The majority of data analyses split off after generating a SAM file:

- Sequencing Reads
- Alignment Software
- SAM file
- Chip-Seq
- RNA-Seq
- SNPs

Learning strategy: create test queries:
- exact match,
- one mismatch,
- 1 deletion etc.
- align and look at the output

The structure of the SAM file:

- SAM headers start with @
- Alignments
SAM format: a tabular text format to represent alignments

Published as The Sequence Alignment/Map format and SAMtools by Heng Li et al. Bioinformatics 25, Volume 25, Issue 16, 2009

“A TAB-delimited text format consisting of a header section, which is optional, and an alignment section.”

Resource: SAM specification
11 required column + optional fields

An alignment consists of 11 tab delimited columns

1.4 The alignment section: mandatory fields

Each alignment line has 11 mandatory fields. These fields always appear in the same order and must be present, but their values can be ‘0’ or ‘-’ (depending on the field) if the corresponding information is unavailable. The following table gives an overview of the mandatory fields in the SAM format:

<table>
<thead>
<tr>
<th>Col</th>
<th>Field</th>
<th>Type</th>
<th>Regex/Range</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QNAME</td>
<td>String</td>
<td>[&quot;-?a-?1,2,265&quot;]</td>
<td>Query template NAME</td>
</tr>
<tr>
<td>2</td>
<td>FLAG</td>
<td>Int</td>
<td>[0, 2^20-1]</td>
<td>bitwise FLAG</td>
</tr>
<tr>
<td>3</td>
<td>RNAME</td>
<td>String</td>
<td>[+-]</td>
<td>Reference sequence NAME</td>
</tr>
<tr>
<td>4</td>
<td>POS</td>
<td>Int</td>
<td>[0, 2^20-1]</td>
<td>1-based leftmost mapping POSITION</td>
</tr>
<tr>
<td>5</td>
<td>MAPQ</td>
<td>Int</td>
<td>[0, 2^8-1]</td>
<td>Mapping Quality</td>
</tr>
<tr>
<td>6</td>
<td>CIGAR</td>
<td>String</td>
<td>[+-]</td>
<td>CIGAR string</td>
</tr>
<tr>
<td>7</td>
<td>RNEXT</td>
<td>String</td>
<td>[+-]</td>
<td>Ref. name of the mate/next segment</td>
</tr>
<tr>
<td>8</td>
<td>PNEXT</td>
<td>Int</td>
<td>[0, 2^20-1]</td>
<td>Position of the mate/next segment</td>
</tr>
<tr>
<td>9</td>
<td>TLEN</td>
<td>Int</td>
<td>[-2^8+1, 2^8-1]</td>
<td>observed Template LENGTH</td>
</tr>
<tr>
<td>10</td>
<td>SEQ</td>
<td>String</td>
<td>[+-]</td>
<td>segment SEQUENCE</td>
</tr>
<tr>
<td>11</td>
<td>QUAL</td>
<td>String</td>
<td>[+-]</td>
<td>ASCII of Phred-scaled base QUALITY+33</td>
</tr>
</tbody>
</table>

Column 1 and 2: QNAME and FLAG

(Queue name and bitwise flags)

QNAME: the name of the query sequence

2. FLAG: bitwise FLAG. Each bit is explained in the following table:

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

- Bit 0x4 is the only reliable place to tell whether the segment is unmapped. If 0x4 is set, no assumptions can be made about RNAME, POS, CIGAR, MAPQ, bits 0x2, 0x10 and 0x100 and the bit 0x20 of the next segment in the template.
Column 2: FLAG
the bitwise representation

1 = 00000001 \rightarrow\text{paired end read}
2 = 00000010 \rightarrow\text{mapped as proper pair}
4 = 00000100 \rightarrow\text{unmappable read}
8 = 00001000 \rightarrow\text{read mate unmapped}
16 = 00010000 \rightarrow\text{read mapped on reverse strand}

The flag 11 \rightarrow 1 + 2 + 8 = 0001011 (conditions 1, 2 and 8 satisfied)

It is used to save space – but it does make things a bit more difficult.
Usually very few flags are needed in practice – 0, 4, 16 are the most generic ones

If you need to construct a more complex flag search for explain SAM flags:
http://picard.sourceforge.net/explain-flags.html

Columns 3, 4: RNAME and POS
Reference and Position

Column 4 POS: 1-based leftmost mapping POSITION of the first matching base.

Very important to remember later when we need to find the 5' end (the actual start)

Column 5: MAPQ - Mapping Quality

- Phred score, identical to the quality measure in the fastq file. quality $Q$, probability $P$:

$$P = 10 ^ {\left(-\frac{Q}{10.0}\right)}$$

If $Q=30$, $P=1/1000 \rightarrow$ on average, one of out 1000 alignments will be wrong

As good as this sounds it is not easy to compute such a quality.

Details of the mapping quality computation – hard to find good answers

- Tool specific – there is no standard of what it should be
- The repeat structure of the reference. Reads falling in repetitive regions usually get very low mapping quality.
- The base quality of the read. Low quality means the observed read sequence is possibly wrong, and wrong sequence may lead to a wrong alignment.
- The sensitivity of the alignment algorithm. The true hit is more likely to be missed by an algorithm with low sensitivity, which also causes mapping errors.
- Paired end or not. Reads mapped in pairs are more likely to be correct.

(from the MAQ manual)
BWA specific high scores

A read alignment with a mapping quality 30 or above usually implies

• The overall base quality of the read is good.

• The best alignment has few mismatches.

• The read has few or just one ‘good’ hit on the reference, which means the current alignment is still the best even if one or two bases are actually mutations or sequencing errors.

BWA specific low scores

Surprisingly difficult to track down the exact behavior

• Q=0 → if a read can be aligned equally well to multiple positions, BWA will randomly pick one position and give it a mapping quality zero.

• Q=25 → the edit distance equals mismatches and is greater than zero

Column 6: CIGAR

• CIGAR = Compact Idiosyncratic Gapped Alignment Report

<table>
<thead>
<tr>
<th>Op</th>
<th>BAM Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>alignment match (can be a sequence match or mismatch)</td>
</tr>
<tr>
<td>I</td>
<td>insertion to the reference</td>
</tr>
<tr>
<td>D</td>
<td>deletion from the reference</td>
</tr>
<tr>
<td>S</td>
<td>skipped region from the reference</td>
</tr>
<tr>
<td>H</td>
<td>soft clipping (clipped sequence present in SEQ)</td>
</tr>
<tr>
<td>E</td>
<td>hard clipping (clipped sequence NOT present in SEQ)</td>
</tr>
<tr>
<td>P</td>
<td>padding (silent deletion from padded reference)</td>
</tr>
<tr>
<td>=</td>
<td>sequence match</td>
</tr>
<tr>
<td>X</td>
<td>sequence mismatch</td>
</tr>
</tbody>
</table>

- If can only be present as the first and/or last operation.
- It may only have N operations between them and the ends of the CIGAR string.
- For mRNA-to-genome alignment, an N operation represents an intron. For other types of alignments, the interpretation of N is not defined.

Columns 7, 8, 9: RNEXT, PNEXT, TLEN

(used in paired end read sequencing)

• RNEXT: the name of the pair

• PNEXT: the position of the pair

• TLEN: the distance between the leftmost positions of the pairs

We will discuss these in more detail later.

These can show the position of reads that are distant – allow us to infer genomic variations
Column 10, 11

- SEQ: the query sequence
- QUAL: the phred encoded quality sequence

SEQ may contain the original sequence or the segment it was aligned to. Not all tools do the same thing.

Column 12 and beyond

Homework 15

Take a subsequence of about 100 bp out of the Ebola genome.

Modify (mutate) the sequence a little bit (so that it still a matches) then align both sequences (original, mutated) with bwa against the genome.

- What did the mapping quality of your mutated sequence change of the mutated sequence?
- At what coordinate does the query align?
- Explain how you can see in the alignment the modifications that you have introduced.

Keep modifying the sequence one small change at the time. How many modifications were you able to make until bwa was unable to align the sequence?