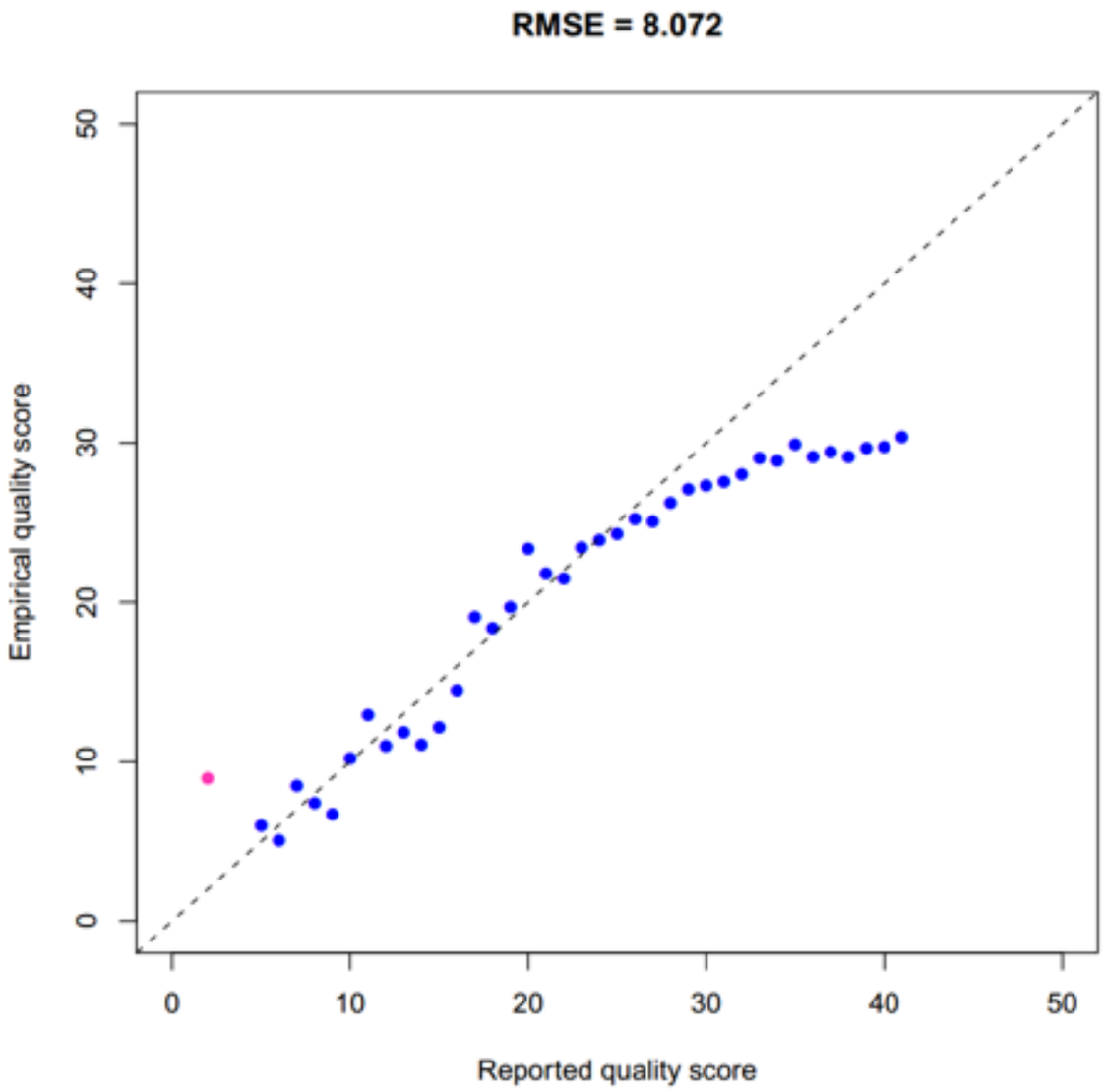


Figure 2: (a) Before BQSR

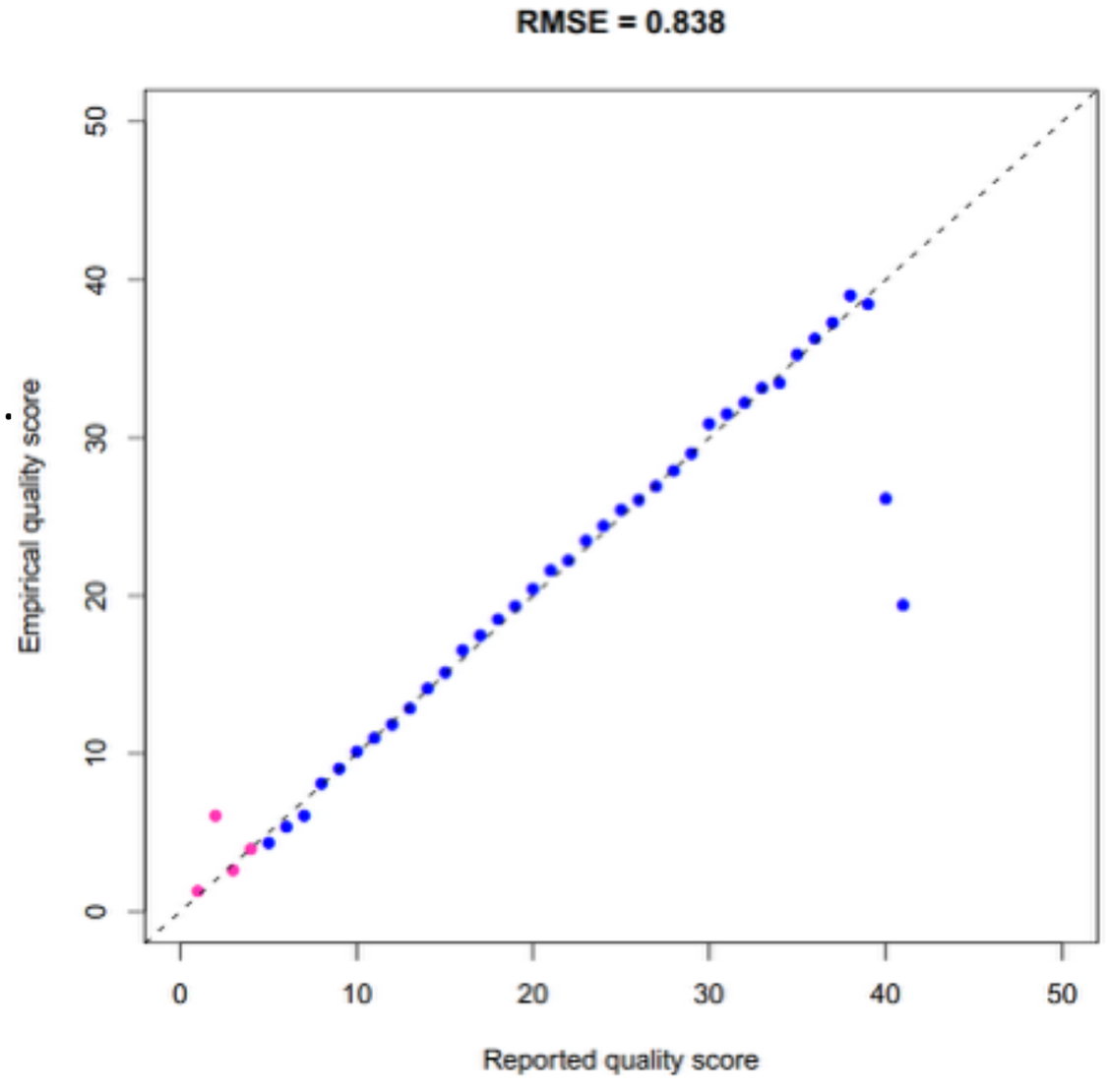


Figure 2: (b) After BQSR

Distribution of quality scores

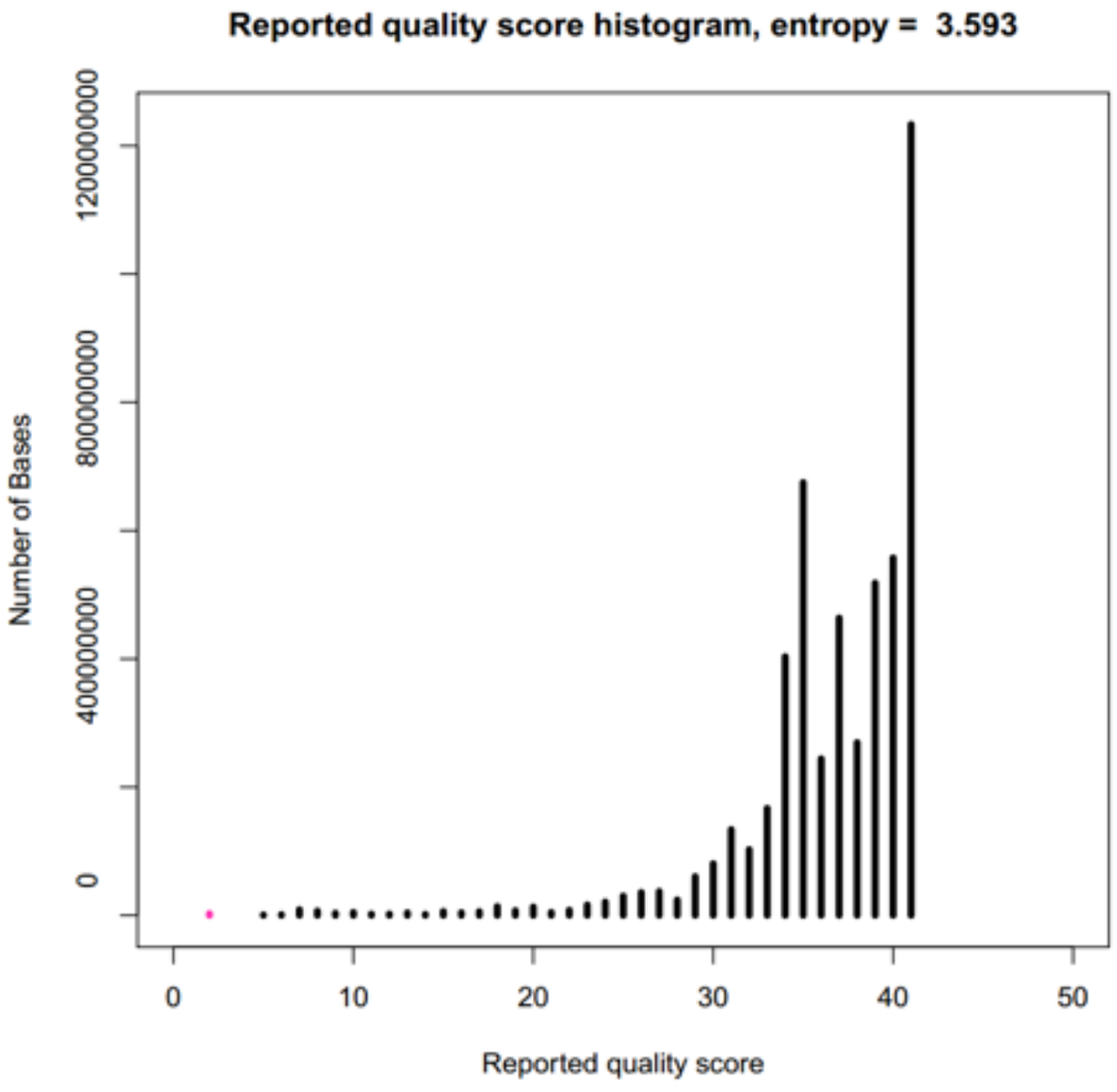


Figure 3: (a) Before BQSR

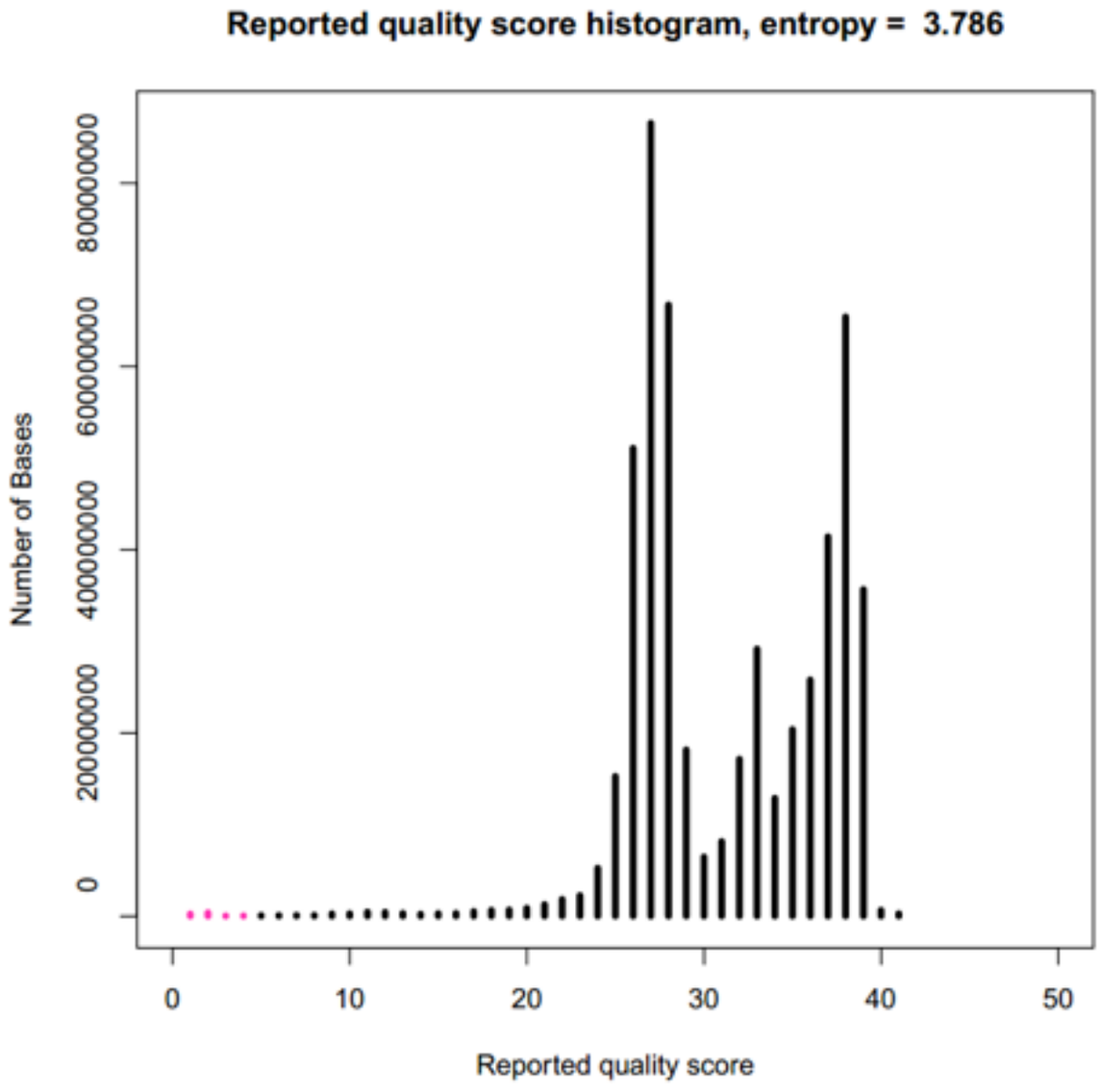


Figure 3: (b) After BQSR

Residual error by machine cycle

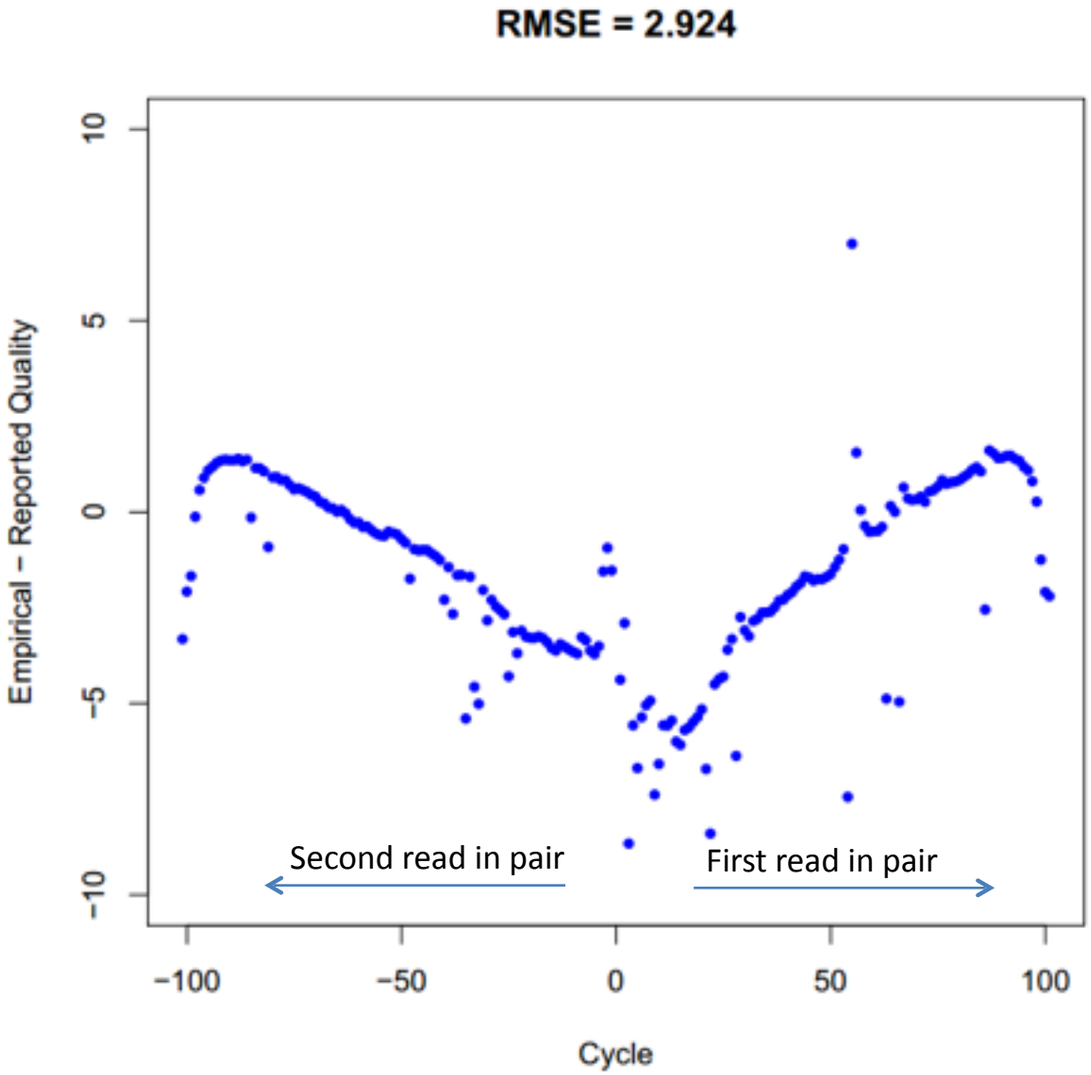


Figure 4: (a) Before BQSR

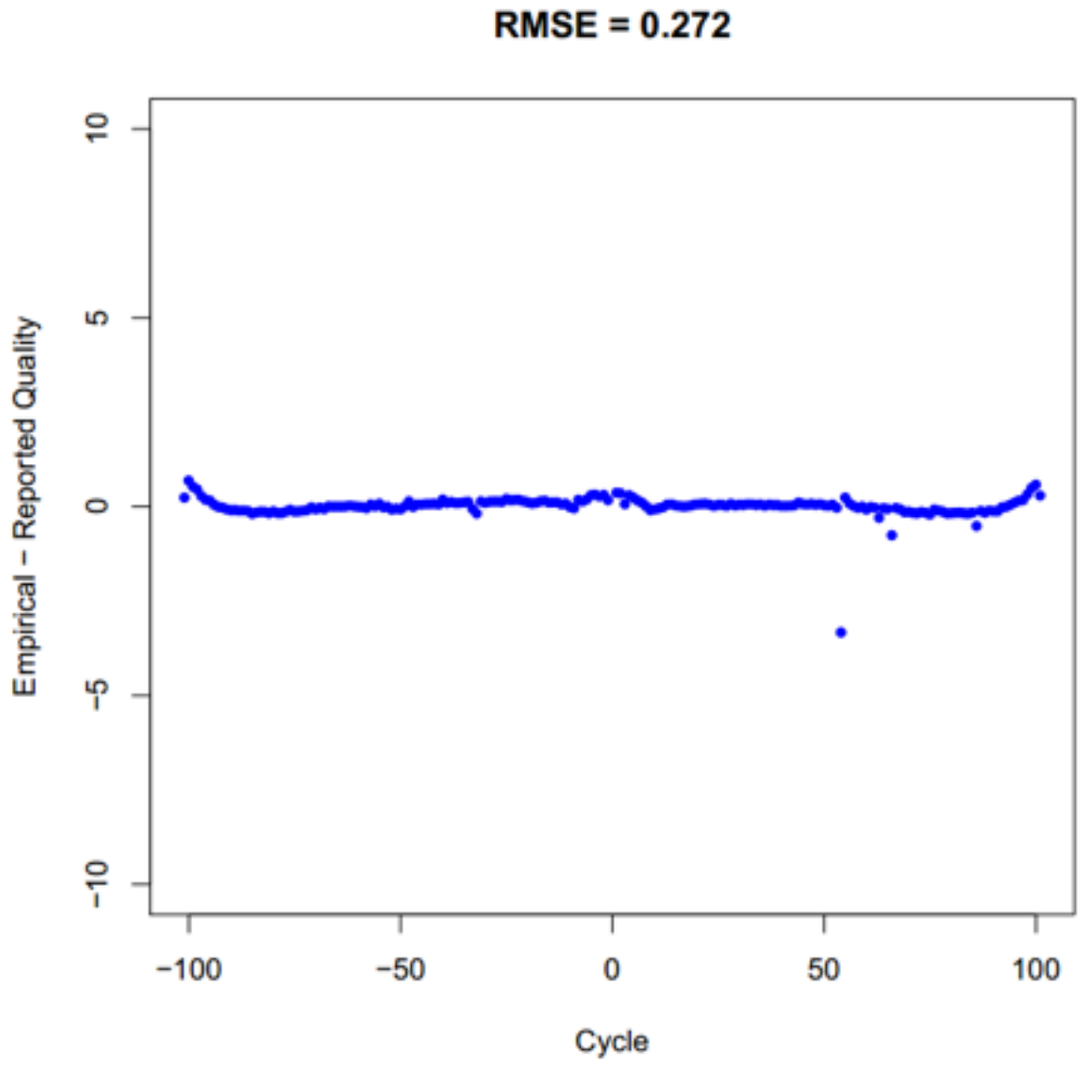


Figure 4 (b) After BQSR

Residual error by dinucleotide

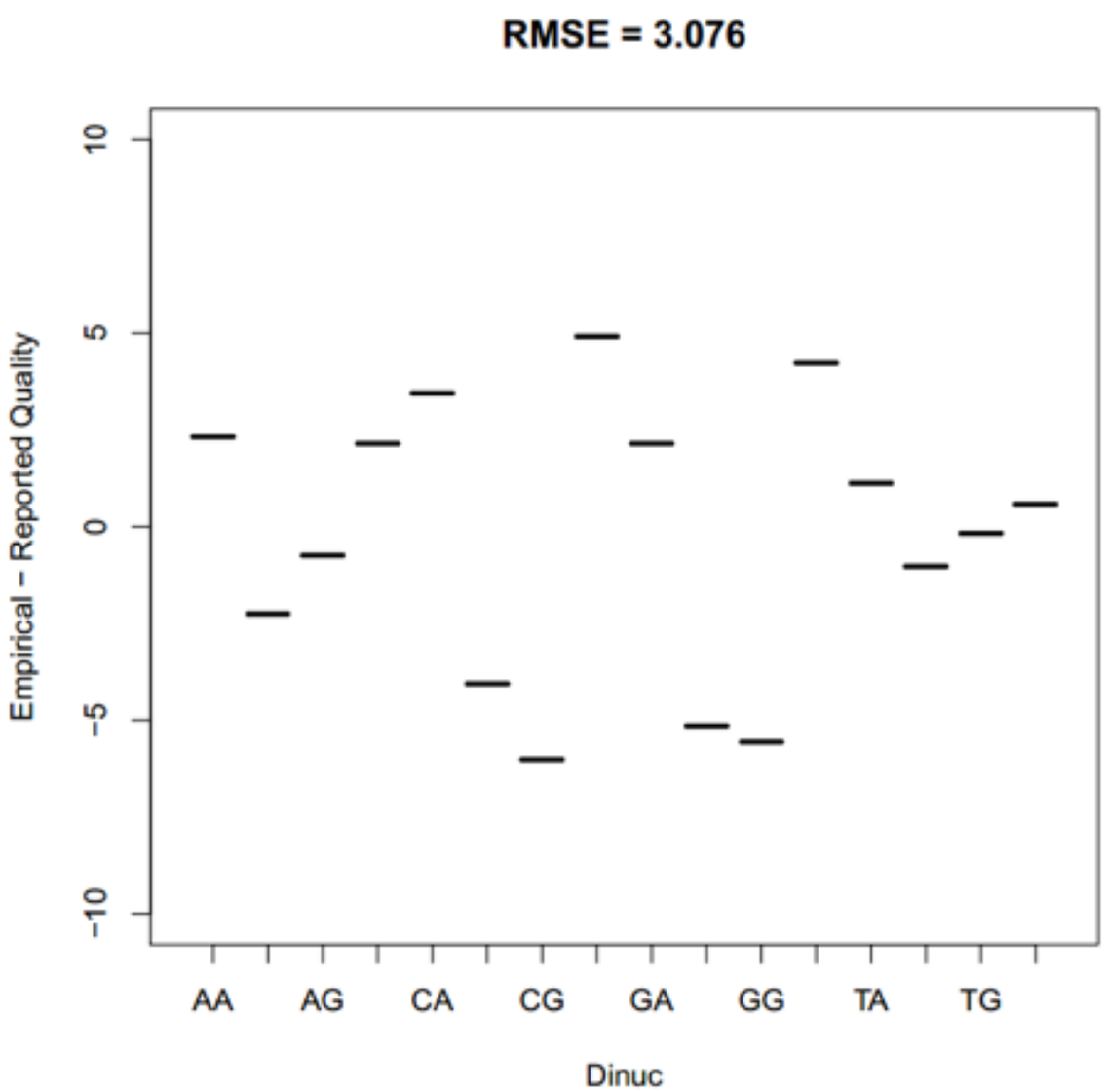


Figure 5: (a) Before BQSR

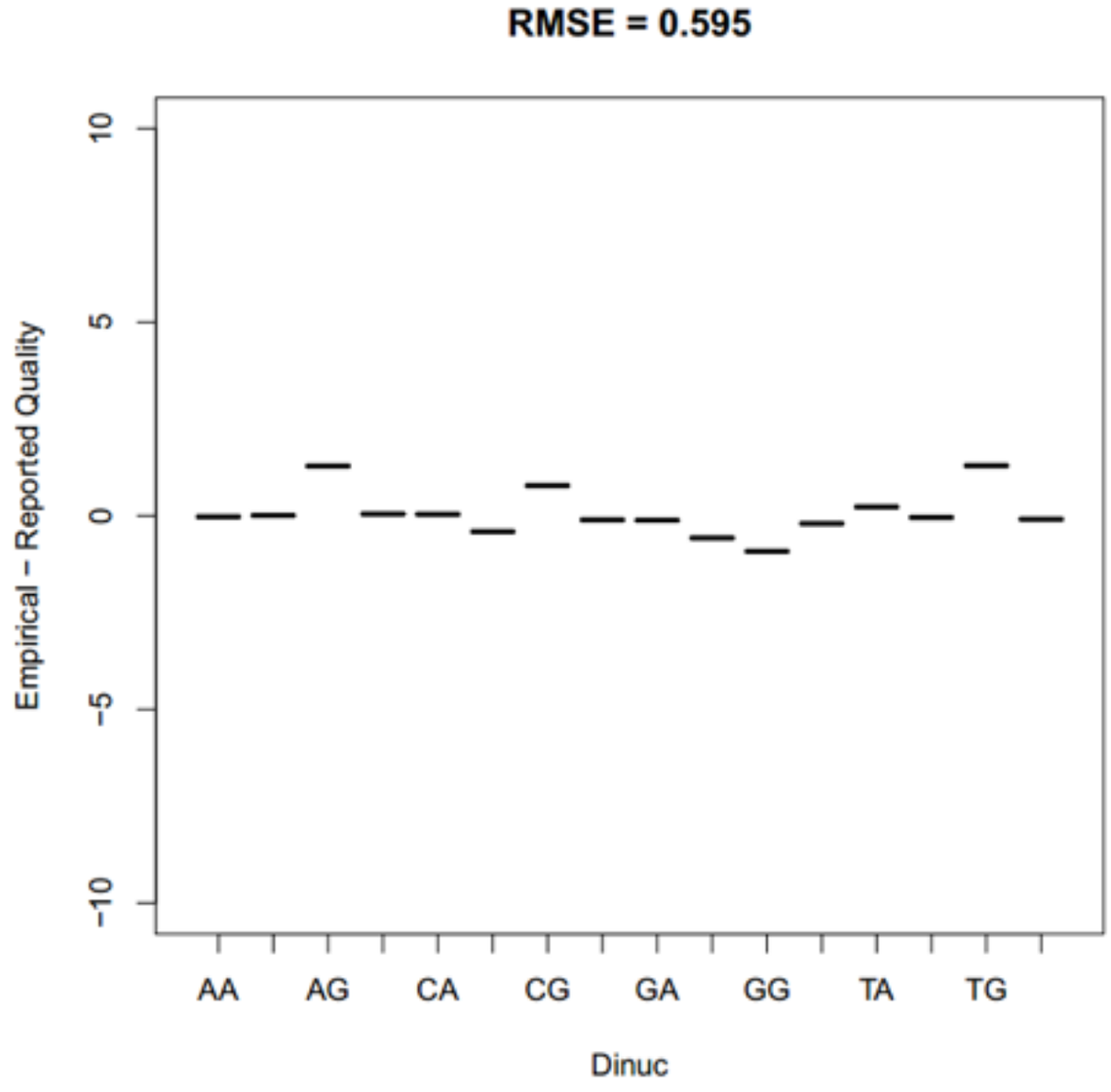


Figure 5: (b) After BQSR

Local realignment of reads

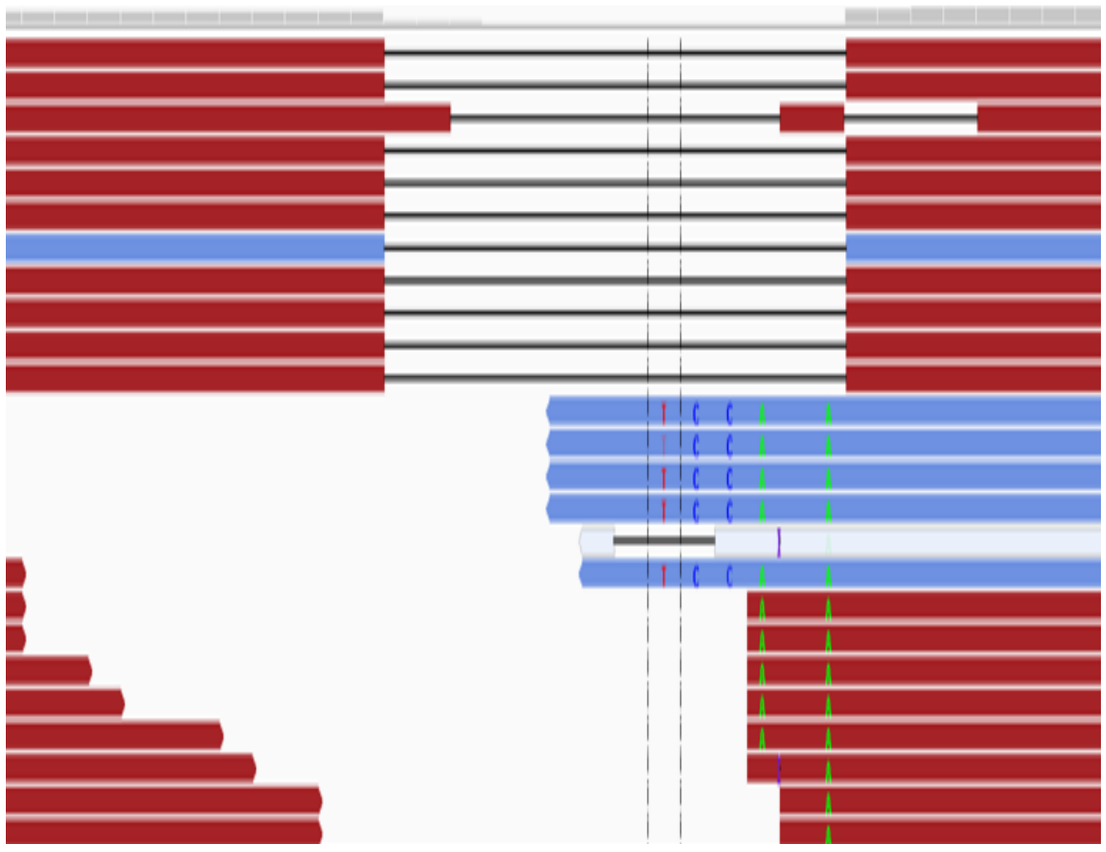


Figure 7: (a) Before Realignment

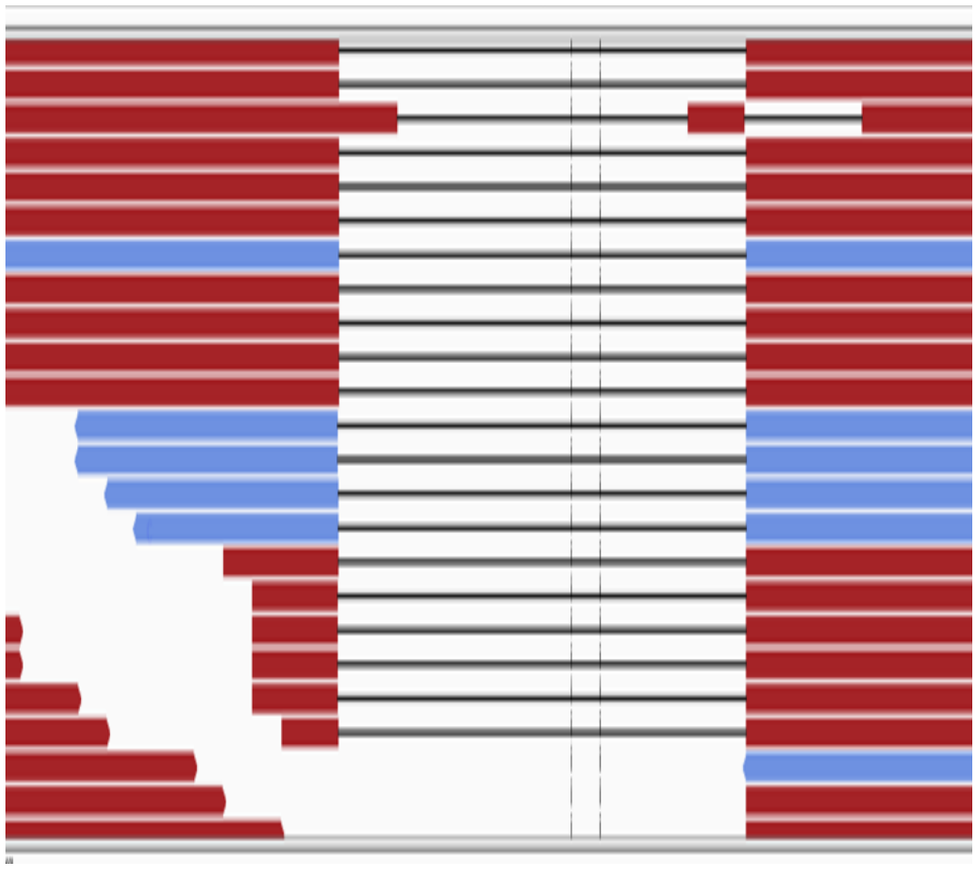


Figure 7: (b) After Realignment

Variant calling - Samtools

- Bcftools (part of samtools package) is used to convert between VCF (variant call format) and BCF (binary VCF), and to call variants
- mpileup output in BCF format can directly piped into bcftools

General VCF format

SAM/BAM + related specifications: https://github.com/samtools/hts-specs

```
##fileformat=VCFv4.2
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G   A      29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T   A      3    q10    NS=3;DP=11;AF=0.017               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   PASS   NS=2;DP=10;AF=0.333,0.667          GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2 2/2:35:4
20     1230237 .         T   .      47   PASS   NS=3;DP=13;AA=T                    GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC G,GTCT 50   PASS   NS=3;DP=9;AA=G                     GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Tag	Description
I16	16 integers: 1 #reference Q13 bases on the forward strand 2 #reference Q13 bases on the reverse strand 3 #non-ref Q13 bases on the forward strand 4 #non-ref Q13 bases on the reverse strand 5 sum of reference base qualities 6 sum of squares of reference base qualities 7 sum of non-ref base qualities 8 sum of squares of non-ref base qualities 9 sum of ref mapping qualities 10 sum of squares of ref mapping qualities 11 sum of non-ref mapping qualities 12 sum of squares of non-ref mapping qualities 13 sum of tail distance for ref bases 14 sum of squares of tail distance for ref bases 15 sum of tail distance for non-ref bases 16 sum of squares of tail distance for non-ref
INDEL	Indicating the variant is an INDEL.
DP	The number of reads covering or bridging POS.
DP4	Number of 1) forward ref alleles; 2) reverse ref; 3) forward non-ref; 4) reverse non-ref alleles, used in variant calling. Sum can be smaller than DP because low-quality bases are not counted.
PV4	P-values for 1) strand bias (exact test); 2) baseQ bias (t-test); 3) mapQ bias (t); 4) tail distance bias (t)
FQ	Consensus quality. If positive, FQ equals the phred-scaled probability of there being two or more different alleles. If negative, FQ equals the minus phred-scaled probability of all chromosomes being identical. Notably, given one sample, FQ is positive at hets and negative at homs.
AF1	EM estimate of the site allele frequency of the strongest non-reference allele.
CI95	Equal-tail (Bayesian) credible interval of the site allele frequency at the 95% level.
PC2	Phred-scaled probability of the alternate allele frequency of group1 samples being larger (,smaller) than of group2 samples.
PCHI2	Posterior weighted chi^2 P-value between group1 and group2 samples. This P-value is conservative.
QCHI2	Phred-scaled PCHI2
RP	Number of permutations yeilding a smaller PCHI2

Bcftools info-tags

Where is the genotype?

- The genotype is decoded in the PL format-tag
- eg: ref=C; alt=A,G; PL=7,0,37,13,40,49
- PL is a list of phred-scaled genotype likelihoods
- From the the given example, the most probable genotype is C/A (10^{0}=1)

GT:	CC	CA	AA	CG	AG	GG
PL:	7	0	37	13	40	49
=:	$10^{-0.7}$	10^{0}	$10^{-3.7}$	$10^{-1.3}$	10^{-4}	$10^{-4.9}$

Variant calling - GATK

- UnifiedGenotyper: multi sample SNP+INDEL caller, accurate SNP calls, multi allelic calls possible
- HaplotypeCaller: recently developed, same SNP detection ability but better INDEL detection

Variant calling - GATK

- -stand_call_conf: The minimum phred-scaled confidence threshold at which variants not at 'trigger’ track sites should be called
- -stand_emit_conf: The minimum phred-scaled confidence threshold at which variants not at 'trigger' track sites should be emitted (and filtered if less than the calling threshold)
- --genotype_likelihoods_model: Genotype likelihoods calculation model to employ --SNP is the default option, while INDEL is also available for calling indels and BOTH is available for calling both together (SNP|INDEL|POOLSNP|POOLINDEL|BOTH) variant calls
- --min_base_quality_score: Minimum base quality required to consider a base for calling
- --max_alternate_alleles: Maximum number of alternate alleles to genotype

Variant annotation– GATK

- **MappingQualityRankSumTest:** This tool calculates the u-based z-approximation from the Mann-Whitney Rank Sum Test for mapping qualities (reads with ref bases vs. those with the alternate allele).
- **AlleleBalance:** The allele balance is the fraction of ref bases over ref + alt bases.
- **BaseCounts:** Count of A, C, G, T bases across all samples
- **ChromosomeCounts:** Allele counts and frequency for each ALT allele and total number of alleles in called genotypes
- **QualByDepth:Variant** confidence (from the QUAL field) / unfiltered depth of non-reference samples.
- **ReadPosRankSumTest:** U-based z-approximation from the Mann-Whitney Rank Sum Test for the distance from the end of the read for reads with the alternate allele
- **MappingQualityZeroBySample:** Count for each sample of mapping quality zero reads
- **HaplotypeScore:** Consistency of the site with two (and only two) segregating haplotypes.
- **LowMQ:** Triplet annotation: fraction of MAQP == 0, MAPQ < 10, and count of all mapped reads
- **RMSMappingQuality:** Root Mean Square of the mapping quality of the reads across all samples.
- **BaseQualityRankSumTest:** U-based z-approximation from the Mann-Whitney Rank Sum Test for base qualities