Sequence Alignment

Contents

1 NGS sequencers output

NGS sequencers generate data in FASTQ and CSFASTA (FASTA) formats:

- FASTA files: the most common standard for storing reference or consensus sequence data. FASTA only stores sequences.
- FASTQ files: the most common format for storing raw sequence data. FASTQ stores both sequence and associated sequence quality values (QV).

1.1 FASTQ file

Let's check a FASTQ file:

```
head -n 4 Data/SRR835775_1.first1000.fastq | less -S
```

```
## @SRR835775.1 1/1
## TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTCACCCTAACCCTAACCCTAACCGTATCCGTCACCCTAACCCTAAC
## +
## ???B1ADDD8??BB+C?B+:AA883CEE8?C3@DDD3)?D2;DC?8?=BAD=@C@(.6.6=A?=?@####################################
```

1.2 FASTA file

Now let's check the FASTA file of human genome chromosome 1:

```
# Assemble parts
gunzip -c Reference/Homo_sapiens.GRCh38.dna.chromosome.1.part1.fa.gz > \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part1.fa
gunzip -c Reference/Homo_sapiens.GRCh38.dna.chromosome.1.part2.fa.gz > \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part2.fa
gunzip -c Reference/Homo_sapiens.GRCh38.dna.chromosome.1.part3.fa.gz > \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part3.fa
gunzip -c Reference/Homo_sapiens.GRCh38.dna.chromosome.1.part4.fa.gz > \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part4.fa
gunzip -c Reference/Homo_sapiens.GRCh38.dna.chromosome.1.part5.fa.gz > \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part5.fa
```

```
cat Output/Homo_sapiens.GRCh38.dna.chromosome.1.part1.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part2.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part3.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part4.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part5.fa > \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.fa

rm Output/Homo_sapiens.GRCh38.dna.chromosome.1.part1.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part2.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part3.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part4.fa \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.part5.fa

# Show
head -n 1000 Output/Homo_sapiens.GRCh38.dna.chromosome.1.fa | tail -5 | less -S
```

```
## CCAAATTTCTTTCCAATTCATCTTTGTTCTTCCCTTTCCTTTTTACTCTCTTTAAACATT
## CTATGGACTCTGCCTCCTTCACACTGATATTGAACGCCCATAGTTTCATATTTTGGATTG
## CGATTGTTTTATTTTAAAATGGCAAATGTTCATGTTATAAAGAGAATTTTTCAGTCTTTA
## GACTAATAGGTTCATGTAGTTTGGGATTTTCCTCTTTAAGAAAATTAATTATCACTCACA
## CTCCAAGACAAACACCATTTCAGTAGCAATATGAATTTCAGTAGTAATAGGAATCTCCAA
```

2 Aligned sequences

The SAM file format External link and its compressed version, the BAM format, were developed to store information about the alignment of sequences to a reference genome.

2.1 SAM file

Let's check a SAM file:

```
head -n 10 Data/Sample_sam.sam | tail -1 | less -S
```

```
## SRR1039508.10901400  16  1   32585836    3   43M *   0   0   CTGCTAAAAATACAAAAATCAGCCGGGCCTGGTGGTGTG
```

3 RNA-seq

Initially, we should go from reads to transcripts. Note that a gene might also have different transcripts. Also, it is highly advised to perform some QC and pre-processing before alignment.

Approaches to transcript building:

- Assembly approaches
- Alignment approaches (to a reference genome): BWA, Bowtie2, STAR

Before continuing, let's check the position of a read (which should be on chr16) using a simple BLAST search:

```
library(BSgenome.Hsapiens.UCSC.hg19)
matchPattern(
  "TCGATCCATCGATTGGAAGGCACTGATCTGGACTGTCAGGTTGGTGGTCTTATTTGCAAGTCC",
  Hsapiens$chr16
)
```

```
## Views on a 90354753-letter DNAString subject
## subject: NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN...NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

Figure 1: 2-2-1

```
## views:
##          start      end width
##   [1] 72130081 72130143    63 [TCGATCCATCGATTGGAAGGCAC...TGGTGGTCTTATTTGCAAGTCC]
```

Now let's check another read:

```
library(BSgenome.Hsapiens.UCSC.hg19)
matchPattern(
  "TCGCTCTCTCCGTTTCAGGGAAGCCAGCAAGTCCAGTCCGAGTAATGAAGGGCGGGGAGCAGG",
  Hsapiens$chr16
)
```

```
## Views on a 90354753-letter DNAString subject
## subject: NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN...NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
## views: NONE
```

Nothing? Let's check if it is from the other strand:

```
library(BSgenome.Hsapiens.UCSC.hg19)
matchPattern(reverseComplement(
  DNAString(
    "TCGCTCTCTCCGTTTCAGGGAAGCCAGCAAGTCCAGTCCGAGTAATGAAGGGCGGGGAGCAGG"
  )
),
Hsapiens$chr16)
```

```
## Views on a 90354753-letter DNAString subject
## subject: NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN...NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
## views:
##          start      end width
##   [1] 72130180 72130242    63 [CCTGCTCCCCGCCCTTCATTACT...TTCCCTGAAACGGAGAGAGCGA]
```

3.1 QC and pre-processing

First, let's run QC using FASTQC algorithm:

```
fastqc --noextract Data/SRR1039508.fastq --outdir Output/
```

```
## Started analysis of SRR1039508.fastq
## Approx 5% complete for SRR1039508.fastq
## Approx 10% complete for SRR1039508.fastq
## Approx 15% complete for SRR1039508.fastq
## Approx 20% complete for SRR1039508.fastq
## Approx 25% complete for SRR1039508.fastq
## Approx 30% complete for SRR1039508.fastq
## Approx 35% complete for SRR1039508.fastq
## Approx 40% complete for SRR1039508.fastq
## Approx 45% complete for SRR1039508.fastq
## Approx 50% complete for SRR1039508.fastq
## Approx 55% complete for SRR1039508.fastq
## Approx 60% complete for SRR1039508.fastq
## Approx 65% complete for SRR1039508.fastq
## Approx 70% complete for SRR1039508.fastq
## Approx 75% complete for SRR1039508.fastq
## Approx 80% complete for SRR1039508.fastq
## Approx 85% complete for SRR1039508.fastq
## Approx 90% complete for SRR1039508.fastq
## Approx 95% complete for SRR1039508.fastq
## Approx 100% complete for SRR1039508.fastq
## Analysis complete for SRR1039508.fastq
```

The output looks like this:

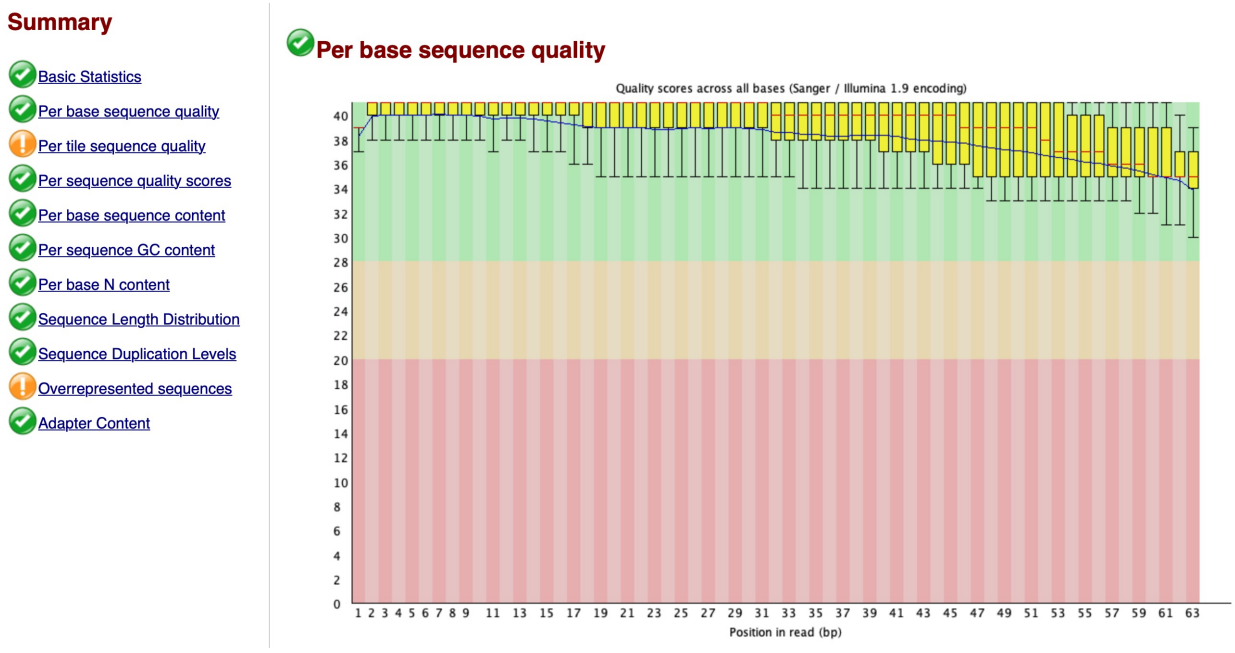

Figure 2: 2-2-2

Now let's do some preprocessing. Steps required include:

- Filter the data by removing adapters.
- Trim low quality cycles.
- Remove low quality reads.

A good program for Illumina data is **trimmomatic**:

```
Java -jar /Users/amirvalizadeh/Trimmomatic/dist/jar/trimmomatic-0.40-rc1.jar \
SE -phred33 Data/SRR1039508.fastq Output/SRR1039508_qc.fastq \
ILLUMINACLIP:/Users/amirvalizadeh/Trimmomatic/adapters/TruSeq3-PE.fa:2:30:10 \
LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:50 HEADCROP:10 CROP:80
```

```
## TrimmomaticSE: Started with arguments:
##  -phred33 Data/SRR1039508.fastq Output/SRR1039508_qc.fastq ILLUMINACLIP:/Users/amirvalizadeh/Trimmoma
## Automatically using 4 threads
## Using PrefixPair: 'TACACTCTTTCCCTACACGACGCTCTTCCGATCT' and 'GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT'
## ILLUMINACLIP: Using 1 prefix pairs, 0 forward/reverse sequences, 0 forward only sequences, 0 reverse
## Input Reads: 100000 Surviving: 97309 (97.31%) Dropped: 2691 (2.69%)
## TrimmomaticSE: Completed successfully
```

Notice that:

- **SE** means single end reads, for paired-end data use **PE**.
- **phred** is the quality scoring scheme.
- **ILLUMINACLIP** points to the file with the adapters.
- **LEADING** and **TRAILING** remove bases at the beginning and end of the reads below a score of 3.
- **SLIDINGWINDOW** slides across the sequence and removes those inside a window of length 4 that have an average quality below 15.
- **MINLEN** removes sequences that are shorter than 50 after filtering.
- **HEADCROP** removes the first 10 sequences at the start of the run.
- **CROP** deletes everything after the first 80 bases (recall the quality was not so good after 80 cycles).

Now, let's run QC again:

```
fastqc --noextract Output/SRR1039508_qc.fastq --outdir Output/
```

```
## Started analysis of SRR1039508_qc.fastq
## Approx 5% complete for SRR1039508_qc.fastq
## Approx 10% complete for SRR1039508_qc.fastq
## Approx 15% complete for SRR1039508_qc.fastq
## Approx 20% complete for SRR1039508_qc.fastq
## Approx 25% complete for SRR1039508_qc.fastq
## Approx 30% complete for SRR1039508_qc.fastq
## Approx 35% complete for SRR1039508_qc.fastq
## Approx 40% complete for SRR1039508_qc.fastq
## Approx 45% complete for SRR1039508_qc.fastq
## Approx 50% complete for SRR1039508_qc.fastq
## Approx 55% complete for SRR1039508_qc.fastq
## Approx 60% complete for SRR1039508_qc.fastq
## Approx 65% complete for SRR1039508_qc.fastq
## Approx 70% complete for SRR1039508_qc.fastq
## Approx 75% complete for SRR1039508_qc.fastq
## Approx 80% complete for SRR1039508_qc.fastq
## Approx 85% complete for SRR1039508_qc.fastq
## Approx 90% complete for SRR1039508_qc.fastq
## Approx 95% complete for SRR1039508_qc.fastq
## Analysis complete for SRR1039508_qc.fastq
```

The output looks like this:

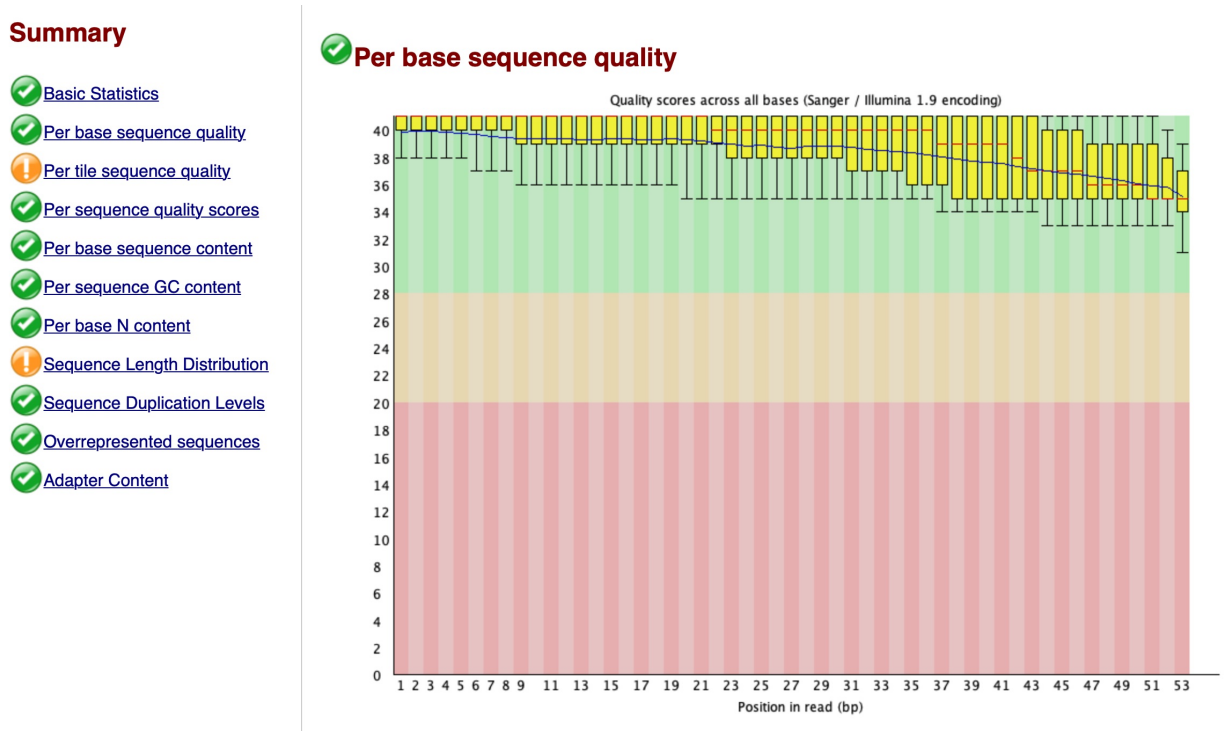

Figure 3: 2-2-3

3.2 Sequence alignment with STAR

1- Prepare the reference genome FASTA file (only chromosome 1 for this example). Take a peak:

```
head -n 1000 Output/Homo_sapiens.GRCh38.dna.chromosome.1.fa | tail -10 | less -S
```

```
## CACCTATTCACTCACAAGCTTAAACTCTTAACTTTTCTCCACATATCAGTGACTATTTCC
## TACAGCTTTTCTTTTACTTTCCATGTTTGCAGTGACAATATACATAAACAGTGTATGAAA
## ACTCAAGTAAAATCTACTCTCTCAGGTGTTCATAATGTATCAATGTATATTGCTTTAAGC
## CTGAAGGTAACCTAAGTAAAGATGTACCATGTTCCACCAATGCTTCTTTTGATCATCATT
## TTATCCTGTTTTTTCTTTAGGATTCTTTCTTATTCCTTCCCCTGACCCTTCTTTTATTCT
## CCAAATTTCTTTCCAATTCATCTTTGTTCTTCCCTTTCCTTTTTACTCTCTTTAAACATT
## CTATGGACTCTGCCTCCTTCACACTGATATTGAACGCCCATAGTTTCATATTTTGGATTG
## CGATTGTTTTATTTTAAAATGGCAAATGTTCATGTTATAAAGAGAATTTTTCAGTCTTTA
## GACTAATAGGTTCATGTAGTTTGGGATTTTCCTCTTTAAGAAAATTAATTATCACTCACA
## CTCCAAGACAAACACCATTTCAGTAGCAATATGAATTTCAGTAGTAATAGGAATCTCCAA
```

2- Prepare the reference genome annotation (gtf) file (only chromosome 1 again). Take a peak:

```
# Assemble
gunzip -c Reference/Homo_sapiens.GRCh38.107.chrom1.gtf.gz > \
Output/Homo_sapiens.GRCh38.107.chrom1.gtf

# Show
head -n 1 Output/Homo_sapiens.GRCh38.107.chrom1.gtf | head -c 41 | less -S
```

```
## 1    ensembl_havana  gene    1471765 1497848 .   +
```

3- Run the code below to generate the genome index:

```
STAR --runThreadN 12 \
--runMode genomeGenerate \
--genomeSAindexNbases 12 \
--genomeDir Output \
--genomeFastaFiles Output/Homo_sapiens.GRCh38.dna.chromosome.1.fa \
--sjdbGTFfile Output/Homo_sapiens.GRCh38.107.chrom1.gtf \
--sjdbOverhang 62
```

```
##  STAR --runThreadN 12 --runMode genomeGenerate --genomeSAindexNbases 12 --genomeDir Output --genomeFa
##  STAR version: 2.7.10a_alpha_220207   compiled:  :/Users/travis/build/alexdobin/travis-tests/STARcomp
## Sep 28 00:22:02 ..... started STAR run
## Sep 28 00:22:02 ... starting to generate Genome files
## Sep 28 00:22:35 ..... processing annotations GTF
## Sep 28 00:23:01 ... starting to sort Suffix Array. This may take a long time...
## Sep 28 00:23:13 ... sorting Suffix Array chunks and saving them to disk...
## Sep 28 00:31:30 ... loading chunks from disk, packing SA...
## Sep 28 00:31:51 ... finished generating suffix array
## Sep 28 00:31:51 ... generating Suffix Array index
## Sep 28 00:33:05 ... completed Suffix Array index
## Sep 28 00:33:05 ..... inserting junctions into the genome indices
## Sep 28 00:34:20 ... writing Genome to disk ...
## Sep 28 00:34:21 ... writing Suffix Array to disk ...
## Sep 28 00:34:24 ... writing SAindex to disk
## Sep 28 00:34:25 ..... finished successfully
```

4- Run the code below to map the reads to the indexed genome:

```
STAR --runThreadN 12 \
--genomeDir Output \
--readFilesIn Output/SRR1039508_qc.fastq

mv Aligned.out.sam Output/Aligned.out.sam
mv Log.final.out Output/Log.final.out
mv Log.progress.out Output/Log.progress.out
mv SJ.out.tab Output/SJ.out.tab
mv Log.out Output/Log.out
```

```
##  STAR --runThreadN 12 --genomeDir Output --readFilesIn Output/SRR1039508_qc.fastq
##  STAR version: 2.7.10a_alpha_220207   compiled:  :/Users/travis/build/alexdobin/travis-tests/STARcomp
## Sep 28 00:34:26 ..... started STAR run
## Sep 28 00:34:26 ..... loading genome
## Sep 28 00:34:35 ..... started mapping
## Sep 28 00:36:04 ..... finished mapping
## Sep 28 00:36:04 ..... finished successfully
```

Check the log file:

```
cat Output/Log.final.out
```

```
##                                 Started job on |    Sep 28 00:34:26
##                              Started mapping on |    Sep 28 00:34:35
##                                     Finished on |    Sep 28 00:36:04
##        Mapping speed, Million of reads per hour |    3.94
##
##                            Number of input reads |    97309
##                        Average input read length |    52
```

```
##                                 UNIQUE READS:
##                Uniquely mapped reads number |    18097
##                     Uniquely mapped reads % |    18.60%
##                       Average mapped length |    52.39
##                    Number of splices: Total |    1608
##         Number of splices: Annotated (sjdb) |    1521
##                    Number of splices: GT/AG |    1575
##                    Number of splices: GC/AG |    21
##                    Number of splices: AT/AC |    1
##            Number of splices: Non-canonical |    11
##                   Mismatch rate per base, % |    2.47%
##                      Deletion rate per base |    0.02%
##                     Deletion average length |    1.58
##                     Insertion rate per base |    0.02%
##                    Insertion average length |    1.53
##                          MULTI-MAPPING READS:
##     Number of reads mapped to multiple loci |    2255
##          % of reads mapped to multiple loci |    2.32%
##     Number of reads mapped to too many loci |    108
##          % of reads mapped to too many loci |    0.11%
##                               UNMAPPED READS:
##   Number of reads unmapped: too many mismatches |    0
##        % of reads unmapped: too many mismatches |    0.00%
##             Number of reads unmapped: too short |    76800
##                  % of reads unmapped: too short |    78.92%
##                Number of reads unmapped: other |    49
##                     % of reads unmapped: other |    0.05%
##                                   CHIMERIC READS:
##                     Number of chimeric reads |    0
##                          % of chimeric reads |    0.00%
```

Check some lines of the output:

```
head -n 40 Output/Aligned.out.sam | tail -1 | less -S
```

```
## SRR1039508.8775416   16  1   120152584   255 53M *   0   0   AGGAGCAAATAAGGAAAGAGGTATTACAAACCAAGAGTA
```

Finally, it's better to convert SAM files to BAM:

```
samtools view -S -b Output/Aligned.out.sam > Output/Aligned.out.bam
```

Let's import it now and check its data:

```
library(Rsamtools)
bam <- scanBam("Output/Aligned.out.bam")
```

Field	Type	Brief description
QNAME	String	Query template name
FLAG	Int	bitwise flag
RNAME	String	Reference sequence name
POS	Int	1-based leftmost mapping position
MAPQ	Int	Mapping quality
CIGAR	String	CIGAR string
RNEXT	String	Reference name of the mate/next read
PNEXT	Int	Position of the mate/next read
TLEN	Int	Observed template length

Field	Type	Brief description
SEQ	String	Segment sequence
QUAL	String	ASCII of Phred-scaled base quality

QNAME: Reads/segments having identical QNAME are regarded to come from the same template. A QNAME '' *indicates the information is unavailable.* ***FLAG:*** *This data could be translated using the following website:* http://broadinstitute.github.io/picard/explain-flags.html*

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

Figure 4: 2-2-4

RNAME: Reference sequence name of the alignment. **POS:** 1-based leftmost mapping POSition of the first CIGAR operation that "consumes" a reference base. The first base in a reference sequence has coordinate 1. POS is set as 0 for an unmapped read without coordinate. **MAPQ:** Mapping quality. A value 255 indicates that the mapping quality is not available. **CIGAR:**

Op	BAM	Description	Consumes query	Consumes reference
M	0	alignment match (can be a sequence match or mismatch)	yes	yes
I	1	insertion to the reference	yes	no
D	2	deletion from the reference	no	yes
N	3	skipped region from the reference	no	yes
S	4	soft clipping (clipped sequences present in SEQ)	yes	no
H	5	hard clipping (clipped sequences NOT present in SEQ)	no	no
P	6	padding (silent deletion from padded reference)	no	no
=	7	sequence match	yes	yes
X	8	sequence mismatch	yes	yes

Figure 5: 2-2-5

RNEXT: Reference sequence name of the primary alignment of the next read in the template. For the last read, the next read is the first read in the template. **PNEXT:** 1-based Position of the primary alignment of the next read in the template. Set as 0 when the information is unavailable. This field equals POS at the primary line of the next read. **TLEN:** For primary reads where the primary alignments of all reads in the template are mapped to the same reference sequence, the absolute value of TLEN equals the distance between the mapped end of the template and the mapped start of the template **SEQ:** If not a '*', the length

of the sequence must equal the sum of lengths of M/I/S/=/X operations in CIGAR. An '=' denotes the base is identical to the reference base. **QUAL:** Base quality.

Let's check the data for our example:

```
bam[[1]]$qname[1:4]
```

```
## [1] "SRR1039508.12982779" "SRR1039508.15992012" "SRR1039508.18072752"
## [4] "SRR1039508.21513031"
```

```
bam[[1]]$flag[1:4]
```

```
## [1] 16 16  0  0
```

```
bam[[1]]$strand[1:4]
```

```
## [1] - - + +
## Levels: + - *
```

```
bam[[1]]$rname[1:4]
```

```
## [1] 1 1 1 1
## Levels: 1
```

```
bam[[1]]$pos[1:4]
```

```
## [1] 104153271 180189598  55154187  58781345
```

```
bam[[1]]$cigar[1:4]
```

```
## [1] "53M" "53M" "53M" "53M"
```

```
bam[[1]]$seq[1:4]
```

```
## DNAStringSet object of length 4:
##     width seq
## [1]    53 GGGGCCCTTCAGAAGTCGCTGGGCTCAGAAGGCTCTTAGTCGTGCTTGAGAGT
## [2]    53 ATGAATGGCTCAAGAGGCAGAAGAGAAATAAAATTCCCTACAGTTTCTTTAAA
## [3]    53 CACTAAGGATTGTCAGGTGCTCCTCCAAGGCCTGCTGAATAAGGCTACTGGGC
## [4]    53 TCTTAAATATCCTTTCTGGAATTTTCAGAAACAAAACATAAAAAAATTATATA
```

```
bam[[1]]$qual[1:4]
```

```
## PhredQuality object of length 4:
##     width seq
## [1]    53 DDDDDEEEFFFFFFHIIHIHC=JJJJJJJJJJJJJJJIGIJIJJJJJJJJJJ
## [2]    53 HHHJJJJJJJJJJJJJJJJJJJJJJJJIJJJJJJJJJGHJJJJJJJJJJJJJ
## [3]    53 JJJJJJJJHJIJIJJJJFHIJJJJIJIJJJJJJJJIJJJJJJJJJIIJJJJJI
## [4]    53 JIJJJJJJJJGHIGJJJJJHIJJJJJIIIJJJJJJJJJJJIJJJIGHIJJHFE
```

At last, let's delete the output files:

```
rm Output/Aligned.out.bam Output/Aligned.out.sam Output/chrLength.txt \
Output/chrName.txt Output/chrNameLength.txt Output/chrStart.txt \
Output/exonGeTrInfo.tab Output/exonInfo.tab Output/geneInfo.tab \
Output/Genome Output/genomeParameters.txt Output/Log.final.out \
Output/Log.out Output/Log.progress.out Output/SA Output/SAindex \
Output/SJ.out.tab Output/sjdbInfo.txt Output/sjdbList.fromGTF.out.tab \
Output/sjdbList.out.tab Output/transcriptInfo.tab \
Output/Homo_sapiens.GRCh38.107.chrom1.gtf \
Output/Homo_sapiens.GRCh38.dna.chromosome.1.fa
```