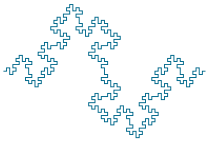
Bios 824: HTS Module

Bios 824: HTS File Formats

Biostatistics and Bioinformatics



Spring 2019

Section 1

Introductory Remarks

HTS versus NGS

- NGS: Next Generation Sequencing
- NGS assays were proposed to replace array based genomic assays (RNA microarray and genome-wide genotyping arrays)
- HTS: High-Throughout Sequencing
- Sequencing assays are technologies of today
- NGS is an outdated term
- I suggest that you use HTS to refer to these technologies

Reference-based approach

- Given is a library of sequencing reads (data): $R_1, \ldots, R_n$
- Each read R_i is a string of neucleotide letters (*e.g.*, $R_1 =$GGAGATGAGTA, $R_2 =$GACCACNTCAGC)
- Each read R_i consists of L_i base *calls* $\tilde{B}_{i1}, \ldots, \tilde{B}_{iL_i}$
- Under a reference-based approach, it is typically assumed that each read *originates* from a *reference*
- One of the key objectives is then to *map*, using a computational algorithm, each read back to this reference
- Note that the algorithm may map reads to the wrong place in the reference or fail to map reads
- We will exclusively focus on reference-based approaches
- There is an active field of development for reference-free approaches

Outline:HTS standard file formats

- FASTA format: represent references
- FASTQ format: represent unaligned sequence data
- SAM/BAM: file format for representing mapped (to a reference) sequencing data
- Pile-up: file format for presenting base calls from DNA-Seq
- GTF/GFF: file format for identifying locations of genomic features (*e.g.*, genes, exons)
- VCF: file format for summarizing genotype and mutation calls (skip)

Section 2

FASTA Format

FASTA Format

```
>seq1
ATATNTGATATAGACCTTCACGGGCCACACATTGGAGGATTCCCGGGC
>seq2
GTGTAGTANGATGAGGAGGNCTA
>seq3
AATATGATGATCCTCATAG
```

- Each record consists of two lines
- Description line: Prefixed by > is a label for the sequence
- Second line: A nucleotide sequence

FASTA file for genomes of organisms

`ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_29/GRCh38.primary_assembly.genome.fa.gz`

- The description line for each record is typically a chromosome
- The corresponding sequence is a long string
- This is a simplistic description

FASTA: Simple Example

```
>seq1
ATATNTGATATAGACCTTCACGGGCCACACATTGGAGGATTCCCGGGC
read1:             ACGGGCCACA                    <- match
read2:       ACCTTCACG                           <- match
read3:          TTCCCGGGC              TTCCCGAGC <-?????
```

Section 3

FASTQ Format

FASTQ: Overview

https://en.wikipedia.org/wiki/FASTQ_format

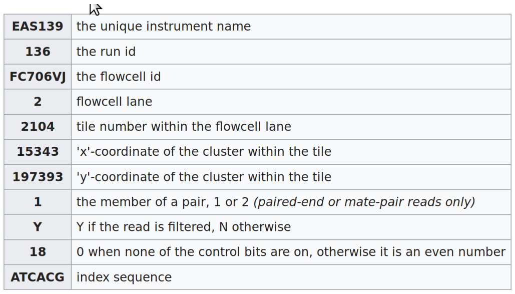
```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

- For each sequencing read, the FASTQ files holds a record consisting of four lines
  1. the read id (sequence identifier)
  2. the called read
  3. a +
  4. Phred scores (same length as the read)

FASTQ: Illumina sequence identifiers

- The read id for each record must be unique
- Illumina uses rather descriptive read ids

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```



EAS139	the unique instrument name
136	the run id
FC706VJ	the flowcell id
2	flowcell lane
2104	tile number within the flowcell lane
15343	'x'-coordinate of the cluster within the tile
197393	'y'-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 *(paired-end or mate-pair reads only)*
Y	Y if the read is filtered, N otherwise
18	0 when none of the control bits are on, otherwise it is an even number
ATCACG	index sequence

Section 4

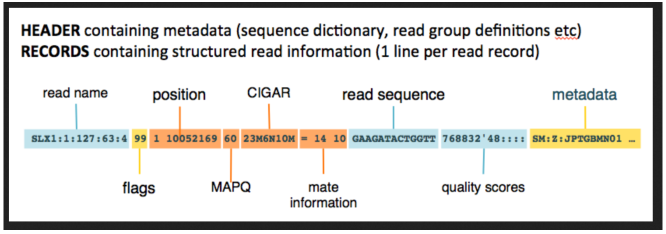
SAM/BAM Format

SAM/BAM

- SAM: Sequence Alignment/Map
- BAM: Binary version of SAM
- Specifications: `http://samtools.github.io/hts-specs/SAMv1.pdf`

SAM: Overview

`https://gatkforums.broadinstitute.org/gatk/discussion/11014/sam-bam-cram-mapped-sequence-data-formats`



HEADER containing metadata (sequence dictionary, read group definitions etc)
RECORDS containing structured read information (1 line per read record)

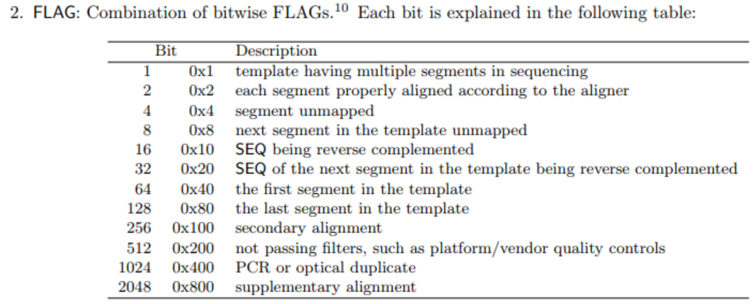
```
SLX1:1:127:63:4 99 1 10052169 60 23M6N10M = 14 10 GAAGATACTGGTT 768832'48:::: SM:Z:JPTGBMN01 …
```

FASTQ: Column names

- QNAME: Query template NAME.
- FLAG: Combination of bitwise FLAGs
- RNAME: Reference sequence NAME of the alignment.
- POS: 1-based leftmost mapping POSition of the first matching base.
- MAPQ: MAPping Quality. It equals -10 log10 prob of mapping position is wrong, rounded to the nearest integer.
- CIGAR: Concise Idiosyncratic Gapped Alignment Report (CIGAR) string.
- RNEXT: Reference sequence name of the primary alignment of the NEXT read in the template.
- PNEXT: Position of the primary alignment of the NEXT read in the template
- TLEN: signed observed Template LENgth.
- SEQ: segment SEQuence.
- QUAL: ASCII of base QUALity plus 33

FLAG

https://samtools.github.io/hts-specs/SAMv1.pdf

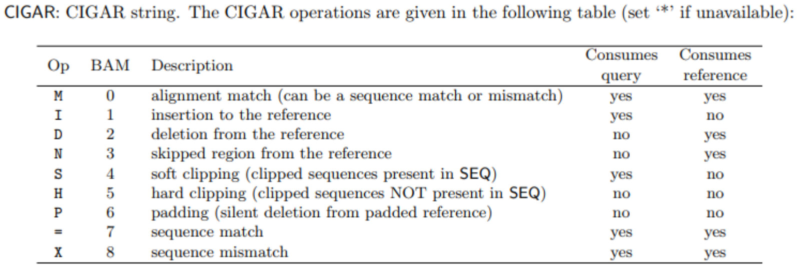


2. FLAG: Combination of bitwise FLAGs.[10] Each bit is explained in the following table:

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

CIGAR

https://samtools.github.io/hts-specs/SAMv1.pdf

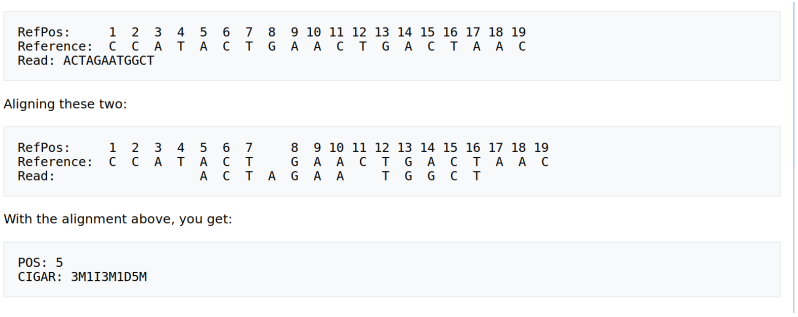


CIGAR: CIGAR string. The CIGAR operations are given in the following table (set '*' if unavailable):

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

CIGAR: Example

https://genome.sph.umich.edu/wiki/SAM



```
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: ACTAGAATGGCT
```

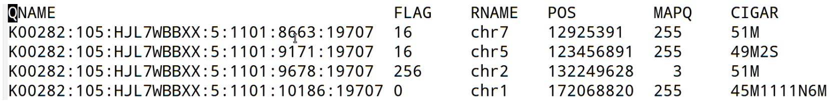
Aligning these two:

```
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       T   G   G   C   T
```

With the alignment above, you get:

```
POS: 5
CIGAR: 3M1I3M1D5M
```

CIGAR/FLAG Example



```
QNAME                                   FLAG    RNAME   POS        MAPQ    CIGAR
K00282:105:HJL7WBBXX:5:1101:8663:19707  16      chr7    12925391   255     51M
K00282:105:HJL7WBBXX:5:1101:9171:19707  16      chr5    123456891  255     49M2S
K00282:105:HJL7WBBXX:5:1101:9678:19707  256     chr2    132249628    3     51M
K00282:105:HJL7WBBXX:5:1101:10186:19707 0       chr1    172068820  255     45M1111N6M
```

Section 5

Pile-up Format

Pile up: Example

http://samtools.sourceforge.net/pileup.shtml

```
seq1 272 T 24  ,.$.....,,.,.,...,,,.,..^+. <<<+;<<<<<<<<<<<=<;<;7<&
seq1 273 T 23  ,.....,,.,.,...,,,.,..A <<<;<<<<<<<<<3<=<<<;<<+
seq1 274 T 23  ,.$....,,.,.,...,,,.,...    7<7;<;<<<<<<<<<=<;<;<<6
seq1 275 A 23  ,$....,,.,.,...,,,.,...^l.  <+;9*<<<<<<<<<=<<:;<<<<
seq1 276 G 22  ...T,,.,.,...,,,.,....  33;+<<7=7<<7<&<<1;<<6<
seq1 277 T 22  ....,,.,.,.C.,,,.,..G.  +7<;<<<<<<<&<=<<:;<<&<
seq1 278 G 23  ....,,.,.,...,,,.,....^k.   %38*<<;<7<<7<=<<<;<<<<<
seq1 279 C 23  A..T,,.,.,...,,,.,..... ;75&<<<<<<<<<=<<<9<<:<<
```

Pile up: columns

http://samtools.sourceforge.net/pileup.shtml

```
seq1 277 T 22  ....,,.,.,.C.,,,.,..G.  +7<;<<<<<<<&<=<<:;<<&<
```

- chromosome
- 1-based coordinate
- reference base
- the number of reads covering the site
- read bases
- base qualities

Pile up: Read Bases

http://samtools.sourceforge.net/pileup.shtml

```
seq1 277 T 22  ....,,.,.,.C.,,,.,..G.  +7<;<<<<<<<&<=<<:;<<&<
```

- dot (.) base call *agrees* with reference base on forward strand
- comma (,) base call *agrees* with reference base on reverse strand
- ACGTN base call *disagrees* with reference base on forward strand
- acgtn base call *disagrees* with reference base on reverse strand

Pile up: Insertions

http://samtools.sourceforge.net/pileup.shtml

```
seq2 156 A 11  .$......+2AG.+2AG.+2AGGG    <975;:<<<<<
```

- `\+[0-9]+[ACGTNacgtn]+`: indicates there is an insertion between this reference position and the next reference position
- Example: 2bp insertions on three reads

Pile up: Deletions

http://samtools.sourceforge.net/pileup.shtml

```
seq3 200 A 20 ,,,,,..,.-4CACC.-4CACC....,.,,.^~. ==<<<<<<<<<<<::<;2<<
```

- `\-[0-9]+[ACGTNacgtn]+`: indicates there is deletion between this reference position and the next reference position
- Example: 4bp insertions on two reads

Pile up: Quality Reads

http://samtools.sourceforge.net/pileup.shtml

```
seq1 277 T 22  ....,,.,.,.C.,,,.,..G.  +7<;<<<<<<<&<=<<:;<<&<
```

```python
from math import pow
mycall = '....,,.,.,.C.,,,.,..G.'
myqual = '+7<;<<<<<<<&<=<<:;<<&<'
# Convert first give phred ascii to phred quality scores
print([ord(phscore)-33 for phscore in myqual[0:5]])
# Convert first give phred ascii to base-call error probabilities
```

```
## [10, 22, 27, 26, 27]
```

```python
print([pow(10, -(ord(phscore)-33)/10) for phscore in myqual[0:5]])
```

```
## [0.1, 0.001, 0.001, 0.001, 0.001]
```

Section 6

GTF/GFF Format

GTF/GFF

- Following alignment to a reference, the next step in RNA-Seq analysis is to map reads to genetic features
- Examples: genes or exons
- Some refer to this as mapping or read counting (the number of reads mapped to each feature)
- To this end, one needs to know the locations of the genetic features
- For example: chr6:43782011-43782087 is the location for exon 3 of the gene *VEGFA*
- GTF: Gene Transfer Format
- GFF: General Feature Format

`http://genome.ucsc.edu/goldenPath/help/customTrack.html#GTF`
`https://www.gencodegenes.org/pages/data_format.html`

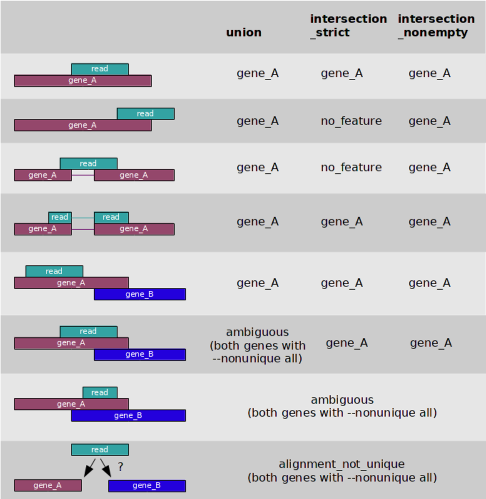
GTF: Example

`https://useast.ensembl.org/info/website/upload/gff.html`

- seqname - name of the chromosome
- source - name of the program that generated this feature, or the data source (database or project name)
- feature - feature type name, e.g. Gene, Variation, Similarity
- start - Start position of the feature, with sequence numbering starting at 1.
- end - End position of the feature, with sequence numbering starting at 1.
- score - A floating point value (unused)
- strand - defined as + (forward) or - (reverse). frame - One of ’0’, ’1’ or ’2’. ’0’ indicates that the first base of the feature is the first base of a codon, ’1’ that the second base is the first base of a codon, and so on..
- attribute - A semicolon-separated list of tag-value pairs, providing additional information about each feature.

Read Counting

`https://htseq.readthedocs.io/en/release_0.9.1/count.html`



	union	intersection_strict	intersection_nonempty
read gene_A	gene_A	gene_A	gene_A
read gene_A	gene_A	no_feature	gene_A
read gene_A gene_A	gene_A	no_feature	gene_A
read read gene_A gene_A	gene_A	gene_A	gene_A
read gene_A gene_B	gene_A	gene_A	gene_A
read gene_A gene_B	ambiguous (both genes with --nonunique all)	gene_A	gene_A
read gene_A gene_B	ambiguous (both genes with --nonunique all)		
read ? gene_A gene_B	alignment_not_unique (both genes with --nonunique all)		

Section 7

VCF Format

VCF: Format

- VCF: Variant Call Format
- Specifications: `http://samtools.github.io/hts-specs/VCFv4.3.pdf`

Peter J. A. Cock, C.J. Fields, N Goto, M. L. Heuer, and P.M. Rice. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, 38(6):1767–1771, 12 2009.

Section 8

Final Project

Data

- FASTQ: `ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR376/ERR376998/ERR376998.fastq.gz`
- Reference barcode library: `https://www.nature.com/nbt/journal/v32/n3/extref/nbt.2800-S7.xlsx`