

# Short Read Mapping and File Formats

# What are we doing today?

- I. Examining Read Quality
- II. Quality/Adapter Trimming
- III. Read Mapping
- IV. SAM, BAM and CRAM format

# Quality Control

## Goals

- Is my data of sufficient quality to use?
- The instrument assigns a confidence value to each base. Are the bases high quality overall?
- Does the complexity look normal?
  - PCR and library prep problems can lead to duplication of the same sequences over and over
- Are there adapters or other over-represented sequences?
- Are there lane batch effects?

# FASTQC



Babraham Bioinformatics

Accepts input formats:

- FastQ (all quality encoding variants)
- GZip compressed FastQ
- SAM
- BAM

Does a 12 point analysis of quality

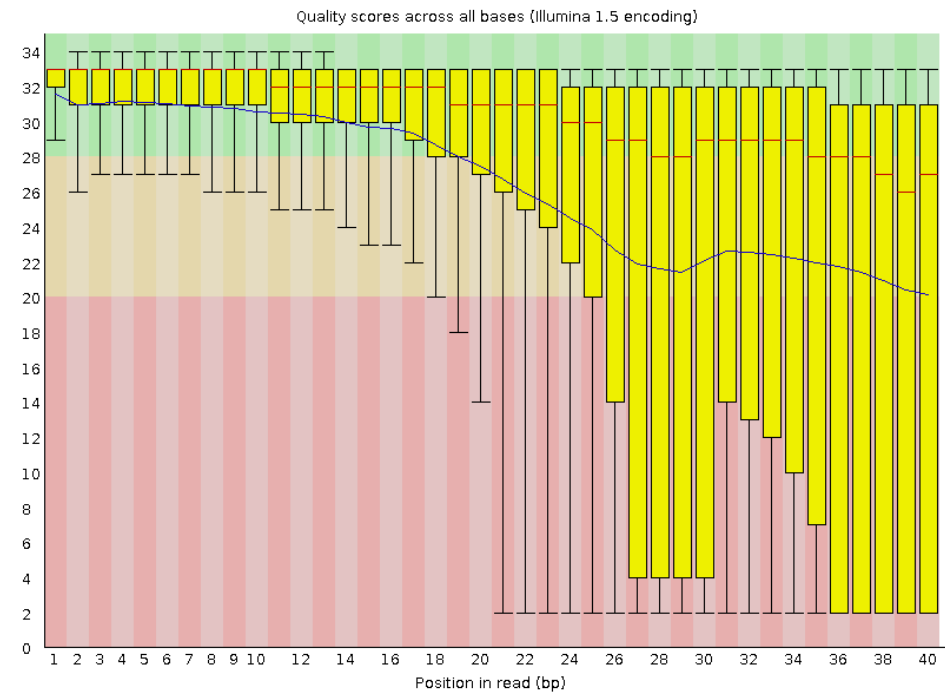
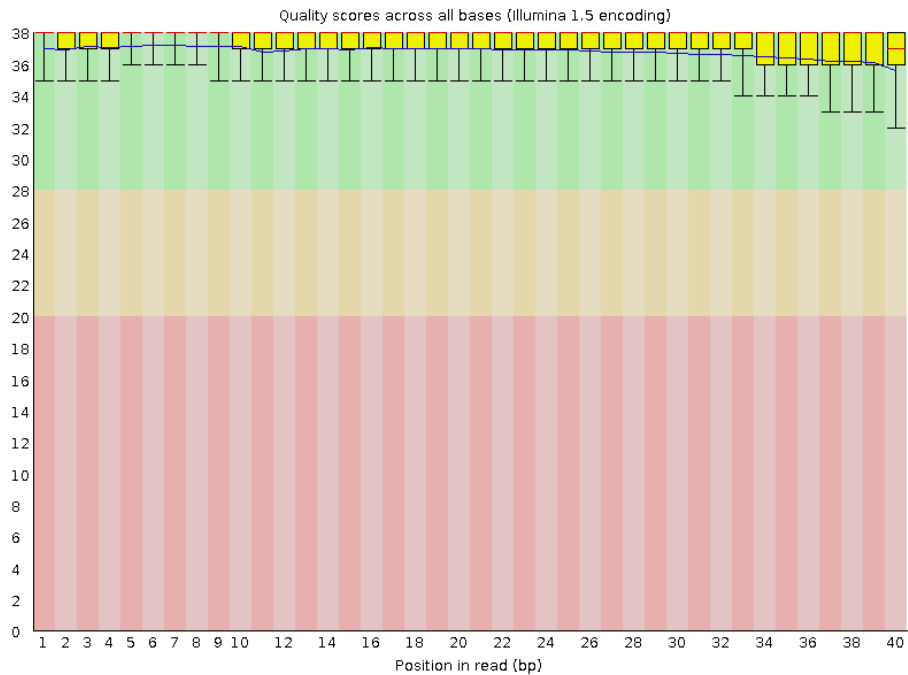
Generates an html output file

- ✓ [Basic Statistics](#)
- ✓ [Per base sequence quality](#)
- ✓ [Per tile sequence quality](#)
- ✓ [Per sequence quality scores](#)
- ! [Per base sequence content](#)
- ✓ [Per sequence GC content](#)
- ✓ [Per base N content](#)
- ✓ [Sequence Length Distribution](#)
- ✓ [Sequence Duplication Levels](#)
- ✓ [Overrepresented sequences](#)
- ✓ [Adapter Content](#)
- ! [Kmer Content](#)

# FASTQC



Babraham Bioinformatics

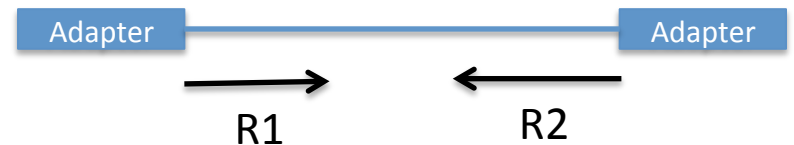


# Trimming

- From the quality control step, we know where the problems are
- All illumina reads tend to have degrading quality at the end of the read
- Get rid of the bad data, keep the good data
  - Cut adapter and other Illumina-specific sequences from the read.
  - Trim off low quality bases
  - Drop a read entirely if is too low quality or too short

# Trimmomatic

- Optimized for Illumina NGS
- **Very flexible**
- Handles paired end data well
- Threaded
- Detects adapter read through
- No read through:
- Read through:



Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatics*, btu170.

# Trimmomatic

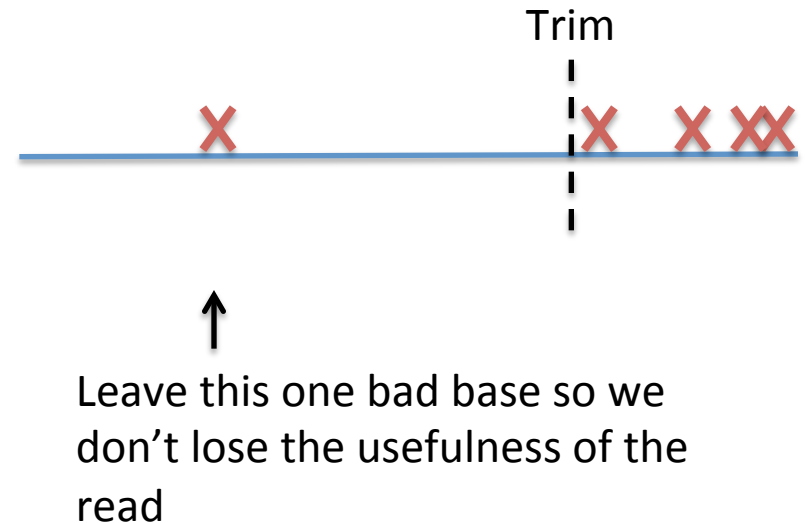
The current trimming steps are:

- **ILLUMINACLIP**: Cut adapter and other illumina-specific sequences from the read. Comes with basic Illumina adapters, make sure yours are in there or add yours!
- **SLIDINGWINDOW**: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- **LEADING**: Cut bases off the start of a read, if below a threshold quality
- **TRAILING**: Cut bases off the end of a read, if below a threshold quality
- **CROP**: Cut the read to a specified length
- **HEADCROP**: Cut the specified number of bases from the start of the read
- **MINLEN**: Drop the read if it is below a specified length after trimming



# Trimmomatic

- Maximum Information Quality Filtering:
- Retain low-quality bases early in a read in order to make sure the read is sufficiently long to be informative
- Trimming process becomes more strict later in the read



# Skewer

- Faster than trimmomatic
- A bit less flexible
- If your data is in pretty good shape and needs only basic trimming, this is a great option

<https://github.com/relipmoc/skewer>

Jiang, H., Lei, R., Ding, S.W. and Zhu, S. (2014) Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. BMC Bioinformatics, 15, 182.

# Trimming is critical for downstream success

- Trimming is important for all read sets, but truly critical for those with lower quality
- Resequencing applications
- Alignment of reads with and without trimming using BWA

Bolger et al., 2014.  
Trimmomatic: a flexible  
trimmer for Illumina sequence  
data. Bioinformatics

## Higher quality data

- 79% aligned if no trimming done
- 83% aligned after trimming

## Lower quality data

- 7% aligned if no trimming done
- 80% aligned after trimming

# Trimming is critical for downstream success

- Assembly applications
- Same trend

## Higher quality data

- 58% increase in contig N50 size (from 60kbp to 95kbp)
- 28% increase in maximum contig size

## Lower quality data

- 77% increase in contig N50 size (from 100kbp to 177kbp)
- 55% increase in maximum contig size

Bolger et al., 2014.  
Trimmomatic: a flexible  
trimmer for Illumina sequence  
data. Bioinformatics

# Trimming

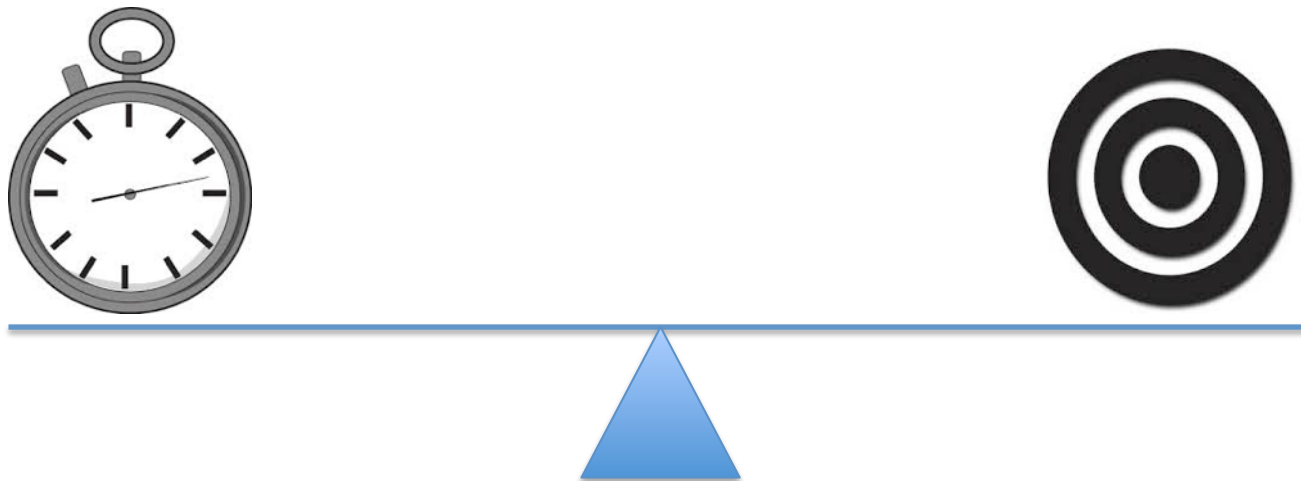
- Current community wisdom:
  - Quality trimming reduces error
  - But also reduces content and contiguity
- Gentle trimming is preferred – many times the defaults are too stringent, you will lose lots of data!
- Application matters
  - For mapping and variant calling, possibly no trimming (phred 5)
  - For assembly, a bit more trimming is good (phred 10)

**LETS START THE LAB AND COME  
BACK SOON...**

## **II. READ MAPPING**

# But we've already covered alignment.

- Alignment methods must have tradeoffs for speed vs accuracy
- Depending on the application, may want to make different tradeoffs
- Global vs local
- Different types of alignment objectives lead to different categories of aligners





# Types of Alignment software

From wikipedia (open source and commercial):

- Database search only (23 including BLAST)
  - Aligning a query or set of queries to a set of database sequences
- Pairwise alignment (43 including dotplotters)
  - Aligning two sequences to each other
- Multiple sequence alignment (47)
  - Aligning 3 or more sequences together
- Genomic analysis (16)
  - Aligning genome length sequences to each other, or aligning cDNA to a genome
- Motif finding
  - Finding motifs in a database (17)

Many aligners  
fit into more  
than one  
category

# Short Read Mappers

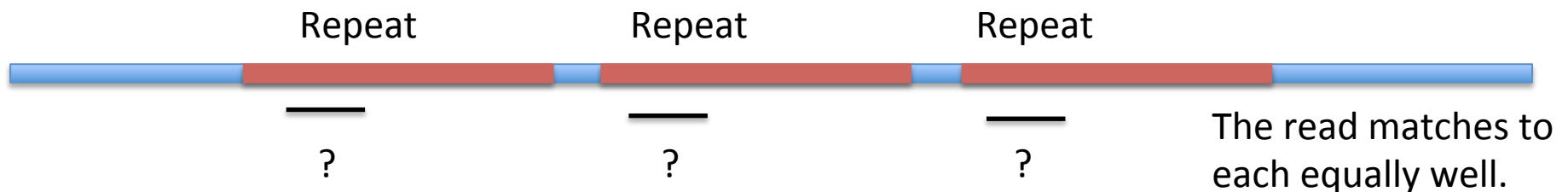
- BLAST is much faster than original algorithms (Smith Waterman for example)
- Still too slow for the amount of data produced by NGS technology
- Resequencing usually involves comparing very similar sequences (>90% identity in residues) to a reference genome
- Software that leverages this high percent identity can be faster
- Can generally utilize a global strategy instead of a local strategy

# Short Read Mappers

- Orders of magnitude faster than BLAST
- several tens of millions of reads mapped per hour per CPU
- Only matches of 95% identity or greater are found
- Usually only output the best hit or the set of hits all equivalently good
  - The point is usually to find the origin in the reference genome
  - Other genomic regions of lower identity are not considered useful

# Uniqueness

- Some reads can be mapped uniquely to the reference
- Some map to multiple locations
- Multiply mapped reads are difficult to apply to downstream applications
  - RNASeq – which gene do they represent?
  - SNP – which location carries the substitution?
- How to deal with multiply mapped reads?
  - Throw them away
  - This introduces bias and ignores real genomic regions that may be biologically important – will discuss more for RNASeq



# Clever Tricks to find “Best” Alignments

- Use the quality values
  - Penalize mismatches at high quality bases more than mismatches at low quality bases
- Paired End information
  - If one read does not map uniquely, but the other does, use that information to place the non-unique one
  - Need to know your insert size

# Decisions for the end user

- How many mismatches are allowed for a read to be considered mapped?
  - Heterozygosity between sample and reference
  - Incomplete/low quality reference
- How many matches to report?
  - Does your downstream analysis need/want to include multiple matches?

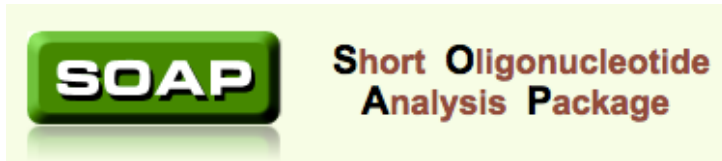
**Explore the documentation and parameters for your software of choice**

**Is it doing what you think its doing?**

# Lots of choices

Many, many software packages

Blat  
BWA  
Eland  
GSNAP  
HISAT2  
Maq  
RMAP  
Stampy  
SHRiMP



**mosaik**



Prize for best named:  
VelociMapper



What's in the box matters.

# How to choose?

- Good documentation
- Memory efficient
- Responsive mailing list or help forum
- Maintained and updated when bugs are found
- Too many emerge each year for the literature to keep up, but currently, most popular ones are very similar in mapping rate and time
  - BWA
  - HISAT2



# What mappers have in common:

## Indexing Strategies

- Usually, the first step is to transform part of the data into a more suitable form for fast searching
- Indexing – creating a glossary or look up table
- Without indexing you would have to scan everything each time you did a search
- Consider web search engines



# Burrows-Wheeler Aligner

- Has three algorithms
- Individual chromosomes cannot be longer than 2GB
- Output in SAM format



# Burrows-Wheeler Aligner

<http://bio-bwa.sourceforge.net/>

- three algorithms, but one is the most common:
- **BWA-MEM**
  - Use for any sequences greater than 70bp up to 1Mb
  - Will work with reads with
    - 2% error for 100bp
    - 3% error for a 200bp
  - Has split read support – i.e. “It may produce multiple primary alignments for different part of a query sequence.”
    - structural variations, gene fusion or reference misassembly
  - By reporting multiple alignment locations, it does not always work well with other software
  - Can fix by using the `–M` flag

### **III. SAM, BAM AND CRAM FORMAT**

# SAM Format

- SAM = Sequence Alignment/Map format
- Tab delimited plain text
- Store large nucleotide sequence alignments
  - Alignment of every read
  - Including gaps, SNPs and structural variants
  - Pairing of reads
  - Can record more than one alignment location in the genome
  - Stores quality values
  - Stores information about duplication

# SAM Format

## Strengths

- Flexible
- Useful for operations on very large sequences
- Extremely detailed documentation
  - <https://samtools.github.io/hts-specs/SAMv1.pdf>
- Manipulations can be done with the software samtools

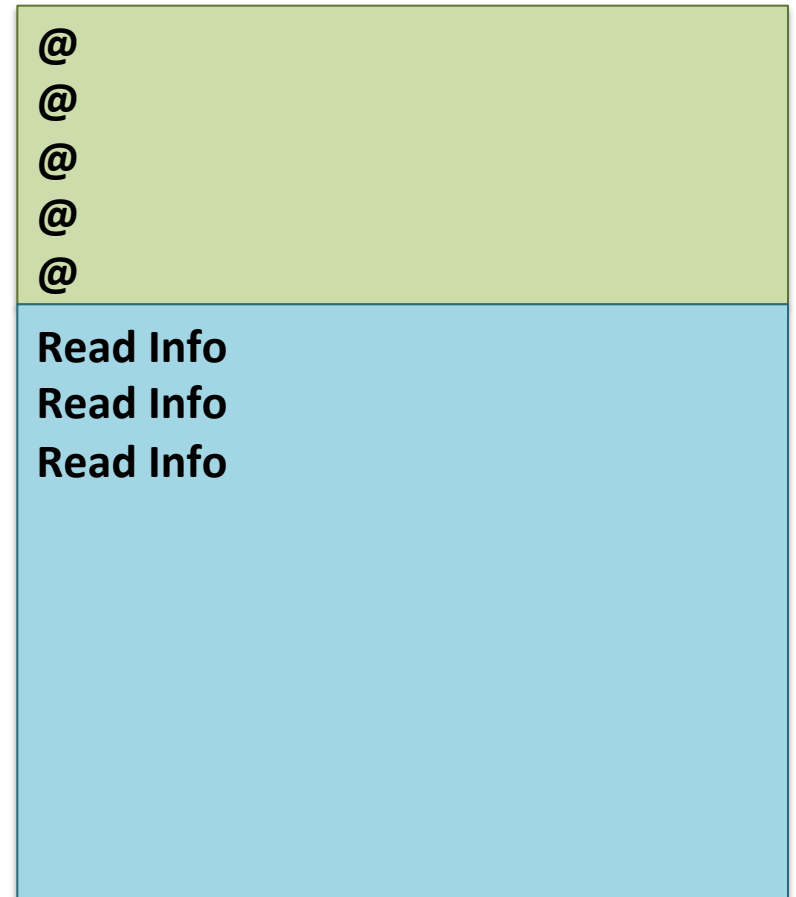
# SAM - Header

- Structure

- Optional Header at top of file



- Alignment information



# SAM - Header

- Header lines start with @ symbol
- Always at top of file
- Contain lots of information about what was mapped, what it was mapped to, and how (metadata)
  - the version information for the SAM/BAM file
  - whether or not and how the file is sorted
  - information about the reference sequences
  - any processing that was used to generate the various reads in the file
  - software version



# Simple Header

@HD = first line

VN = version of SAM format

SO = sort order (this is sorted by coordinates)

@HD VN:1.5 SO:coordinate

@SQ SN:ref LN:45

@SQ = reference sequence

SN = Sequence reference Name

LN = sequence reference length

# Alignment Line

- Below the headers are the alignment records
  - Tab-delimited fields
- 
- 1 QNAME Query template/pair NAME
  - 2 FLAG bitwise FLAG
  - 3 RNAME Reference sequence NAME
  - 4 POS 1-based leftmost POSition/coordinate of clipped sequence
  - 5 MAPQ MAPping Quality (Phred-scaled)
  - 6 CIGAR extended CIGAR string
  - 7 MRNM Mate Reference sequence NaMe ( '=' if same as RNAME)
  - 8 MPOS 1-based Mate POSition
  - 9 TLEN inferred Template LENgth (insert size)
  - 10 SEQ query SEQuence on the same strand as the reference
  - 11 QUAL query QUALity (ASCII-33 gives the Phred base quality)
  - 12+ OPT variable OPTional fields in the format TAG:VTYPE:VALUE

Lets unpack this alignment line, taken from a SAM file:

```
SRR030257.2000020    83 gi|254160123|ref|
NC_012967.1|329575260 36M  =  3295706-82
TGCTGGCGGCGATATCGTCCGTGGTTCCGATCTGGT
?%<91<?>>??AAAAAAAAAAAAAAAAAAAAAAAAAAAA
XT:A: NM:i:0  SM:i:37  AM:i:37  X0:i:1X1:i:0
XM:i:0  XO:i:0  XG:i:0  MD:Z:36
```

# SAM Field 1

Query name

SRR030257.2000020

## Field 2: Flag

83

Bit	Description
1	0x1 template having multiple segments in sequencing
2	0x2 each segment properly aligned according to the aligner
4	0x4 segment unmapped
8	0x8 next segment in the template unmapped
16	0x10 SEQ being reverse complemented
32	0x20 SEQ of the next segment in the template being reverse complemented
64	0x40 the first segment in the template
128	0x80 the last segment in the template
256	0x100 secondary alignment
512	0x200 not passing filters, such as platform/vendor quality controls
1024	0x400 PCR or optical duplicate
2048	0x800 supplementary alignment

64 + 16 + 2 + 1

1 = Read is paired

2 = Read mapped in proper pair

16 = Read mapped to reverse strand

64 = First in pair

Look up aSAM flag: <https://broadinstitute.github.io/picard/explain-flags.html>

# SAM Field 3

Reference sequence name (useful especially if you have multiple chromosomes)

```
gi | 254160123 | ref | NC_012967.1 |
```

# SAM Field 4

Position- 1-based leftmost mapping POSition of the first matching base

3295752

# SAM Field 5

## Mapping Quality

- equals  $-10 \log_{10} \text{Pr}\{\text{mapping position is wrong}\}$ , rounded to the nearest integer
- Same as phred!
- Probability of 99.9% = map quality of 30
- Probability of 0% = map quality of 0
- value 255 indicates that the mapping quality is not available.

60

.000001% probability wrong



# SAM Field 6

## 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

36M 36 nucleotides match (perfect match)

8S28M 8 nucleotides clipped, 28 match

# More CIGAR

Aligning these two:

RefPos:	1	2	3	4	5	6	7		8	9	10	11	12	13	14	15	16	17	18	19
Reference:	C	C	A	T	A	C	T		G	A	A	C	T	G	A	C	T	A	A	C
Read:					A	C	T	A	G	A	A	C	T	G	G	C	T			

Position:

5


CIGAR:

3M1I3M1D5M

# SAM Field 7

Reference sequence for the next read in the template

- For a forward read, this is the reference where the reverse read maps
- For a reverse read, this is the reference where the forward read maps

 = reverse read maps on the same reference

# SAM Field 8

Position where the next read maps

3295706

(Forward read mapped at 3295752.  
Remember the forward read mapped  
to the reverse strand)

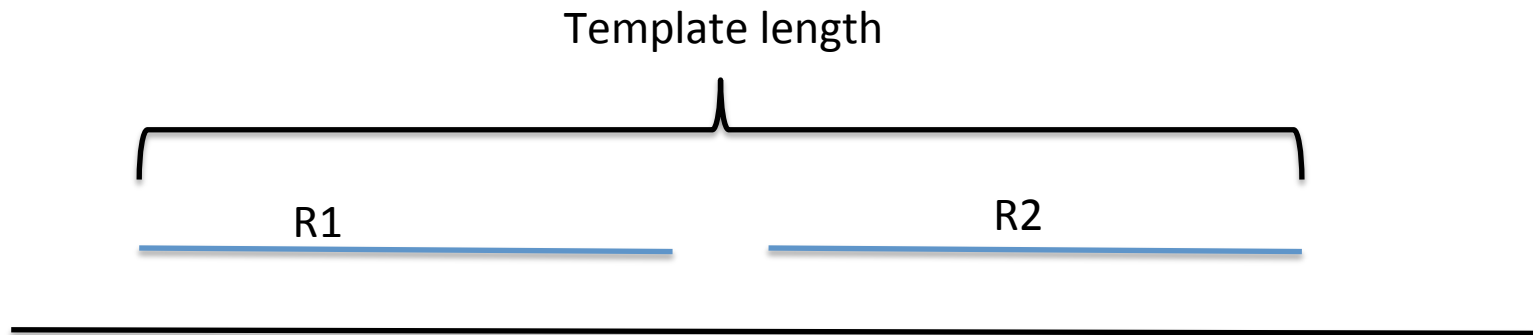
# SAM Field 9

Observed template length

From leftmost base to rightmost base

Negative if this read is the rightmost read

-82



# SAM Field 10

Sequence of the read

TGCTGGCGGCGATATCGTCCGTGGTTCCGATCTGGT

# SAM Field 11

Quality of the read

```
?%<91<?>>??AAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

# SAM Field 12

Optional MORE information

TAG:TYPE:VALUE format

```
XT:A:U   NM:i:0   SM:i:37  AM:i:37  X0:i:1  
X1:i:0   XM:i:0   XO:i:0   XG:i:0   MD:Z:36
```

Anything with an X is specified by the user or by the mapping software, and is not part of the SAM spec.



# Decipher the last fields

XT:A:U	One of Unique/Repeat/N/Mate-sw
NM:i:0	Edit distance to the reference
SM:i:37	Template-independent mapping quality
AM:i:37	Smallest template-independent mapping quality of other segments
X0:i:1	Number of best hits
X1:i:0	Number of suboptimal hits found by BWA
XM:i:0	Number of mismatches in the alignment
XO:i:0	Number of gap opens
XG:i:0	Number of gap extensions
MD:Z:36	String for mismatching positions

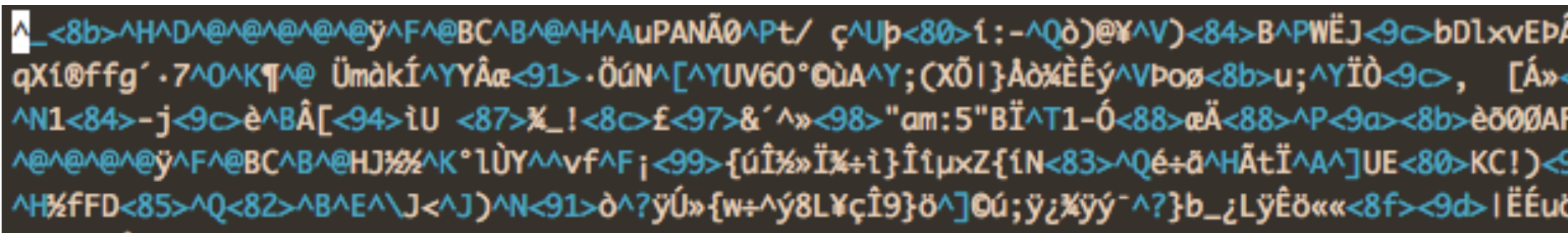
# SAM

Example sam with one read:

```
@SQ SN:gi|254160123|ref|NC_012967.1| LN:4629812
@PG ID:bwaPN:bwa VN:0.7.12-r1039 CL:/lustre/projects/
rnaseq_ws/apps/bwa-0.7.12/bwa sampe ../raw_data/
NC_012967.1.fasta aln_SRR030257_1.sai
aln_SRR030257_2.sai ../raw_data/SRR030257_1.fastq ../
raw_data/SRR030257_2.fastq
SRR030257.1 99 gi|254160123|ref|NC_012967.1|
950180 60 36M = 950295 151
TTACACTCCTGTTAATCCATACAGCAACAGTATTGG
AAA;A;AA?A?AAAAA?;?A?1A;;????566)=*1 XT:A:U
NM:i:1SM:i:37 AM:i:25 X0:i:1 X1:i:0 XM:i:1 XO:i:0
XG:i:0 MD:Z:32C3
```

# BAM Format

- Sister format to SAM
- BAM – Binary version of SAM
- compressed **BGZF** (Blocked GNU Zip Format) - a variant of GZIP (GNU ZIP),
- files are bigger than GZIP files, but they are much faster for random access
- Can index and then look up information embedded in the file with decompressing the whole file
- up to 75% smaller in size
- Not readable by people

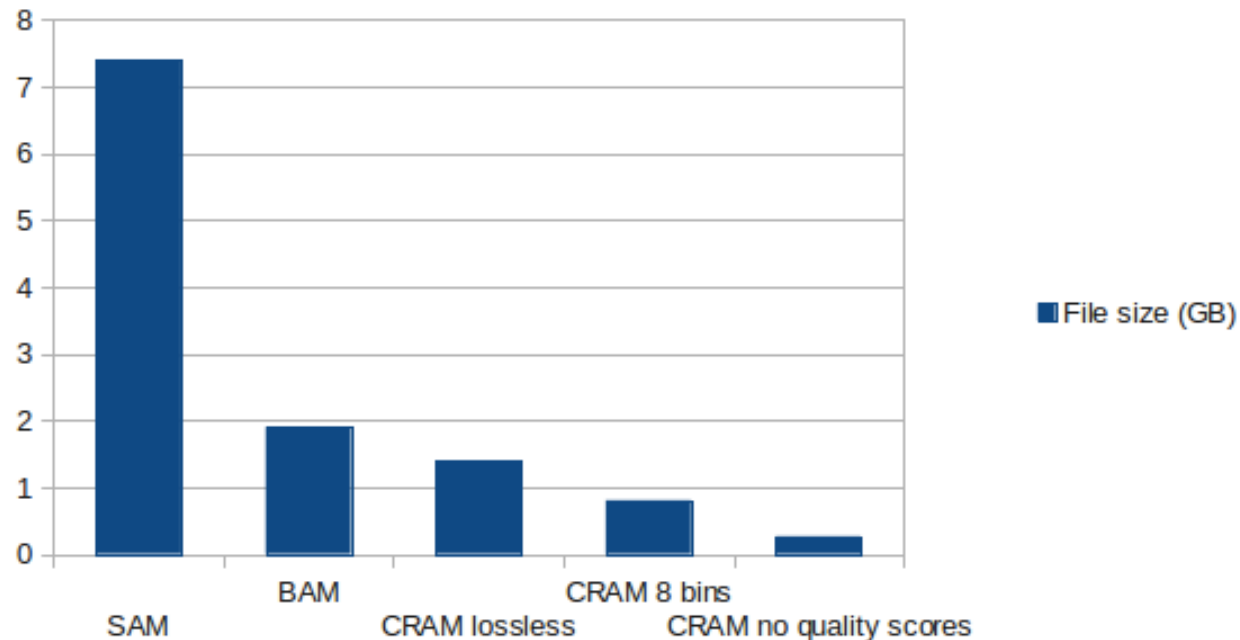


```
^_<8b>^H^D^@^@^@^@^@y^F^@BC^B^@^H^A^uPAN^0^P^t/ ç^Up<80>í:-^Qð)^@Y^V)^<84>B^PWËJ<9C>bDl^xvEp^  
qXí@ffg'.7^0^K^@ ÜmàkÍ^YY^Â^<91>·ÖúN^[^YUV60°0ùA^Y;(XÖI}Àð%ÈËý^Vpø<8b>u;^YÏÒ<9C>,[Á»  
^N1<84>-j<9C>è^BÂ[<94>iU <87>%_!<8C>f<97>&'^»<98>"am:5"BÏ^T1-Ó<88>æÄ<88>^P<9a><8b>èðððAF  
^@^@^@^@y^F^@BC^B^@HJ%K^lÙY^^vf^F; <99>{úÎ%»Î%+i}ÎîµxZ{íN<83>^Qé+ð^HÄtÏ^A^]UE<80>KC!)<9  
^H%FFD<85>^Q<82>^B^E^\\J<^J)^N<91>ð^?ýÚ»{w+^ý8L¥çÎ9}ð^]0ú;ý¿%ýý^-^?}b_¿LÿÊö««<8f><9d>|ËÉuð
```

# CRAM

- Introduced in 2011 by EMBL/EBI
- Even smaller and more efficient than BAM files
- Rare

EBI has a cram toolkit  
<https://www.ebi.ac.uk/ena/software/cram-toolkit>



Fritz, Markus Hsi-Yang, et al. "Efficient storage of high throughput DNA sequencing data using reference-based compression." *Genome research* 21.5 (2011): 734-740.

<http://www.uppmax.uu.se/using-cram-to-compress-bam-files-on-uppnex>