Bioinformatics for Next-Generation Sequencing
ICIPE, Nairobi, November 2015

Read Mapping – Concept & Approaches

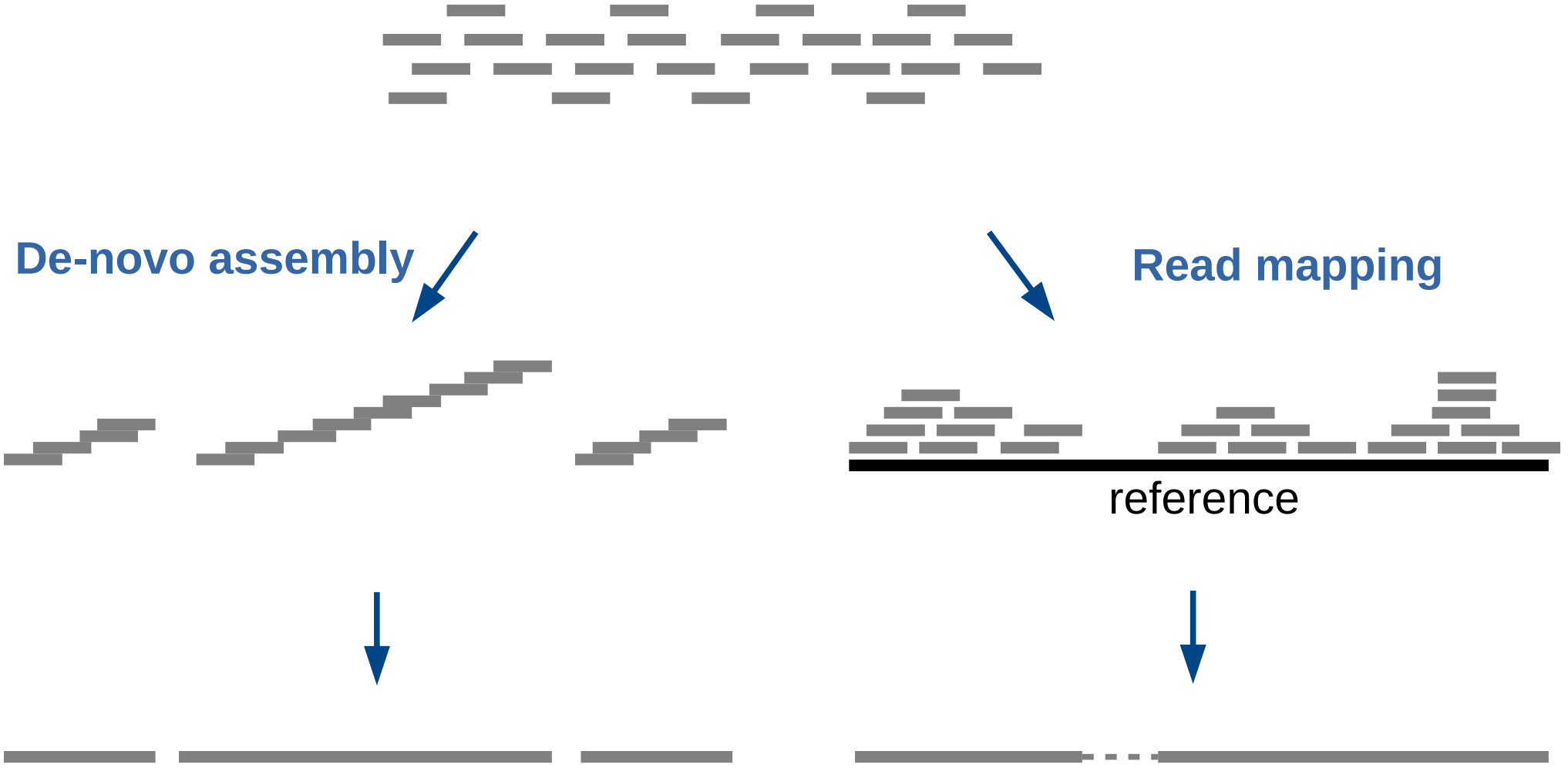
Simon Martin

Department of Zoology
University of Cambridge

Outline

- Why read mapping, and what is it?
- Challenges
- How read mappers work
- Choosing a mapper
- What goes in? - fastq data
- What comes out? - SAM/BAM files

So you have your next-gen data... now what?



Why do we map?

When we have

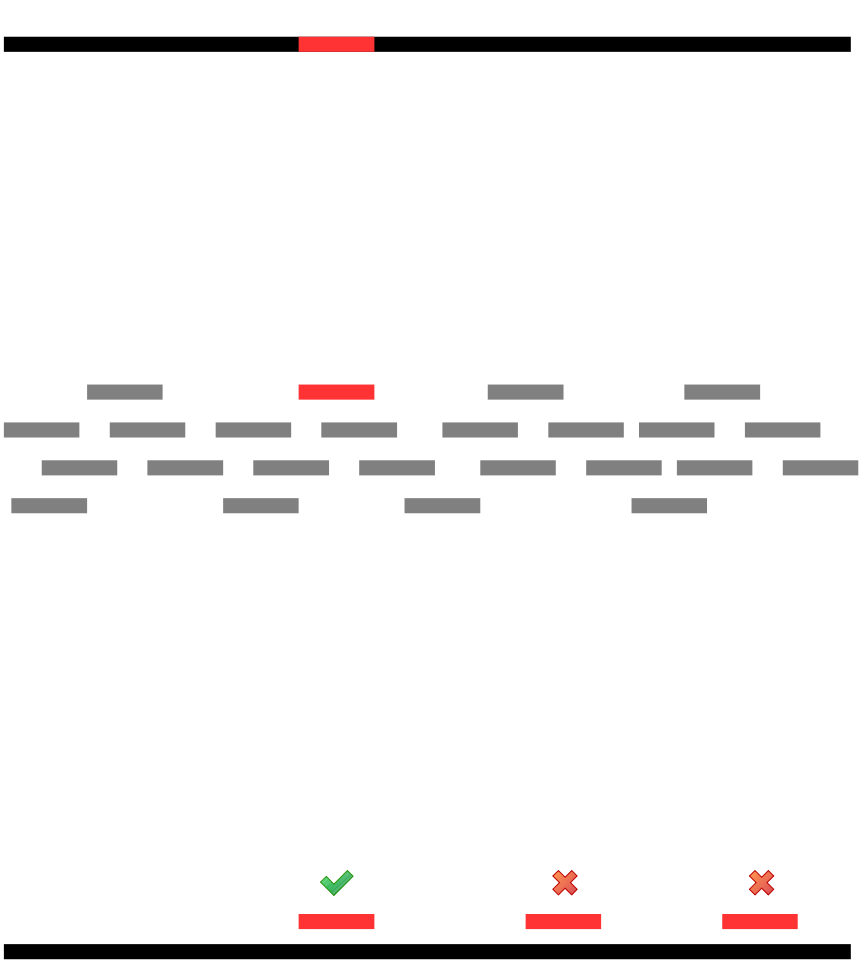
- sequence **reads**
- a **reference** genome, transcriptome, or region
- From the **same (or similar)** species

and we want...

- To know where each read belongs
- To infer the sequence (genotype) of the template (i.e. the samples we have sequenced)

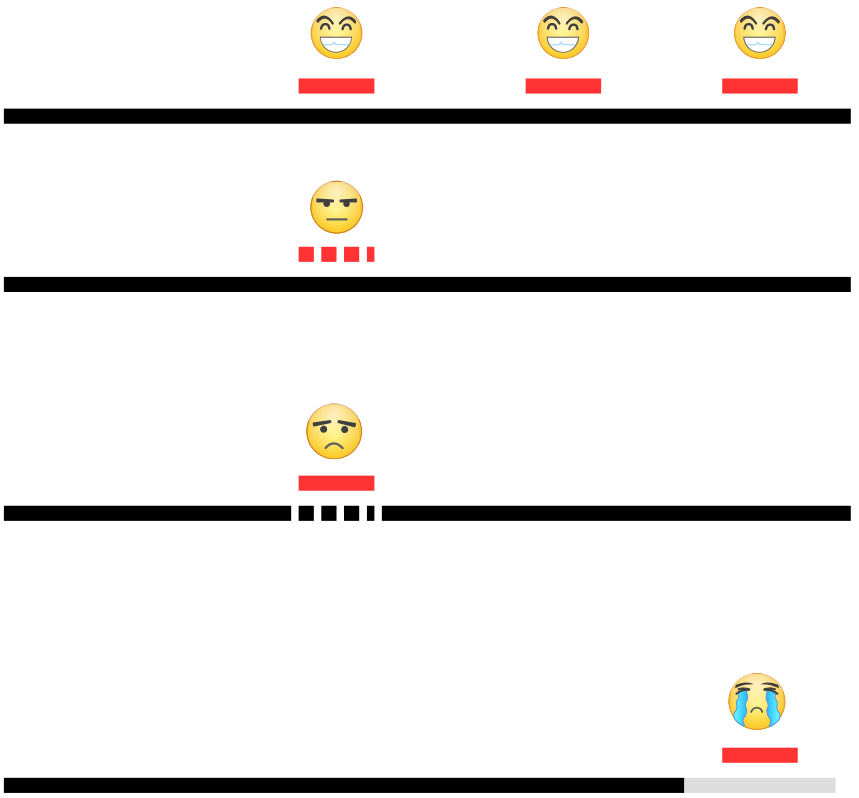
What is read mapping?

- Each sequence read has a single location in the template genome
- We need to find its homologous position in a reference sequence
- This position is unknown and must be inferred from the read itself



Challenges

- Multiple optimal mappings
- Sequencing errors
- Real differences between template and reference
- Missing regions in the reference genome



Multiple mappings

- In theory a read should have only one perfect match
  - Read of length *n* has 4^n possible sequences
  - 100 bp read should occur by chance every 1.6×10^{60} bp
  - Read of 16 bp should be *unique* in the human genome (3.2 GB).
- But...
  - Genomes are repetitive (two thirds of human genome!)
  - Base usage is biased

- Skewed GC content increases the chance of repeats dramatically!

Sequencing error

	Technology	Read length	Approx. error rate
	Sanger	900	1/100 000
Next Gen	Illumina HiSeq	100+100 Paired	1/1000
	Ion Torrent PGM	200-400	1/100
	PacBio RS	15000	1/10

Wikipedia.org

Differences between template and reference

- Intra or inter-specific variation
  - Diversity / divergence time
  - Depends where in the genome you look

Sample / Reference	Differences
Human / Human	1/1000
Human / Neanderthal	1/500 ?
D. melanogaster / *D. melanogaster*	1/100
D. simulans / *D. melanogaster*	1/40

- RNA modifications

Aligning a read

```
read        CGCCAGACT--TAGTGTGCTCTG
              || |||  ||  ||| ||||
reference   ATAAGCACACTTCTAA-GTG-TCTGGGTGC
```

- Many possible solutions to the problem (unless sequences are identical)
- Alignments can be scored according to:
  - Matches
  - Mis-match
  - Insertions and deletions (and their length)
  - Clipping

Aligning a read

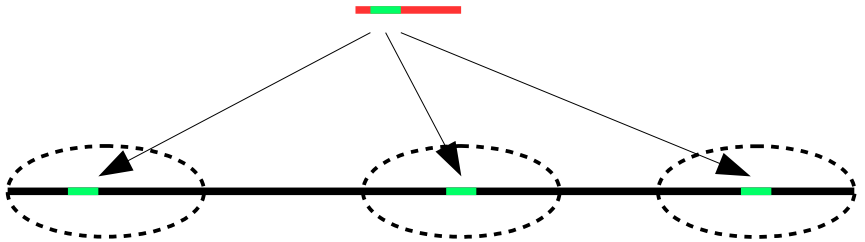
```
CAGACT--TAGAGTCCTCTG
||  |||   ||   ||| ||||
CACACTTCTAA-GTC-TCTG
```

```
CAGACTTAGAGTCCTCTG
||  ||||   |  |||  |||
CACACTTCTAAGTCTCTG
```

- Which of the alignments is correct / better?
- Best score depends on scoring algorithm
- Nearly infinite number of solutions to an alignment problem
- Evaluating all alignments too expensive for mapping millions of reads

Mappers use heuristics

- Usually involves two steps:
  - Find exact/near-exact matches for one or more sub-strings (“seeds”)
  - Only search for optimal alignments in regions with seed matches



- Seed matching can also be sped up – e.g. “hash table”, Burrows-Wheeler Transform
  - Compresses genome, speeds up searches for string matches

Some Read Mappers

- BarraCUDA
- BLASR
- Bfast
- Bowtie2
- Brat
- BWA
- CLCbio
- Eland
- GenomeMapper
- GnuMap
- Karma
- Novoalign
- MAQ
- Mosaik
- MrFast
- Pash
- PASS
- PerM
- RazerS
- REAL
- RMAP
- Segemehl
- Slider
- SSHANA
- SOAP
- SplacerS
- Stampy
- Tophat
- Vmatch
- Zoom

Differ in:

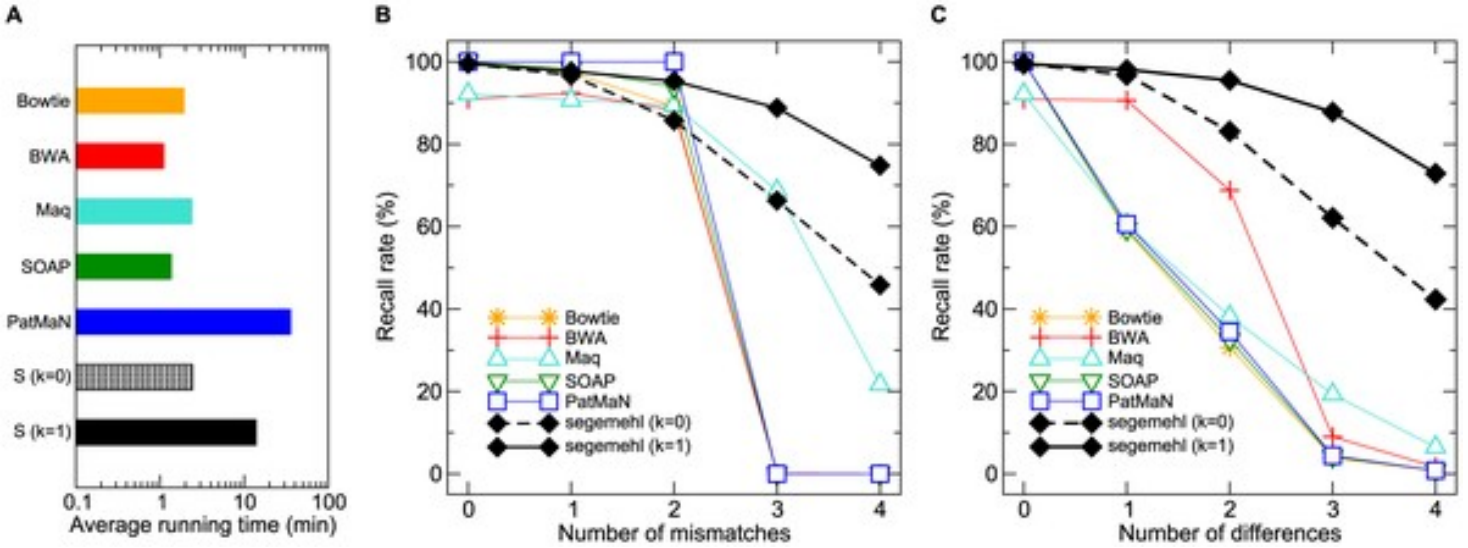
- Number of seeds
- Mismatches in seeds
- Indexes / transforms for seed search
- Runtime vs precision
- Allowance for alignment gaps
- Multiple mappings per read
- Reporting mapping quality
- Support for types of sequence data

Choosing a mapper / aligner

- What type of sequence data do I have?
- How similar is my template to the reference?
- How accurate do I want my alignments to be?
- Am I expecting many indels, how large?
- How much data / time do I have?

Choosing a mapper

Comparison of recall rates and running time for several short read aligners.



Hoffmann et al. (2009) PLoS Comput Biol 5(9): e1000502

Choosing a mapper

Precision and recall by amount of variation for 4 datasets, by polymorphism: (number of SNPs, Indel size).

		(0,0)		(1,0)		(2,0)		(4,0)		(0,3)		(1,3)		(2,3)		(4,3)	
	Program	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.
50 paired	SHRiMP	99.7	96.6	99.6	96.4	99.6	95.7	99.3	89.3	99.3	93.5	99.3	90.6	98.6	85.7	97.6	69.7
	BFAST	95.4	93.8	94.3	91.6	92.6	86.2	87.0	63.5	91.6	78.8	89.3	71.8	86.8	61.9	80.7	38.8
	BWA	91.1	65.2	85.4	27.7	64.7	5.4	17.7	0.3	62.0	4.4	49.2	1.5	29.6	0.4	11.9	0.1
	Bowtie	97.5	46.6	97.5	11.1	96.9	1.0	0.0	0.0	97.1	1.3	100	0.2	100	0.0	0.0	0.0
75 paired	SHRiMP	99.6	97.5	99.6	97.2	99.6	97.3	99.6	96.9	99.3	96.6	99.5	96.9	99.4	96.5	99.2	94.5
	BFAST	97.4	97.1	97.1	96.8	96.8	96.5	95.9	94.5	96.4	96.0	96.0	95.5	95.9	94.8	94.1	89.5
	BWA	93.2	62.3	86.5	30.2	68.2	8.8	14.7	0.4	65.0	7.5	41.5	2.2	22.4	0.6	11.7	0.1
	Bowtie	98.1	18.1	98.4	2.6	96.2	0.1	100	0.0	97.1	0.5	100	0.0	0.0	0.0	0.0	0.0
50 single	SHRiMP	99.7	93.3	98.9	92.6	98.0	91.1	94.8	72.5	97.0	89.5	95.3	83.5	93.0	69.6	83.4	25.6
	BFAST	98.9	93.0	97.9	90.5	96.2	83.7	87.7	50.7	95.2	80.4	92.8	68.7	89.0	53.5	78.0	24.6
	BWA	95.3	79.7	93.0	33.7	71.8	2.1	15.2	0.0	89.5	5.6	83.7	1.1	61.9	0.1	0.0	0.0
	Bowtie	95.2	65.5	92.1	15.7	49.1	0.3	2.5	0.0	92.1	2.2	85.4	0.4	36.8	0.0	0.0	0.0
75 single	SHRiMP	99.7	96.0	99.6	95.8	99.4	95.6	98.9	94.4	99.2	95.5	98.8	94.9	98.5	93.7	97.2	79.7
	BFAST	99.3	96.0	99.1	95.6	98.8	95.1	97.4	91.6	98.5	95.1	98.0	94.1	97.4	92.1	94.3	81.6
	BWA	97.5	78.2	97.0	38.0	95.1	6.5	56.4	0.0	96.7	9.4	94.6	1.2	90.4	0.2	100	0.0
	Bowtie	97.4	42.0	96.2	6.0	75.7	0.1	0.0	0.0	95.8	0.8	96.3	0.1	100	0.0	0.0	0.0

David M et al. Bioinformatics 2011;27:1011-1012

Recall rates for four sets of 2 million simulated 72-bp paired-end reads, mapped back to the human reference by five read mapping algorithms.

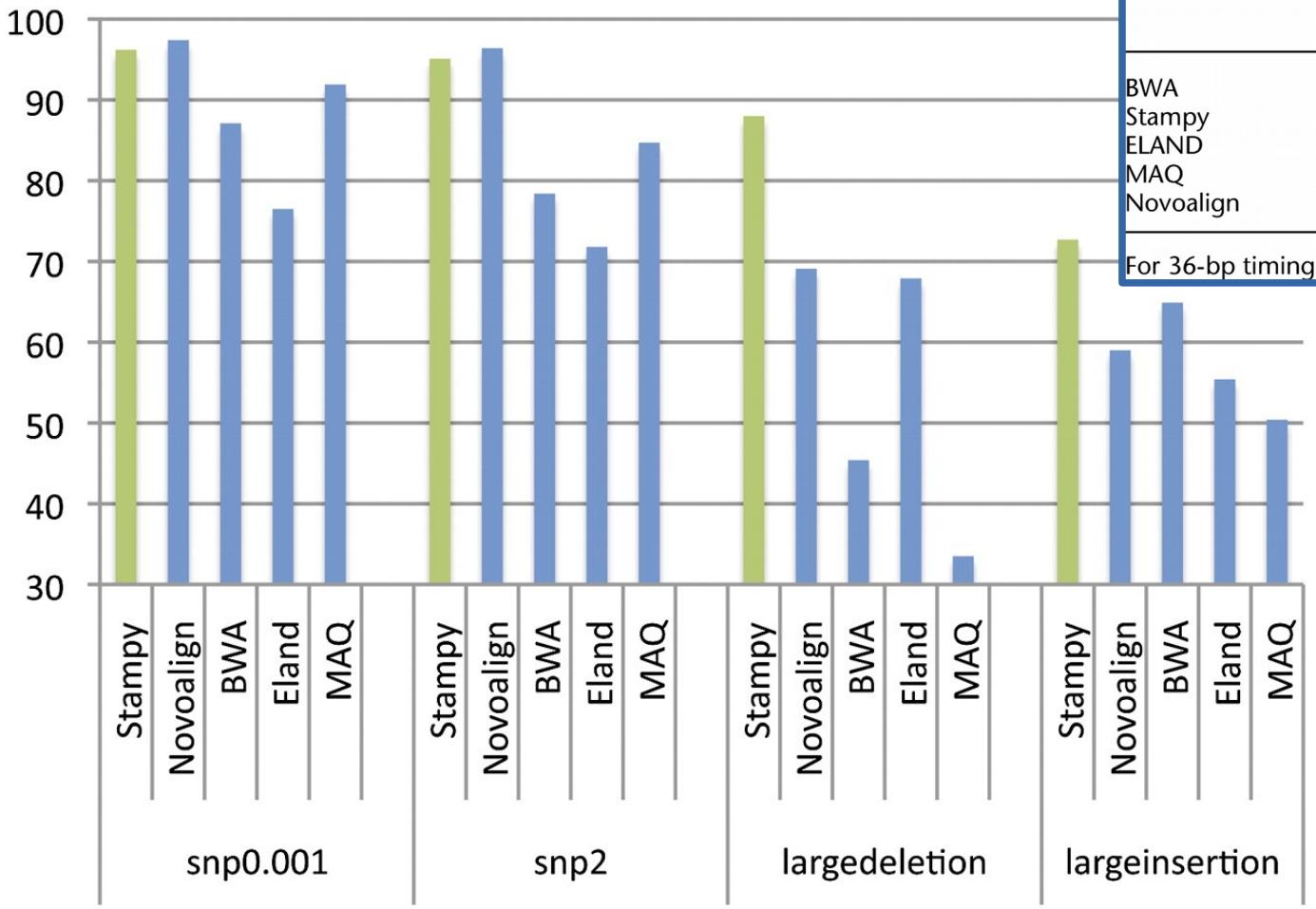


Table 1. CPU hours required for mapping a gigabase of 72-bp sequence data by the five mappers considered here

	Single-end (h/Gb)	Paired-end (h/Gb)
BWA	3.2	3.4
Stampy	10.7	14.6
ELAND	20.7	19.4
MAQ	29.1	24.8
Novoalign	81.1	61.6

For 36-bp timings, see Supplemental Table S2.

Lunter & Goodson Genome Res. 2011;21:936-939

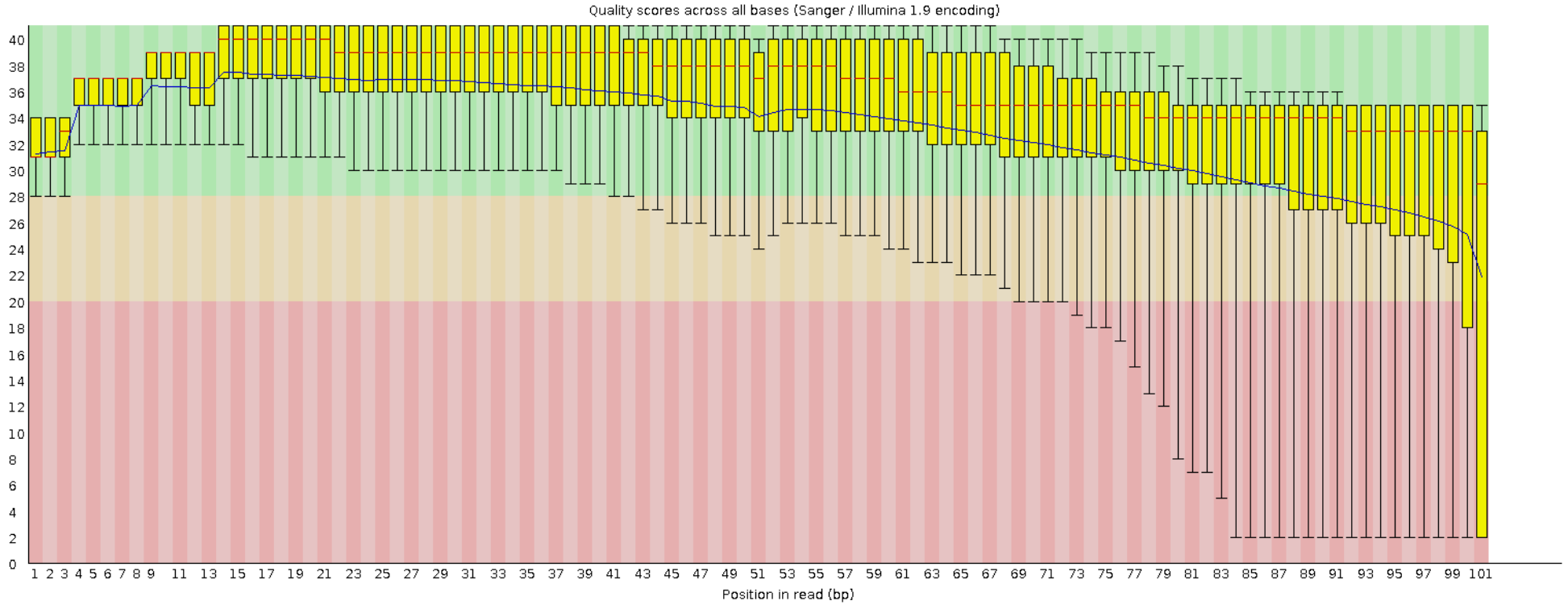
What goes in: fastq reads

```
$ head -4 myreads.fastq
@DHKW5DQ1:192:C0PTHACXX:2:1101:3309:2499 2:N:0:TGCGTGAA
ACACCCCAGCAGCCCGAGTACGGGATAAAGCGGAACATACCGCCTAATTCTTGGCACCAACATAATTTAAGTTCGCGGCGGGAAGCTCGGTAAACATAACC
+
@B@FFFFFHHHHHIJJII<EHEGIGIIJJGIIIGIIJGIGHIEHHFFDFFECEEEDEDDDBBCDDDEDDEC>CDDBDDDDDD-5<8>C><+8?C#######
```

- 4 lines
  - Sequence ID, raw sequence, '+', Quality scores
- Quality used “phred” scores
  - $Q = -10\ log_{10} p$ and $p = 10^{-Q/10}$
  - *If Q = 30, p* = 0.001 (1/1000 probability of error)
  - Encoded as ASCII characters
  - With some offset (usually 33 or 64)

What goes in: fastq reads

- FastQC

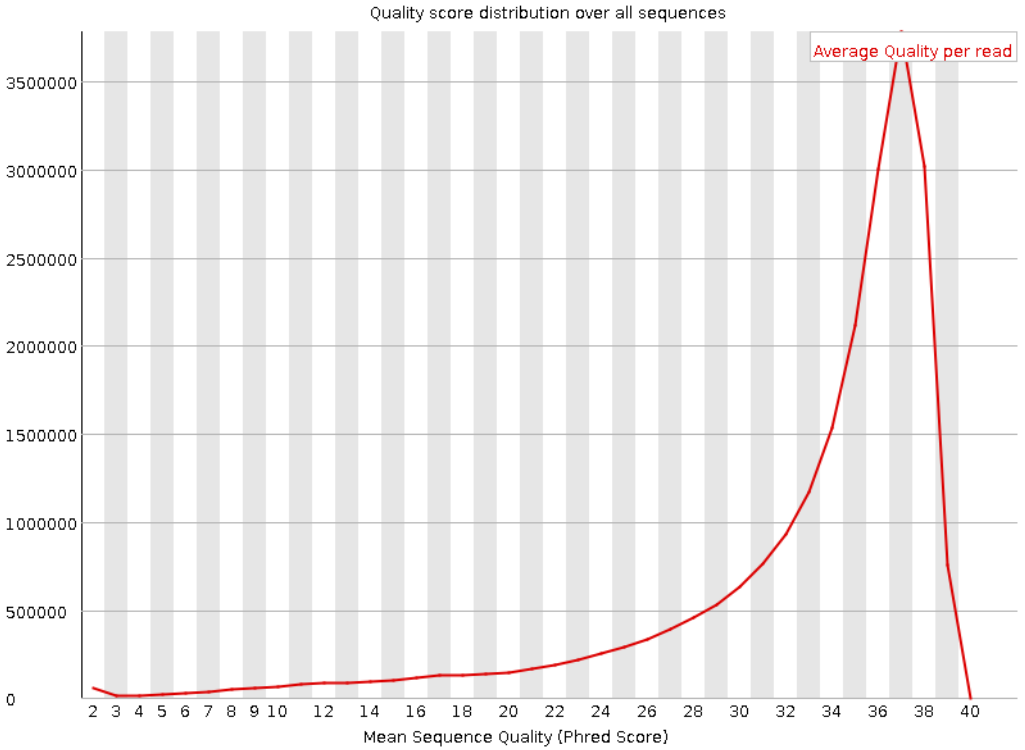


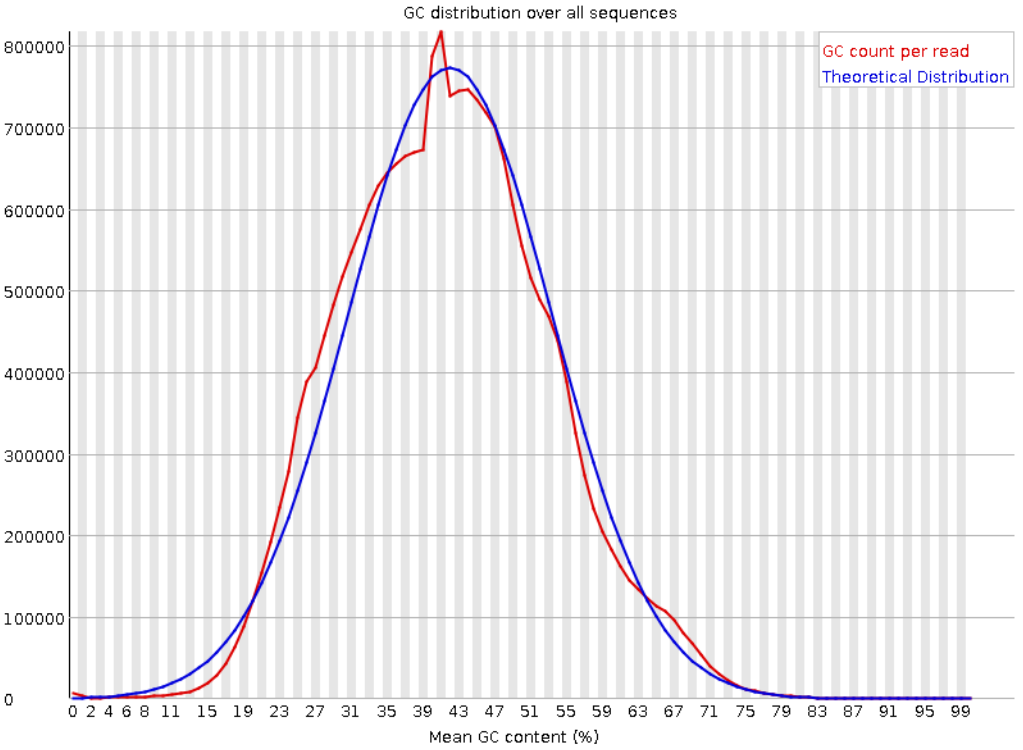
- fastx_trimmer to trim a defined part of reads

```
$ fastx_trimmer -f 1 -l 80 -i myreads.fastq -o myreads.trimmed.fastq
```

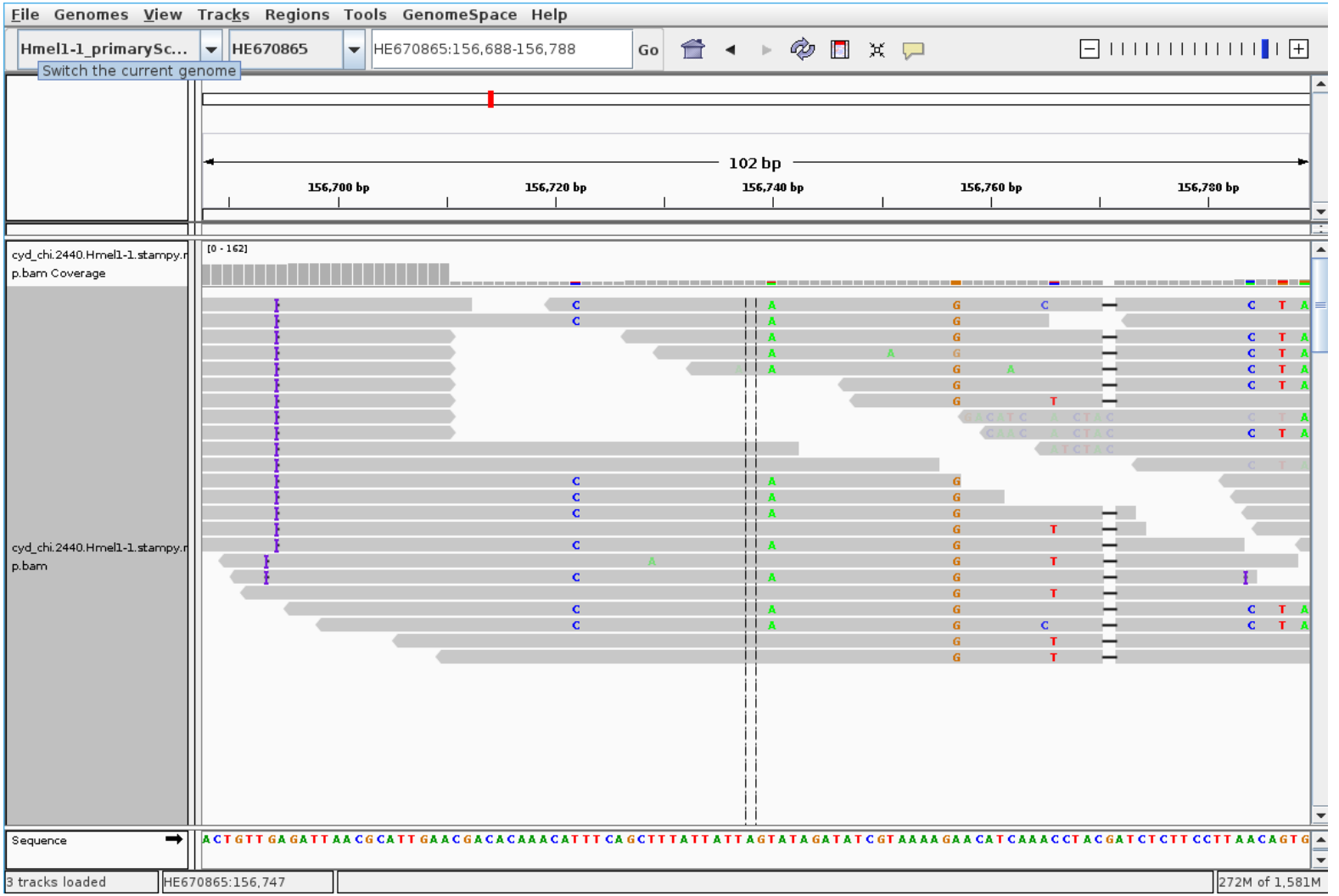
What goes in: fastq reads

- FastQC produces lots of useful graphics





What comes out: SAM/BAM



- Viewers: IGV, Tablet, EagleView, HawkEye, BamView etc.

What comes out: SAM/BAM

- Sequence Alignment/Map format
- Has 'Header' and 'Alignment' Sections

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                      ATAGCT..............TCAGC
-r003                            ttagctTAGGC
-r001/2                                       CAGCGGCAT
```

The corresponding SAM format is:

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                           ttagctTAGGC
-r001/2                                        CAGCGGCAT
```

The corresponding SAM format is:

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

SAM/BAM format

Header Section

- Lines start with `@`
- Format `TAG:VALUE`

```
@HD     VN:1.0  GO:none SO:coordinate
@SQ     SN:HE669515     LN:1113
@SQ     SN:HE669513     LN:11036
@SQ     SN:HE669511     LN:14485
@SQ     SN:HE669509     LN:1831
@SQ     SN:HE669507     LN:15013
@SQ     SN:HE669505     LN:8974
@SQ     SN:HE669503     LN:3243
```

- `@HD` – header line, `VN:` Format version, `SO:` sort order
- `@SQ` – reference sequences, `SN:` sequence name, `LN:` length

SAM/BAM format

Header Section

```
@SQ     SN:HE670532     LN:59893
@SQ     SN:HE669517     LN:203704
@SQ     SN:HE668723     LN:77637
@SQ     SN:HE668283     LN:18120
@SQ     SN:HE669516     LN:5707
@SQ     SN:HE669860     LN:7066
@RG     ID:cyd_cyd.2158 PL:Sanger       SM:cyd_cyd.2158
@PG     ID:dvtgm        PN:stampy       VN:1.0.17_(r1481)
```

- `@RG` – Read groups, `ID:` Unique ID, `SM:` Sample
- `@PG` – Program: `PN:` Program name, `VN:` version

SAM/BAM format

Alignment Section

```
HWUSI-EAS243L_0007:4:33:17261:12460#0   163     HE669357        3266    67      53M5I43M        =       3752
    583     TATCTCTTAAATTTCAATACAACACCAAAAATCTATTCATAATTGATTATTATATTATTTAATAACTTATATTGCAGGTGTGTCCATCTCAAGAATA
CGAA    IIIIIIIIHIIIIIIIIIIIIIIHIIIIIIIIGIIIIIIIGIIIIHGFEIFGIGIIHIIIHGIGIIIIIIEIIIHIIH?GCGEGGIIHIIHGFHHHHIHH8
```

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
2	FLAG	Int	`[0,2`^{16}`-1]`	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	`[0,2`^{31}`-1]`	1-based leftmost mapping POSition
5	MAPQ	Int	`[0,2`^{8}`-1]`	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next read
8	PNEXT	Int	`[0,2`^{31}`-1]`	Position of the mate/next read
9	TLEN	Int	`[-2`^{31}`+1,2`^{31}`-1]`	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

SAM/BAM format

Alignment Section

- Bitwise Flag
  - Twelve possible yes/no answers encoded by a number (but how?)
  - We can chat about this if there is time.

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate
0x800	supplementary alignment

SAM/BAM format

Alignment Section

- CIGAR string
  - Describes locations of matches (or mismatches), insertions, deletions and how long the runs are
  - `53M5I43M` means 53 matches followed by a 5 bp insertion followed by 43 matches
  - NOTE, match doesn't mean the bases are identical, just that they line up.

SAM/BAM format

Alignment Section

- CIGAR string

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                            ttagctTAGGC
-r001/2                                       CAGCGGCAT
```

The corresponding SAM format is:

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

SAM/BAM format

Some simple SAMtools commands (see http://samtools.sourceforge.net/samtools.shtml#1)

Take a look at your bam file

```
samtools view myfile.bam | less
```

Look at the header only

```
samtools view -H myfile.bam | less
```

Compress sam to bam

```
Samtools view -Sb myfile.sam > myfile.bam
```

General mapping statistics

```
samtools flagstat myfile.bam

109057232 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
103930724 + 0 mapped (95.30%:nan%)
109057232 + 0 paired in sequencing
54523649 + 0 read1
54533583 + 0 read2
89500562 + 0 properly paired (82.07%:nan%)
103173202 + 0 with itself and mate mapped
757522 + 0 singletons (0.69%:nan%)
9528350 + 0 with mate mapped to a different chr
1898612 + 0 with mate mapped to a different chr (mapQ>=5)
```

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                           ttagctTAGGC
-r001/2                                        CAGCGGCAT
```

The corresponding SAM format is:

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Handy bam file visualisation

```
samtools tview myfile.bam
```